Preface

This volume contains the abstracts to be presented at SLTC2018: The Seventh Swedish Language Technology Conference 2018 to be held on November 7-9, 2018 in Stockholm.

Note that these abstracts are provided mainly for the benefit of conference participants. This is not a proceeding, and the abstracts are not archival.

There were 34 submissions. Each submission was reviewed by at least three committee members. The committee decided to accept 31 papers.

Thanks to our sponsors:
Gold: Stora Skuggans Värdshus
Silver: TT Nyhetsbyrån and Lingsoft
Bronze: Convertus, Digital Grammars, IQVIA and Voice Provider

Thanks also to Vetenskapsrådet, Riksbankens Jubileumsfond and Wenner-Gren-Stiftelserna for generous grants.

November 6, 2018
Kista

Hercules Dalianis & Mats Wirén
Chairs of SLTC 2018

Program Committee

Yvonne Adesam	Department of Swedish, University of Gothenburg
Lars Ahrenberg	Linköping University
David Alfter	University of Gothenburg
Andrea Andrenucci	Stockholm University
Krasimir Angelov	University of Gothenburg
Henrik Björklund	Umeå University
Johanna Björklund	Dept. Comp. Sci., Umea University SE–901 87 Umea, Sweden
Gerlof Bouma	University of Gothenburg
Hercules Dalianis	DSV-Stockholm University
Dana Dannells	University of Gothenburg
Simon Dobnik	University of Gothenburg
Frank Drewes	Umeå University, Dept. of Computing Science
Martin Duneld	DSV - Department of Computer and Systems Sciences at Stockholm University
Gintare Grigonyte	University of Stockholm
Joakim Gustafson	KTH Royal Institute of Technology
Aron Henriksson	Department of Computer & Systems Sciences, Stockholm University
Christine Howes	University of Gothenburg
Richard Johansson	University of Gothenburg
Arne Jönsson	Department of Computer and Information Science, Linöping University, SE-581 83, Linöping, SWEDEN
Viggo Kann	KTH Royal Institute of Technology
Jussi Karlgren	Gavagai and KTH Royal Institute of Technology
Dimitrios Kokkinakis	UNIVERSITY OF GOTHENBURG
Marco Kuhlmann	Linköping University
Staffan Larsson	University of Gothenburg
Peter Ljunglöf	University of Gothenburg and Chalmers University of Technology
Beata Megyesi	Uppsala University
Joakim Nivre	Uppsala University
Pierre Nugues	Lund University, Department of Computer Science Lund, Sweden
Eva Pettersson	Uppsala University
Aarne Ranta	Chalmers University of Technology
Vasiliki Simaki	Lancaster University
Maria Skeppstedt	DSV, Stockholm University
Eriks Sneiders	Dept. of Computer and Systems Sciences, Stockholm University
Nina Tahmasebi	University of Gothenburg
Sumithra Velupillai	TCS, School of Computer Science and Communication, KTH Royal Institute of Technology
Elena Volodina	Gothenburg University
Rebecka Weegar	Stockholm University
Mats Wirén	Stockholm University
Robert Östling	Department of Linguistics, Stockholm University

Additional Reviewers

Dahlgren, Adam
Jonsson, Anna
Klang, Marcus

Table of Contents

A Generalized Principal Component Analysis for Word Embedding

Ali Basirat

Department of Linguistics and Philology
Uppsala University
ali.basirat@lingfil.uu.se

Abstract

Word embeddings are fundamental objects in neural natural language processing approaches. Despite the fact that word embedding methods follow the same principles, we see in practice that most of the methods that use PCA are not as successful as the methods that are developed in the area of language modelling and make use of neural networks to train word embeddings. In this paper, we address the limiting factors of PCA for word embedding and propose solutions to mitigate those factors. Our experimental results show that principal word embeddings generated with our approach are better than or as good as other sets of word embeddings when they are used in different NLP tasks.

1. Introduction

Word embeddings are algebraic vectors that play an important role in the modern approaches of natural language processing (Collobert et al., 2011; Kalchbrenner and Blunsom, 2013; Chen and Manning, 2014). These vectors provide continuous representations of words and make it possible to use powerful machine learning methods and tools such as deep neural networks to process natural languages.

Different word embedding methods proposed in literature can be divided into two main categories: 1) methods that are developed in the area of distributional semantics (Schütze, 1992; Lund and Burgess, 1996; Landauer and Dumais, 1997; Sahlgren, 2006; Pennington et al., 2014; Lebret and Collobert, 2014; Basirat and Nivre, 2017), and 2) methods that are developed in the area of language modelling (Bengio et al., 2003; Collobert et al., 2011; Mikolov et al., 2013a). Levy and Goldberg (2014) show that these methods are highly connected to each other. In a general view, both categories of word embedding methods generate word embeddings from low-rank factors of a co-occurrence matrix, whose elements are the frequency of seeing words together. The low-rank factorization of the co-occurrence matrix is performed *explicitly* in the methods that are developed in the area of distributional semantics, but it is performed *implicitly* in the methods that are developed in the area of language modelling. The implicit matrix factorization is often computed while training a neural language model, and the explicit matrix factorization is often computed using principal component analysis (PCA).

Despite the fact that word embedding methods follow the same procedure as described above, we see in practice that most of the PCA-based methods that are developed in the area of distributional semantics are not as good as the methods that are developed in the area of language modelling. For example, the hyperspace analogue to language (HAL) (Lund and Burgess, 1996), developed in the area of distributional semantics, is not as successful as word2vec (Mikolov et al., 2013c), developed in the area of language modelling. In this paper, we focus on the PCA-based word embedding methods due to the simplicity and the mathematically well foundations of PCA. First, we study *what are the limitations of using PCA for word embeddings*. Then, we introduce solutions on *how to use PCA for word embedding in effective and efficient ways*. Finally, we compare the results obtained from PCA-based word embedding method and other word embedding methods on different NLP tasks.

2. PCA Limitations

Principal component analysis (PCA) is a method used to study the structure of a data matrix. The main aim of PCA is to reduce the dimensionality of a data set in such a way that most of the variance in the data is retained. PCA deals with the study of the structure of the covariance between a vector of random variables $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_m)^T$. It looks for a vector of independent latent variables $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_k)$ ($k \ll m$), inferred from the original variables, $\mathbf{X}$, that retains most of the variation in the original data. The latent variables, called *principal components*, are linear functions of the original variables:

$$\mathbf{Y} = A^T(\mathbf{X} - \mathbf{E}[\mathbf{X}]) \tag{1}$$

In Eq. 1, $\mathbf{E}[\mathbf{X}]$ is the expected vector of the vector of random variables, $\mathbf{X}$, and the $m \times k$ matrix $A = [A_1 \ldots A_k]$ is composed of the k dominant eigenvectors of the covariance matrix of $\mathbf{X}$.

In word embeddings, the elements of $\mathbf{X}$ are random variables corresponding to contextual units of a language. Each element of $\mathbf{X}$ is a mixture of binomial random variables that counts the frequency of seeing words in the domain of a contextual unit. The components of the mixture model correspond to words of the language. Depending on the number of context units forming $\mathbf{X}$, the distribution of $\mathbf{X}$ can be very far from the normal distribution.

Although PCA does not make any assumption about the data distribution, Jolliffe (2002) argues that it works better on data with a normal distribution. However, co-occurrence data in $\mathbf{X}$ follows a distribution which is closer to Zipfian distribution but far from the normal distribution. This is one of the limiting factors of using PCA for word embedding.

Another limiting factor of using PCA for word embedding is the size of the covariance matrix of $\mathbf{X}$, $\Sigma^{\mathbf{X}}$, whose eigenvalue decomposition is needed to compute word embeddings. Depending on the number of contextual units,

the size of $\Sigma^{\mathbf{X}}$ can be very large. The eigenvalue decomposition of such a matrix needs huge amount of processing resources (i.e., CPU time and memory) that might not be easily accessible in many cases. In practice, instead of computing $\Sigma^{\mathbf{X}}$, we compute principal components by singular value decomposition (SVD) of a data matrix sampled from $\mathbf{X} - \mathbf{E}[\mathbf{X}]$. In word embeddings, this sample matrix is a mean-centred co-occurrence matrix. A co-occurrence matrix is often a large sparse matrix, but a mean-centred co-occurrence matrix is a large dense matrix. The process of computing singular value decomposition of a mean-centred co-occurrence matrix can be very demanding due to the size and the density of the matrix.

To sum up, two limiting factors of using PCA for word embedding are as follows. First, the distribution of the co-occurrence data from which word embeddings are computed is unsuitable for word embedding. Second, the principal component analysis of a mean-centred co-occurrence matrix needs huge amount of processing resources that might not be easily accessible.

3. PCA for Word Embedding

In order to mitigate the first limiting factor of using PCA for word embedding, we propose performing a transformation function on $\mathbf{X}$. The transformation reshapes the data distribution of $\mathbf{X}$ to a distribution that is more suitable for word embedding. To this end, we use the following transformation that maximizes the entropy of $\mathbf{X}$:

$$\hat{\theta} = \arg\max_{\theta} H(f(\mathbf{X}; \theta))$$

where $\mathbf{X}$ is a random vector whose elements are the frequency of seeing words in different contexts, and $f(\mathbf{X}; \theta)$ is an element-wise transformation function defined with parameter θ. The transformation function f can be any monotonically increasing concave function that preserves the given order of the data and magnifies small numbers in its domain. Some examples of such transformation functions are the logarithm, the hyperbolic tangent, and the power transformation functions. Using an optimal value of $\hat{\theta}$, these functions compress data along the top eigenvectors of the covariance matrix of $\mathbf{X}$ and expand data along the remaining eigenvectors of the covariance matrix while preserving the order of eigenvectors with respect to their eigenvalues. As a result, the distribution of the $f(\mathbf{X}; \hat{\theta})$ will be closer to a normal distribution in comparison to the distribution of $\mathbf{X}$.

The second limiting factor of using PCA for word embedding is related to the processing resources needed to decompose a mean-centred co-occurrence matrix. As mentioned above, due to the size of a co-occurrence matrix, it is not easy to compute the SVD of such a matrix using standard methods of SVD. We propose using a randomized SVD algorithm for this aim. Let X be an $m \times n$ matrix sampled from $\mathbf{X}$, E an m-dimensional vector, and $\mathbf{1}_n$ an n-dimensional vector of ones. Algorithm 1, inspired by the randomized matrix factorization method introduced by Halko et al. (2011), returns a rank-k approximation of the singular value decomposition of the data matrix $\mathcal{X} = X - E\mathbf{1}_n^T$, whose columns are centred around the vector E. The algorithm consists of three main steps.

The first is to approximate a rank K basis matrix Q_1 ($k < K \ll n$) that captures most of the information in the input matrix X. Halko et al. (2011) propose setting the parameter $K = \min(m, 2k)$. The rank K basis matrix Q_1 is estimated on lines 2–4 in Algorithm 1. On Line 2, a random matrix is drawn from the standard Gaussian distribution. This random matrix is then used on Line 3 to form an $m \times K$ sample matrix X_1 consisting of K m-dimensional vectors. The columns of X_1 are independent random points sampled from the range of X. This basically means that the basis matrix of the column space of X_1 is approximately equivalent to the basis matrix of the column space of X. This basis matrix is the Q_1 matrix of the QR factorization of the matrix X_1 (see Line 4). Due to the relatively smaller size of X_1 in comparison to the size of X ($m \times K$ versus $m \times n$), the economy-size QR factorization of X_1 can be computed in a more efficient way than the QR factorization of X. This enables us to approximate the basis of the column space of X in an efficient way.

In order to compute the basis of the mean-centred matrix $\mathcal{X} = X - E\mathbf{1}_n^T$, we update the parameter Q_1 with regard to the mean vector E. Line 5 uses the QR-update algorithm proposed by Golub and Van Loan (1996, p. 607) to update the QR factorization of $X_1 = Q_1 R_1$ with respect to the input vector E. For a given QR factorization such as $Q_1 R_1 = X_1$ and two vectors u and v, the QR-update algorithm computes the QR-factorization of

$$QR = X_1 + uv^T$$

by updating the already available factors Q_1 and R_1. Replacing u with $-E$ and v with $\mathbf{1}_n$, the QR-update on Line 5 returns the matrix Q that captures most of the information in the mean-centred matrix $\mathcal{X} = X - E\mathbf{1}_n^T$. Since Q_1 is an approximation of the basis matrix of the column space of X, the matrix Q also can be considered an *approximation* of the basis matrix of the column space of $\mathcal{X}$. Note that we compute the basis matrix of the mean-centred matrix $\mathcal{X}$ without explicitly building the matrix $\mathcal{X}$. This enables us to make use of the sparsity in the co-occurrence matrix X and estimate the QR factorization of $\mathcal{X}$ in an efficient way.

Algorithm 1 The rank-k approximation of the singular value decomposition of $X - E\mathbf{1}_n^T = U\Sigma V^T$.

1: **procedure** CENTRED-SVD(X, E, k, K)
2: Draw an $n \times K$ standard Gaussian matrix Ω
3: Form the sample matrix $X_1 \leftarrow X\Omega$
4: Compute the economy-size QR factorization $X_1 = Q_1 R_1$
5: Compute $QR = Q_1 R_1 - E\mathbf{1}_n^T$ using the QR-update algorithm
6: Form $Y \leftarrow Q^T X - Q^T E\mathbf{1}_n^T$
7: Compute the singular value decomposition of $Y = U_1 \Sigma V^T$
8: $U \leftarrow QU_1$
9: **return** (U, Σ, V)
10: **end procedure**

The second step is to project the matrix $\mathcal{X}$ to the space spanned by Q, i.e., $Y = Q^T\mathcal{X}$. This step is performed on

Line 6. Finally, in the third step, Line 7, the SVD factors of $\mathcal{X}$ are estimated from the $K \times n$ matrix Y in two steps. First, the rank-k SVD approximation of Y is computed using a standard method of singular value decomposition. Then the left singular vectors are updated by $U \leftarrow QU_1$ resulting in $U\Sigma V^T = QY$ (Line 8).

4. Experiments

Using a power transformation function with an optimal power value of $\hat{\theta}$ and the CENTRED-SVD algorithm, we train a PCA-based word embedding model to generate a set of word embeddings. The word embeddings are evaluated on multiple NLP tasks including word similarity benchmarks (Faruqui and Dyer, 2014), part-of-speech tagging, named-entity recognition, and dependency parsing (Chen and Manning, 2014). The results are compared with other results obtained from popular word embedding methods such as `word2vec` consisting of the continuous bag-of-words (`CBOW`) and skip-gram (`SGRAM`) models (Mikolov et al., 2013b), `GloVe` (Pennington et al., 2014), `HPCA` (Lebret and Collobert, 2014), and random indexing (`RI`) (Sahlgren, 2006).

Table 1 summarizes the results. We report the average of all word similarity benchmarks computed by the tool of Faruqui and Dyer (2014) as the word similarity benchmark (column Sim. Corr.). In terms of the average of word similarities, principal word embeddings work better than `RI` and `HPCA`. However, we see that the word vectors generated by `SGRAM` result in a higher value of word similarity correlation than the principal word embeddings. We use the evaluation framework introduced by Nayak et al. (2016) to compute the contributions of word embeddings in part-of-speech tagging and named-entity recognition. In part-of-speech tagging, the word vectors generated by `SGRAM` result in maximum tagging accuracy. The tagging results obtained from the principal word embeddings are higher than the results obtained from `RI`, and `GloVe`, but lower than the other sets of word vectors, `HPCA`, `CBOW`, and `SGRAM`. In named-entity recognition, principal word embeddings work better than `RI`, `HPCA`, and `CBOW`, but weaker than `SGRAM` and `GloVe`. We use the dependency parser proposed by Chen and Manning (2014) to evaluate word embeddings with regard to their contributions in dependency parsing. In dependency parsing, we see that principal word embeddings are among the successful word embeddings. In general, we see that principal word embeddings in many instances (except for NER) are better than or as good as other sets of word embeddings.

A comparison between the efficiency of principal word embeddings and other methods is shown in Table 4.. In order to mitigate the effect of the architectural differences between the different word embedding methods, we first compare `RI`, `HPCA`, `GloVe`, and principal word embeddings. Then we compare principal word embeddings with `word2vec`. The comparison is based on the CPU time needed to perform the dimensionality reduction. We see that in terms of CPU time required to perform the dimensionality reduction, the most efficient method is `RI` which is around five times faster than the principal word embedding. The principal word embedding is almost two times slower than `HPCA` but nine times faster than `GloVe`. All of these methods require the same amount of time, around two hours, to scan the training corpus, build and transform the contextual matrix. Therefore, the total amount of time required by the principal word embedding method to extract a set of word vectors from the raw corpus is around two hours and 15 minutes. By contrast, `word2vec` needs more than ten hours to generate the final word vectors. This shows that the principal word embedding method is faster than `GloVe` and `word2vec` but slower than `RI` and `HPCA`. Together with what is shown in Table 1, this shows that the principal word embedding method is more efficient than `word2vec` and `GloVe` and on par with them in terms of the extrinsic evaluation metrics.

	Sim. Corr.	POS	NER	UAS	LAS
`RI`	0.2	95.4	93.6	90.5	88.2
`HPCA`	0.2	96.3	96.2	90.7	88.6
`CBOW`	0.5	96.3	96.2	**92.1**	**90.1**
`SGRAM`	**0.6**	**96.4**	97.2	**92.1**	**90.0**
`GloVe`	0.5	95.2	**97.4**	91.9	89.9
`PWE`	0.5	96.0	96.4	91.9	89.9

Table 1: The comparison between principal word embeddings (`PWE`) and other sets of word embeddings. `CBOW` and `SGRAM` are two variants of `word2vec`. The results of POS tagging are obtained from sections 19–21 of WSJ. The results of NER are obtained from the `testa` data set from CoNLL-2003 shared task (Tjong Kim Sang and De Meulder, 2003). UAS and LAS stand for the unlabelled and labelled attachment scores respectively. The parsing results are obtained on the test set of WSJ.

	`RI`	`HPCA`	`GloVe`	`PWE`
Sec.	**180**	480	8040	900

Table 2: The amount of time (seconds) required by each of the word embedding methods to perform the dimensionality reduction. `PWE` refers to the principal word embedding method introduced in this paper.

References

Ali Basirat and Joakim Nivre. 2017. Real-valued syntactic word vectors (RSV) for greedy neural dependency parsing. In *Proceedings of the 21th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 20–28.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750.

Ronan Collobert, Jason Weston, Léon Bottou, Michael

Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Manaal Faruqui and Chris Dyer. 2014. Community evaluation and exchange of word vectors at wordvectors.org. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 19–24.

Gene H. Golub and Charles F. Van Loan. 1996. *Matrix Computations*. Johns Hopkins University Press, third edition.

N. Halko, P. G. Martinsson, and J. A. Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288.

I.T. Jolliffe. 2002. *Principal Component Analysis*. Springer Series in Statistics. Springer-Verlag.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709.

Thomas K Landauer and Susan T Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

Rémi Lebret and Ronan Collobert. 2014. Word embeddings through Hellinger PCA. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 482–490.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185.

Kevin Lund and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at the International Conference on Learning Representations (ICLR)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Tomas Mikolov, Wentau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Neha Nayak, Gabor Angeli, and Christopher D. Manning. 2016. Evaluating word embeddings using a representative suite of practical tasks. In *Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP*, pages 19–23.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Magnus Sahlgren. 2006. *The Word-space model*. Ph.D. thesis, Stockholm University.

Hinrich Schütze. 1992. Dimensions of meaning. In *Proceedings of the 1992 ACM/IEEE conference on Supercomputing*, pages 787–796.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, pages 142–147.

Linguistic explorations in real-valued syntactic word vectors (RSV)

Marc Tang, Ali Basirat

Department of Linguistics and Philology
Box 635, 751 26 Uppsala, Sweden
marc.tang@lingfil.uu.se, ali.basirat@lingfil.uu.se

Abstract

We study the presence of information provided by word embeddings from real-valued syntactic word vectors for determining the grammatical gender of nouns in Swedish. Our investigation reveals that regardless of being a frequently used word or not, real-valued syntactic word vectors are highly informative for identifying the grammatical gender of nouns. By using a neural network classifier we show that the uncertainty involved in the output of the network is only weakly correlated with the frequency level of words. Moreover, a linguistic analysis of errors demonstrates that while half of the errors can be avoided by using POS tag of words, the remaining errors are linguistically motivated and require extra information about the context of words.

1. Introduction

Word embedding is one of the fundamental techniques used to facilitate natural language processing tasks such as the use of neural networks (Pennington et al., 2014). Word embedding methods are based on the distributional hypothesis: *words that occur in the same contexts tend to have similar meaning* (Sahlgren, 2006). A word embedding method takes a raw (or annotated) corpus as input to count the frequency of seeing words in different contexts. The count data is stored in a co-occurrence matrix, whose rows and columns correspond to contexts and words. Word embeddings use a low-rank approximation of this matrix to associate each word in the corpus with a vector so that word similarities are reflected through vectors similarities.

Previous studies show that word embeddings capture syntactic and semantic regularities in languages and can be successfully applied to different NLP tasks (Kutuzov et al., 2016; Avraham and Goldberg, 2017). In terms of linguistics, however, it is still not very clear what type of information is encoded into word embeddings and how it affects the performance of word embeddings. As an example, Basirat and Tang (2018) proposed a classification framework that takes a set of word embeddings as input and predicts linguistically motivated semantic and morphosyntactic classes of words associated with the word embeddings in Swedish, e.g., count/mass, common/proper, uter/neuter. Surprisingly, the classification accuracy of grammatical gender (uter/neuter), which was expected to be high since the gender of a noun should be predictable from its co-occurrence statistics (e.g., neuter nouns tend to co-occur with determiners and adjectives in the neuter inflection), was not as good as expected (93.6%) and was even lower than the classification accuracy of common/proper nouns (95.2%).

In order to provide a better understanding of the information captured by word embeddings from a linguistic point of view, we further analyze errors made by the classification framework of Basirat and Tang (2018) during the classification of Swedish nouns based on their grammatical gender, c.f., *ett stor-t äpple* (SG.NEUT big.SG.NEUT apple.SG.NEUT) 'A big apple' and *en stor-∅ häst.* (SG.UTER big.SG.UTER horse.SG.UTER) 'A big horse'. This analysis of miss-classified nouns enables us to compare the knowledge provided by linguistic theories and the information encoded into word embeddings. In addition, we study the relationship between word frequencies and the errors made by the classifier by performing a regression analysis that relates word frequencies to the degree of uncertainty of the classifier in predicting the grammatical gender of words (i.e., is the classifier less certain in predicting the grammatical gender of less frequent words).

2. Materials and method

A corpus of Swedish raw sentences is extracted from the Swedish Language Bank Språkbanken and includes the Swedish Wikipedia at Wikipedia Monolingual Corpora, Swedish web news corpora (2001-2013), and the Swedish Wikipedia corpus (6×10^8 tokens in total). Information on nouns are based on the Swedish Associative Thesaurus version 2. The OpenNLP sentence splitter and tokenizer are used for normalization and nouns with a frequency lower than 100 are excluded due to the high ratio of compounds in Swedish (Östling and Wirén, 2013; Ullman and Nivre, 2014). The filtered list contains 15,002 uter nouns (70.89%) and 6160 neuter nouns (29.11%) in the dictionary and 174,538 unique words in the corpus. The unbalanced distribution between uter and neuter nouns is equally represented among high and low frequency words (Figure 1) with a standard deviation of 1.35%.

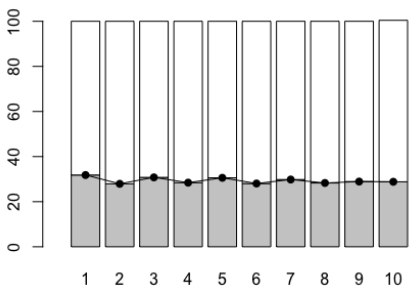


Figure 1: Distribution of uter (white) and neuter (gray) nouns with regard to frequency. The y-axis indicates the total ratio. The x-axis represents the nouns of the corpus partitioned into ten groups by their descending frequency

Word vectors are generated by Real-valued Syntactic Word Vectors (RSV) (Basirat and Nivre, 2017) and fed to a feed-forward neural network, which is used as the classifier. The parameters of the model are set as window size one with asymmetric-backward window type. The dimensionality is fixed at 50 to represent a balance between processing time and precision (Melamud et al., 2016). Other types of word vectors (e.g., Glove) generated similar results as RSV. We only report RSV word vectors due to space limitation. We do not provide an extensive methodological description since our focus is to analyze the output generated by previous studies. For further details on the settings and structure of the model with RSV, please refer to Basirat and Tang (2018).

3. Results

The overall performance of the classification task, described above, is assessed based on the *Precision* and *Recall* of the neural classifier. Precision evaluates how many tokens are correct among all outputs of the classifier, while Recall quantifies how many tokens are correctly retrieved among all expected correct outputs. The two measures are then merged into the F-score, which is equal to the harmonic mean of the precision and recall, i.e., $2(\text{Recall} * \text{Precision}/(\text{Recall} + \text{Precision}))$. As displayed in Table 1, the values of precision and recall, along with the final F-score are all higher for uter nouns. These numbers show that neuter nouns were harder to identify and represented more difficulty for classification both in terms of positive predictive value and sensitivity.

Table 1: The performance of neural network on grammatical gender prediction

	PRECISION	RECALL	F-SCORE
Neuter	88.70%	84.16%	86.37%
Uter	93.34%	95.40%	94.36%
Overall	91.98%	92.12%	92.03%

To visualize how neural network conceives gender of nouns in Swedish, we plot the spatial representation generated by neural network in Figure 2.

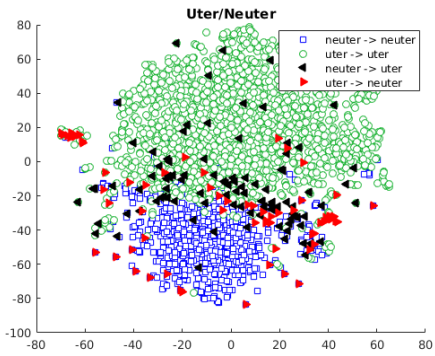
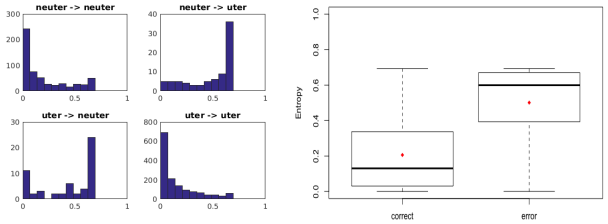


Figure 2: tSNE representation of the word vectors classified by neural network according to their grammatical gender

Such visualization is obtained by reducing the 50 dimensions to a two-dimensional representation using tSNE (Maaten and Hinton, 2008). First, this two-dimensional space reflects the unbalanced distribution between uter and neuter nouns (70.89% and 29.11%) as the cluster formed by uter nouns (green) outsize the agglomeration of neuter nouns (blue). Second, uter and neuter nouns are mostly scattered in two different areas, which implicates that they can be distinguished according to semantic and/or syntactic features of the language. Third, most of the errors were neuter nouns misinterpreted as uter nouns (black triangle).

We equally need to evaluate the confidence level of the model along with its performance. Figure 3a shows the histogram of the entropy (i.e., uncertainty) from the output of neural network. High values of entropy can be interpreted as more uncertainty in the classifier's outputs, which shows the weakness of the information provided by the word vectors with regard to grammatical gender. The most left and right histogram displays a left-oriented skewness. The neural network was thus relatively confident when classifying correctly the nouns according to their gender. Moreover, the erroneous output of neural network are skewed toward the right. the neural network was uncertain when classifying certain nouns, which resulted in a false identification of gender.



(a) Histogram (b) Box plot

Figure 3: Overview of the entropy in correct and erroneous outputs of neural network with regard to grammatical gender. In 3a, the y-axis indicates the amount of words from the test set, whereas the x-axis refers to the entropy

As shown in Figure 3b, the mean and median entropy of the errors (0.50) is much higher than the mean entropy of the correct outputs (0.20) at a statistically significant level as the non-parametric *approximative two-sample Fisher-Pitman permutation test* indicates a strong negative correlation ($z = -16.6$, $p < 0.001$). The entropy is thus representative of the models performance and demonstrate that the neural network based on word embeddings was interpreting the grammatical gender of nouns with high accuracy and confidence within our dataset, with exception to some outliers for which the entropy was unusually high.

4. Frequency and Entropy

While RSV word vectors encode syntactic and semantic regularities of language, we also need to investigate the magnitude of frequency effect on the performance of word embeddings in the classification task. We thus visualize in Figure 4 the general distribution of the entropy with respect to word frequency. If the accuracy of the neural network was purely based on word-frequency, we would expect high entropy for low-frequency word and vice-versa. The left-skewed pattern of tokens of errors apparently support such

hypothesis. However, we may equally find that most of the low-frequency words are also classified correctly by the neural network.

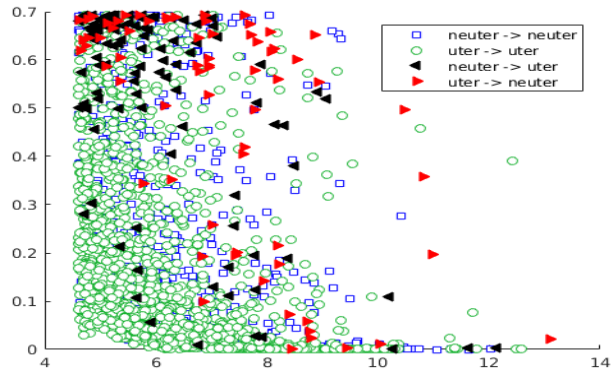


Figure 4: Distribution of the test set with regard to entropy and frequency. The y-axis indicates the entropy, while the x-axis refers to the natural logarithm of frequency.

Since our data does not fit with the conditions of bivariate normal distribution and homoscedasticity, we apply *Kendall's tau non-parametric correlation test*. The correlation between entropy and frequency is negative and statistically significant, but weak. Such statement is valid for the data in general (z = -25.395, *tau* = -0.3663, $p < 0.001$) along with the correct (z = -26.679, *tau* = -0.4011, $p < 0.001$) and erroneous output (z = -6.6165, *tau* = -0.3410, $p < 0.001$). The weak correlation between entropy and frequency is further shown by their non-linear monotonic association, i.e., the lines in Figure 5 show that the increase of frequency may include quite a large quantity of nouns without any significant decrease in terms of entropy.

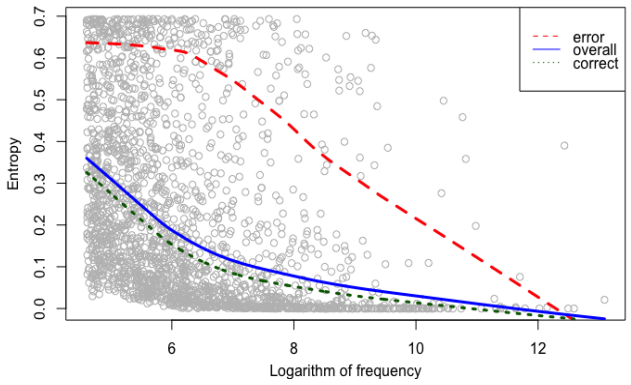


Figure 5: Correlation between entropy and frequency. The y-axis represents the entropy and the x-axis symbolizes the natural logarithm of frequency

However, after a certain level of frequency, the entropy drops relatively fast. The effect of frequency is small within the low-frequency nouns whereas a stronger effect size is observed within the high-frequency words. Moreover, following the assumptions of Zipf's law (Zipf, 1935), we observe that the majority of the nouns are found under the frequency logarithm of eight (86.65%, 1857/2143). Thus, a rerun of Kendall's tau test with solely the subset of nouns with frequency logarithm below eight illustrates that the correlation between entropy and frequency is less stronger within correct tokens (z = -20.419, *tau* = -0.3292 , $p < 0.001$) and even weaker with regard to the errors (z = -3.6542, *tau* = -0.2079, $p < 0.001$), as the *tau* coefficient decreases and the probability of the null hypothesis augments.

5. Error analysis

The performance of RSV word vectors combined with neural network (92.02%) is not as 'perfect' as anticipated since we expected that grammatical gender identification would be easily retrievable from gendered syntactic elements (e.g., determiners). Almost ten percent of errors show that word embeddings alone still face some difficulties. Our analysis (Table 2) shows that the errors can be categorized in the following three categories: noise, bare nouns, and polysemy.

Table 2: Errors of the neural network in the test set

CATEGORY	QUANTITY	RATIO	EXAMPLE
Noise	17	9.94%	
dictionary/corpus	11	6.43%	tidsplan
proper name	6	3.51%	rosengård
Bare noun	44	25.73%	
abstract noun	10	5.85%	fjärilsim
fixed usage	12	7.02%	pistolhot
mass	22	12.87%	fosfat
Polysemy	110	64.33%	
different gender	10	5.85%	vad
different POS	100	58.48%	kaukasiska
Total	171	100%	

The category of noise can be further divided into two sub-categories. First, a noun may be assigned to uter in the dictionary but be used with neuter within our corpus, and vice-versa. As an example, the noun *tennisracket* 'tennis racket' is affiliated to the uter gender. However, it occurs with neuter agreement in our corpora, e.g., *Han håller ett tennisracket i den ena handen och telefonluren i den andra.* (he hold.PRS one.NEUT tennis.racket in the.UTER one.UTER hand.DEF.UTER and handset.DEF.UTER in the.UTER other) 'He holds a tennis racket in one hand and the handset in the other.'. Furthermore, a minority of the noise originates from proper names that resemble common nouns by coincidence. As an example, the noun *rosengård* refers to a 'rose garden' as a neuter common noun. However, it may also refer to a location, which would not be affiliated to any grammatical gender, c.f., *Hon var en mycket omtyckt person i rosengård.* (she be.PAST one.UTER very loved person.UTER in Rosengård). 'She was a very popular person in Rosengård.'.

Nouns that mostly appear in bare form are classified to the gender that has the largest distribution in the language (i.e., uter) since word embeddings cannot retrieve sufficient cues in their surrounding context. For instance, *fjärilsim* 'butterfly (swimming)' is neuter but generally appears in bare form, e.g., *Hon simmar främst medley och fjärilsim.* (she swim.PRS mainly medley and butterfly) 'She mainly swims medley and butterfly.'. Nouns with a fixed usage

represents a similar difficulty, e.g., *pistolhot* 'gunpoint' is annotated as neuter but commonly occurs in the fixed construction *under pistolhot* 'at gunpoint'. Likewise for mass nouns, as they are uncountable and usually appear as definite form or bare noun, c.f., *fosfat* 'phosphate' and *Stora tillgångar på fosfat hade skapat en förmögenhet.* (large asset.PL on phosphate have.PAST create.PRF one.UTER fortune) 'Large assets on phosphate had created a fortune.'.

In cases of polysemy, a noun can have one form but different meanings with different gender. By way of illustration, *kaffe* can refer to 'coffee' as a mass, which is neuter. Nonetheless, 'coffee' can also be referred to via the uter gender if it refers to the abbreviation of 'a cup of coffee', c.f., *kaffet* (coffee.DEF.NEUT) 'the coffee' and *en kaffe* (one.UTER coffee) 'a coffee'. Polysemy may also involve a unique word form that relates to two different meanings with distinct parts of speech. For instance, *flyttande* 'moving' can serve as an adjective or a noun, c.f., *Området är särskilt viktigt som rastplats för flyttande gäss och änder.* (area.DEF.NEUT be.PRES particularly.NEUT important.NEUT as resting.place for moving goose.PL.INDF and duck.PL.INDF) 'The area is particularly important as a resting place for moving geese and ducks.' and *Jag var så trött på flyttandet att inget blev ordentligt.* (I be.PAST so tired on moving.DEF.NEUTER that none become.PAST properly) 'I was so tired of moving that nothing was going well.'.

6. Conclusions

We have studied the presence of information in RSV word vectors about the grammatical gender of Swedish nouns. A simple feed-forward neural network architecture has been used for this aim. The classifier takes RSV word vectors as input and predicts the grammatical genders as output. We consider the performance of the classifier as an indicator of the presence of information. Based on our experimental results, to a large extent, the information about the grammatical gender of Swedish nouns is encoded into RSV word vectors. Moreover, the performance of RSV word vectors was only weakly correlated to the frequency of the words, which indirectly supports the presence of semantic and syntactic information in word embeddings.

The errors generated by the classifier were related to noise in the raw data or cases of polysemy. Polysemy across different POS tags and noise in the data can be partially resolved by the use of a POS tagger and avoiding case normalization. However, this would lead to additional computational costs to generate word embeddings. Polysemy across gender seems more complicated and is related to linguistic theories of gender assignment: Swedish neuter nouns are generally mass nouns (Dahl, 2000), which frequently undergo conversion between different part of speech categories (Gillon, 1999). Uter nouns, on the other hand, were affiliated to the correct gender with higher accuracy, which may be due to the fact that most uter nouns are animate and countable nouns that rarely occur as bare nouns. Therefore, extra information about the contextual environment of words can increase the accuracy of the grammatical gender classifier.

References

Oded Avraham and Yoav Goldberg. 2017. The interplay of semantics and morphology in word embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 422–426. Association for Computational Linguistics.

Ali Basirat and Joakim Nivre. 2017. Real-valued Syntactic Word Vectors (RSV) for Greedy Neural Dependency Parsing. In *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa*, pages 21–28, Gothenburg. Linköping University Electronic Press.

Ali Basirat and Marc Tang. 2018. Lexical and morpho-syntactic features in word embeddings - A case study of nouns in swedish. In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018)*, pages 663–674.

Östen Dahl. 2000. Elementary gender distinctions. In Barbara Unterbeck and Matti Rissanen, editors, *Gender in grammar and cognition*, pages 577–593. Mouton de Gruyter, Berlin.

Brendan S. Gillon. 1999. The lexical semantics of English count and mass nouns. In Evelyne Viegas, editor, *Breadth and depth of semantic lexicons*, pages 19–37. Springer, Dordrecht.

Andrey Kutuzov, Erik Velldal, and Lilja Øvrelid. 2016. Redefining part-of-speech classes with distributional semantic models. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 115–125. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Oren Melamud, David McClosky, Siddharth Patwardhan, and Mohit Bansal. 2016. The role of context types and dimensionality in learning word embeddings. *arXiv*, pages 1–11.

Robert Östling and Mats Wirén. 2013. Compounding in a Swedish blog corpus. *Acta Universitatis Stockholmiensis*, pages 45–63.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Magnus Sahlgren. 2006. *The Word-space model*. Ph.D. thesis, Stockholm University.

Edvin Ullman and Joakim Nivre. 2014. Paraphrasing Swedish Compound Nouns in Machine Translation. In *MWE@ EACL*, pages 99–103.

George K Zipf. 1935. *The psycho-biology of language*. MIT Press, Cambridge.

An Evaluation of Neural Machine Translation Models on Historical Spelling Normalization

Gongbo Tang, Fabienne Cap, Eva Pettersson, Joakim Nivre

Department of Linguistics and Philology, Uppsala University
firstname.lastname@lingfil.uu.se

Abstract

In this paper, we apply different NMT models to the problem of historical spelling normalization for five languages: English, German, Hungarian, Icelandic, and Swedish. The NMT models are at different levels, have different attention mechanisms, and different neural network architectures. Our results show that NMT models are much better than SMT models in terms of character error rate. The vanilla RNNs are competitive to GRUs/LSTMs in historical spelling normalization. Transformer models perform better only when provided with more training data. We also find that subword-level models with a small subword vocabulary are better than character-level models for low-resource languages. In addition, we propose a hybrid method which further improves the performance of historical spelling normalization.

1. Introduction

The processing of historical texts is attracting more and more interest. However, in contrast to modern text, historical text processing faces more challenges. First, there is little annotated data for training a model, which leads to data sparsity issues when using statistical methods. Second, there are a lot of variations in historical texts from different time periods, not only in spelling but also in lexical semantics and syntax. Therefore, the NLP tools developed for modern text cannot be used for these historical texts directly. Spelling normalization is the task of mapping a historical spelling to its modern spelling. It is usually used as a preprocessing step before feeding the historical text into modern NLP tools which leads to much better results compared to analyzing unnormalized historical texts.

There are some papers in which neural machine translation (NMT) models are employed for the spelling normalization task. But the evidence so far is too incomplete to draw any general conclusions about the utility of different NMT models for historical spelling normalization. We are interested in exploring how different properties of NMT models interact with different aspects of the spelling normalization problem and find some generalizations about the use of NMT models for this task.

In this paper, we apply different NMT models to normalize spellings for historical stages of five languages, English, German, Hungarian, Icelandic, and Swedish.

2. NMT Models

When we apply NMT models to the historical spelling normalization task, the first research question is which NMT model is most suitable for this task.

Based on the features of historical spellings and NMT models, we first give four hypotheses about NMT models for spelling normalization:

Hypothesis 1 The performance gap between vanilla recurrent neural networks (RNNs) and gated recurrent units (GRUs)/long short-term memory units (LSTMs) is small. In contrast to conventional NMT models, the historical and modern token pairs are our training data instead of parallel sentence pairs. In our experiments, the average token length varies from 4 to 6, which means that we build the model on much shorter sequences. The long-distance problem will be alleviated.

Hypothesis 2 The gap between NMT models with attention and without attention is also small. Since the average token length is only around five, additionally paying attention to all the tokens in the source sequences may be unnecessary. Thus, we hypothesize that the decoder in the vanilla Encoder-Decoder model can predict most of the targets correctly with only one fixed-size vector from the encoder, even without any attention mechanisms.

Hypothesis 3 Transformer models perform better than soft-attention-based models. Transformer models have more advanced self-attention networks and more fine-grained multi-head attention mechanisms compared to RNN-based models with soft-attention. Thus, Transformer models have better performance in conventional translation tasks. We hypothesize that it is the same in the spelling normalization task.

Hypothesis 4 Subword-level NMT models perform better than character-level NMT models. Character-level and subword-level models are proposed to deal with the problem of out-of-vocabulary words mainly, and subword-level NMT models usually outperform character-level models.

To test our hypotheses, we explore 8 different NMT models which are described in Table 1.

Name	Level	Attention	Architecture
NoAtt-RNN	character	no	RNN
NoAtt-GRU			GRU
NoAtt-LSTM			LSTM
Att-RNN		soft	RNN
Att-GRU			GRU
Att-LSTM			LSTM
Transformer		multi-head	Self-attention
BPE-Soft	subword	soft	LSTM

Table 1: NMT models for the spelling normalization task. *RNN* means vanilla RNNs.

Language	Training	Development	Test	Unchanged	Token	Char	Max	Avg
English	148,852	16,461	17,791	75.8	22,302	102	22	4.16
German	39,887	5,418	5,005	84.4	11,521	100	27	4.74
Hungarian	137,669	17,181	17,214	17.1	69,624	128	27	5.91
Icelandic	52,440	6,443	6,384	50.5	14,845	89	16	4.14
Swedish	28,327	2,590	33,544	64.6	11,129	92	36	4.55

Table 2: Statistics of the datasets. The figures in *Training*, *Development*, and *Test* are the numbers of token pairs. The *Unchanged* (%) means the rate of unchanged spellings in the test set. *Token* and *Char* show the token and the character vocabulary sizes in the training set. *Max* and *Avg* show the max length and average length of token in the training set. All counts are based on case-sensitive data.

3. Experimental Setup

All the datasets[1] are exactly the same as the parallel datasets for the statistical machine translation (SMT) models in Pettersson et al. (2014), which are described in Table 2. The datasets consist of a list of token pairs, which have one historical spelling and the corresponding modernized spelling. Some illustrative English examples are given in Table 3.

Historical	citee	gyve	gyf	late
Modern	city	give	give	late

Table 3: Token pair examples in English.

All the models are trained by the same toolkit, Marian (Junczys-Dowmunt et al., 2018). For subword-level models, we utilize the Byte Pair Encoding (BPE) method (Sennrich et al., 2016) to generate subword units.

The vanilla RNN chooses the "tanh" RNN cell. We enable "mini-bach-fit" which automatically choose the mini-batch size for the given "workspace" size, and the "workspace" is set to 7500. We use *Adam* as the optimizer. The learning rate is set to 0.0003, but we set the warmup steps to 16,000, which means that the learning rate increases linearly before 16,000 steps. A model checkpoint is saved every 500 updates. The evaluation metrics on the development set are cross-entropy and perplexity. We set the early stopping patience to 8 checkpoints. All the neural networks have 6 layers. The size of embeddings is 512. We tie the target embeddings and the output embeddings in the output layer. We use the checkpoint that achieves the best perplexity to generate the normalizations. We set the beam size to 5 during decoding.

4. Results

The baseline from Pettersson et al. (2014) has very high word accuracy and low character error rate (CER) scores in all five languages. The results in the baseline are obtained using character-level SMT models except for Icelandic, where the combination of a Levenshtein-based method and a dictionary-based method achieved the best results. We use word accuracy and CER to evaluate the predictions. In our experiments, we use Levenshtein distance to compute CER. Table 5 gives the detailed results of different models.

[1] http://stp.lingfil.uu.se/histcorp/tools.html

4.1 Word Accuracy

Table 5 shows that NMT models outperform SMT models in four out of five languages, except for Swedish, when we use word accuracy as the evaluation metric. Compared to the other four languages, we get a huge absolute improvement of 12.04% in Hungarian, improving the word accuracy from 80.1% to 92.14%. Our best NMT result in Swedish is still a little lower than the baseline in word accuracy. We attribute the reason to the dataset size, because Swedish has the smallest training set.

We divide the incorrectly normalized spellings into three groups by checking the normalizations of the test set:

1. *Change*: modern spelling is identical to historical spelling, but the model normalized the historical spelling to another spelling.
2. *Copy*: modern spelling is different from historical spelling, but the model copied the historical spelling as the normalization.
3. *Other*: other types of error.

	EN	DE	HU	IS	SE
Change	22.3	28.5	6.1	33.8	25.0
Copy	22.7	41.7	6.1	20.8	23.6
Other	55	29.8	87.8	45.4	51.4

Table 4: Error distributions (%).

Table 4 gives us the error distributions of the best model in each language. There are a lot of *Change* and *Copy* errors. This finding reveals that it is a little bit difficult for the NMT model trained on the data that mixed with changed and unchanged spellings to normalize unchanged spellings.

Therefore, we explore a hybrid method, combining the NMT-based method and the dictionary-based method. More specifically, we first extract a list of unchanged spellings from the training set. During the evaluation, if a word is in this list, we simply copy it as its normalization. If it is not in the list, we feed it to the NMT models. The results in column "△" of Table 5 show that this hybrid method improves the accuracy further. In particular, the improvements on Icelandic are around 5%.

4.2 CER

With the CER measure, we calculate the number of correctly normalized characters, without considering the word

	English			German			Hungarian			Icelandic			Swedish		
	Acc	CER	△	Acc	CER	△	Acc	CER	△	Acc	CER	△	Acc	CER	△
Baseline	94.3	0.07	-	96.6	0.04	-	80.1	0.21	-	84.6	0.19	-	**92.9**	0.07	-
NoAtt-RNN	94.73	0.02	1.19	94.89	0.02	0.90	90.99	0.03	0.82	86.73	0.05	5.18	91.44	0.03	0.39
NoAtt-GRU	94.79	0.02	1.14	94.85	0.02	0.60	91.03	0.03	0.84	86.98	0.05	4.71	91.34	0.03	0.39
NoAtt-LSTM	94.61	0.02	1.20	95.78	0.02	0.64	90.91	0.03	0.83	86.61	0.05	5.17	91.29	0.03	0.41
Att-RNN	94.69	0.02	1.21	94.23	0.02	0.70	91.69	0.02	0.77	**87.59**	0.04	4.95	91.56	0.03	0.38
Att-GRU	94.80	0.02	1.19	94.83	0.02	0.66	91.68	0.02	0.82	87.17	0.05	5.08	91.68	0.03	0.36
Att-LSTM	94.85	0.02	1.17	96.00	0.02	0.44	91.57	0.03	0.78	86.83	0.05	4.93	91.72	0.03	0.36
Transformer	**95.16**	0.02	1.17	95.22	0.02	0.48	**92.14**	0.02	0.80	86.45	0.05	5.15	88.99	0.05	0.49
BPE-Soft	95.02	0.02	1.18	**96.64**	0.01	0.32	91.96	0.03	0.78	87.19	0.03	4.95	91.21	0.03	0.35

Table 5: Evaluation results in word accuracy (Acc, %) and character error rate (CER). "△" denotes the absolute improvement in accuracy (%) of combining the NMT-based method and the dictionary-based method.

	English	German	Hungarian	Icelandic	Swedish
Historical	alys	julius	vètē	uopn	sielffuer
Normalized	alis	jiues	vetem	opnu	själver
Modern	alice	julius	vetém	vopn	själv
Historical	wett	cohærentz	haila	sier	herrskaper
Normalized	wit	cohaerenz	hajola	sjer	herrskaper
Modern	know	kohärenz	hajla	sér	herrskapen

Table 6: Some incorrectly normalized examples from the development set.

level. CER is similar to the BLEU score in MT, and we evaluate at sub-sequence-level rather than the overall accuracy. When we use CER as the evaluation metric, NMT models get the best results for all five languages, even though some models achieve lower accuracy than the baseline. This result is different from the result of Korchagina (2017). In her paper, if the SMT models are better than the NMT models in word accuracy, these SMT models are better than the NMT models in CER as well. We assume that this may be due to different neural network architectures: they use CNNs while we use RNNs and self-attention networks.

	Changed	Incorrect
English	1.45	1.81
German	1.07	1.64
Hungarian	2.58	1.78
Icelandic	1.41	1.64
Swedish	1.32	1.54

Table 7: The average edit distance of the changed spellings in test set and the average edit distance of the incorrectly normalized changed spellings.

Table 7 shows the edit distance of spellings. For the incorrectly normalized changed spellings, the average edit distance is smaller than 2. In other words, we just need less than two edits to translate an incorrectly normalized spelling into the correct one. In the incorrect normalizations, Swedish has the shortest average edit distance 1.54, and English has the longest average edit distance 1.81.

Table 6 gives some incorrectly normalized examples from the development set. Most of the edit distances of spellings are longer than 1. In addition to *Change* and *Copy* errors, some historical spellings are quite different from their modern spelling, such as "wett" in English. For the historical word "wett", it is extinct, people just mapped a semantic related modern word to it. "know" has no relations with "wett" in spelling and pronunciation. Characters with different accents also cause mistakes easily. For example, "vetém" in Hungarian and "sér" in Icelandic.

4.3 NMT versus SMT

In the conventional MT tasks, NMT models usually outperform SMT models. The first reason is that the dense embeddings in NMT are powerful representations. The second reason is that NMT models usually consider a larger context compared to SMT models. This is the same in historical spelling normalization. In our experiments, the most obvious example is Hungarian. The absolute improvement is 12.04% in word accuracy. Compared to other languages, Hungarian has the largest token and character vocabularies and the highest changed rate. It also has the longest average token length.

However, in terms of accuracy, it is still hard for NMT models to exceed SMT models in Swedish. We also find that the performance of NMT models is quite close to the baseline in German which has the second smallest training dataset. We hypothesize that the size of training data is crucial for NMT models to exceed SMT models.

4.4 Different NMT Models

Hypothesis 1 is that the performance gap between vanilla RNNs and GRUs/LSTMs will not be huge. The results in Table 5 reveal that the vanilla RNNs are competitive to the GRUs/LSTMs in this task. These results support our Hypothesis 1 well.

Hypothesis 2 states that NMT models with and without attention will not differ a lot. The models with attention are

Model	BPE-size	English	German	Hungarian	Icelandic	Swedish
BPE-Soft	0	94.85	96.00	91.57	86.83	**91.72**
	100	**95.02**	**96.64**	91.87	**87.19**	91.21
	200	94.91	96.28	91.81	86.89	91.13
	300	94.69	96.50	**91.96**	86.76	90.84
	500	94.54	96.42	91.52	86.51	90.57
	1,000	94.52	96.18	91.44	86.29	89.67
	5,000	93.71	95.06	89.43	84.87	85.47
BPE-Transformer	0	**95.16**	95.22	**92.14**	86.45	88.99
	100	94.21	95.66	90.14	**86.64**	90.07
	200	94.38	96.08	90.71	86.62	**90.17**
	300	94.26	**96.10**	90.87	86.33	89.76

Table 8: Accuracy (%) with different BPE vocabulary sizes. "0" represents the character-level models.

slightly better than models without attention in our experiments, which is in line with the results in Bollmann et al. (2017). However, the gap is quite small. Thus, it fits our Hypothesis 2.

Hypothesis 3 is that Transformer models are better than soft-attention-based models. From Table 5, we can see that *Transformer* achieves higher word accuracy in English and Hungarian compared to soft-attention-based models. It is interesting that English and Hungarian have much more training data compared to the other three languages. This result reveals that Transformer models need more data to exceed RNN-based models.

Hypothesis 4, states that subword-level models are better than character-level models. Our experimental results of *BPE-Soft* models in four languages (except Swedish) show that subword-level models are superior to character-level models when the BPE vocabulary is small.

Many historical spellings only have several instances in the training set. The NMT model cannot translate the token well at the token level. Moreover, there is also a data sparsity problem for the subwords when we set a larger BPE vocabulary. We assume that BPE maybe cannot learn rare subword units very well, because of the data sparsity. We find that the subword-level models perform worse than character-level models when the BPE vocabulary is larger than 300 in all five languages.

We further train Transformer models at subword-level which are called *BPE-Transformer* in Table 8. In German, Icelandic, and Swedish where the data size is small, the subword-level models surpass character-level models. However, the subword-level models in English and Hungarian are clearly not as well as character-level models.

Historical languages which have little training data are considered as low-resource languages, especially the German, the Icelandic, and the Swedish in this paper. Hence the result of Hypothesis 4 can be interpreted as that subword-level models with a small subword vocabulary can further improve the performance compared to character-level models in low-resource languages.

5. Conclusions

In summary, our work can be concluded as follows:

- We evaluate different NMT models on historical spelling normalization in a multilingual setting.
- We find that NMT models are better than SMT models considering CER.
- We show that vanilla RNNs are competitive to GRUs/LSTMs.
- We demonstrate that Transformer models perform better when provided with more training data.
- We reveal that models with a small subword vocabulary are better than character-level models for low-resource languages.

References

Marcel Bollmann, Joachim Bingel, and Anders Søgaard. 2017. Learning attention for historical text normalization by learning to pronounce. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 332–344, Vancouver, Canada. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. *arXiv preprint arXiv:1804.00344*.

Natalia Korchagina. 2017. Normalizing medieval german texts: from rules to deep learning. In *Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language*, number 133, pages 12–17, Gothenburg, Sweden. Linköping University Electronic Press.

Eva Pettersson, Beáta Megyesi, and Joakim Nivre. 2014. A multilingual evaluation of three spelling normalisation methods for historical text. In *Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 32–41, Gothenburg, Sweden. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Unsupervised pre-training of a neural network for detecting healthcare-acquired infections

Claudia Figueras, Rebecka Weegar

Stockholm University
Department of Computer and Systems Sciences, Stockholm University, Stockholm, Sweden
claudiafiguerasj@gmail.com, rebeckaw@dsv.su.se

Abstract

Healthcare-acquired infections (HAIs) are a worldwide problem, causing high morbidity and mortality rates, as well as increased costs. Therefore, the development of a computerised surveillance system that detects these infections from patient records could help to reduce their prevalence, to provide earlier treatment and to lessen the workload for clinicians. This study aims at improving previous results on the use of deep learning in Swedish electronic patient records (EPRs) to detect HAIs using unsupervised pre-training. Unsupervised pre-training was performed by applying `doc2vec` to a manually compiled dataset using EPRs from the Swedish Health Research Bank. Afterwards, the produced embeddings were used to classify for HAI or non-HAI annotated EPRs from the Stockholm EPR Detect-HAI corpus using a recurrent neural network. The results of using the embeddings produced by `doc2vec` and applying a recurrent neural network yielded an accuracy of 0.58, precision of 0.61, recall of 0.71, AUC of 0.83 and AUPRC of 0.91. While the overall results did not improve the results of previous studies, they showed the potential of unsupervised pre-training when small annotated and large unannotated datasets are available.

1. Introduction

Healthcare-acquired infections (HAIs), or nosocomial infections, are infections that occur to a patient in a healthcare facility or other clinical settings and that were not existing when the patient was admitted, including infections that occur during the hospital stay, after the discharge and among the healthcare facility staff (World Health Organization, 2002). HAIs are a worldwide problem that cause more extended hospital stays, long-term disabilities, increased microbial drug resistance, enormous costs for health care systems and for patients and their relatives and avoidable deaths (World Health Organization, 2015). In Sweden it was estimated that HAIs prolong the hospital stay of a patient by four days on average and affects 10% of all inpatients, risking patients health status with higher morbidity and mortality rates (CDC, 2015; Ehrentraut et al., 2016; Jacobson and Dalianis, 2016).

Several studies have attempted to mitigate this issue by developing systems that detect or predict HAIs by applying different machine learning techniques on Electronic Patient Records (EPRs) written in Swedish (Ehrentraut et al., 2012; Ehrentraut et al., 2014; Tanushi et al., 2014; Ehrentraut et al., 2016; Jacobson and Dalianis, 2016). Particularly, one of these studies (Jacobson and Dalianis, 2016), applied deep learning with the purpose of predicting HAIs in a dataset consisting of Swedish EPRs but this approach obtained poorer results than using other traditional machine learning methods applied to the same dataset (Ehrentraut et al., 2014; Ehrentraut et al., 2016). One possible cause of the lower results in the Jacobson and Dalianis (2016) study is that the data they used might be insufficient, as it only contained 213 health records. Thus, this study was an attempt to improve these previous results by using a method to cope with the scarcity of data: unsupervised pre-training.

Unsupervised pre-training consists of pre-training some layers of a neural network (NN) using unannotated data with an unsupervised learning algorithm to set the stage, initializing the weights and other parameters of the network. Then, these parameters will be used together with a supervised learning algorithm to train the NN (Erhan et al., 2010). Unsupervised pre-training has previously shown significant and valuable results when too little annotated data is available, allowing for a higher robustness of the NN as well as improved generalizations to unseen events (Bengio et al., 2007; Ranzato et al., 2007; Erhan et al., 2010). In case of text, as in this project, unsupervised pre-training consisted of creating word embeddings using unannotated data. The embeddings were then used by the NN in a supervised way to classify medical records as HAI or non-HAI. Here, a recurrent NN (RNN) was chosen as it preserves the structural information of sentences and showed impressive results in previous language modelling studies (Mikolov and Kombrink, 2011; Auli and Gao, 2014).

2. Methods

In this project, two datasets from the Swedish Health Record Research Bank have been used. First, a manually compiled dataset (the unannotated dataset) was used in the unsupervised pre-training. Second, Stockholm EPR Detect-HAI Corpus (the annotated dataset) was used in the unsupervised pre-training and in the supervised training of the network[1]. The unannotated dataset consisted of 217,824 care episodes collected during 2009 and 2010. The annotated dataset consisted of 213 records collected during the spring of 2012 and classified as HAI or non-HAI by two domain experts. Details of the annotated dataset can be seen in Table 1.

[1]This research has been approved by the Regional Ethical Review Board in Stockholm (*Etikprövningsnämnden i Stockholm*) permission number 2012/1838-31/3.

	HAI	Non-HAI	Total
Number of health records	131	83	214
Time in hospital [days]	2-144	3-93	2-144
Number of tokens	1,034,760	230,226	1,264,986

Table 1: Table with principal features of Stockholm EPR Detect-HAI corpus. Adapted from Ehrentraut et al. (2016; Jacobson and Dalianis (2016).

2.1 Pre-processing

First, the unannotated dataset was generated by extracting EPRs from the Swedish Health Record Research Bank, trying to capture the same structure as the annotated dataset. For both datasets, the text was pre-processed before feeding the records into the NN. Removal of invalid characters, conversion into lowercase, stop words removal and conversion into word and document embeddings (see following subsection) were performed in both datasets.

2.2 `doc2vec` architecture

`doc2vec` is a computer algorithm that creates document and word embeddings (i.e. mapping from documents or words to numerical vectors). `doc2vec` is a term popularised by Gensim[2] and it is an extension of `word2vec` (Mikolov et al., 2013). It consists of a feedforward NN with one layer that outputs vector representations for text documents. The parameters used in `doc2vec` can be seen in Table 2. Both the unannotated and annotated datasets were fed into `doc2vec` to produce more meaningful vectors than by just using the annotated dataset. This part is the unsupervised pre-training stage.

`doc2vec` parameters	Values
Batch size	500
Vocabulary size	7500
Learning rate	0.001
Window size	3 words
Word embedding size	200
Document embedding size	100

Table 2: Table detailing the parameters and their values used during the unsupervised pre-training stage.

2.3 RNN architecture

The annotated dataset was fed to a RNN using the word and document embeddings produced by `doc2vec`. Specifically, the RNN was of dynamic type, had a size of 100 units, the activation function used was *tanh* and Softmax was used for the final the classification. A RNN was chosen over other networks (e.g. LSTM) because of its simplicity. The loss function was Sparse Softmax Cross Entropy [3] and the backpropagation algorithm, RMSProp. The dropout rate was set to 0.5 during training time, as suggested by previous authors (Srivastava et al., 2014).

The hyperparameters were fine-tuned using previous studies as reference (Erhan et al., 2010; McClure, 2017). Different values of learning rate (LR), epochs and batch sizes were tested in the different iterations, with LR values ranging from 0.005 to 0.5, epoch values of {20, 50, 100, 200} and batch sizes of {5, 10, 25, 50}.

The data was split into 60% training, 20% validation and 20% testing data and each data set was shuffled to randomise the order at each training pass.

3. Results

After several iterations, the model was trained with the hyperparameters selected according to the performance on the validation set and the final model was evaluated on the test set. The results can be seen in Table 3, and they were obtained using 200 epochs, 200 as batch size and a learning rate of 0.005.

Metric	Result
Loss	0.74
Accuracy	0.58
Precision	0.61
Recall	0.71
F-score	0.66
AUC	0.83
AUPRC	0.91

Table 3: Results after the final training of the NN using the best hyperparameter combination (200 epochs, 200 batch size and learning rate of 0.005).

4. Discussion

Precision, recall and F-score values were lower than the ones obtained by Jacobson and Dalianis (2016). Nonetheless, Jacobson and Dalianis (2016) did not provide the AUC (Area under the ROC curve) neither the AUPRC (Area Under the Precision Recall Curve), making the results more difficult to compare. In this project higher recall than precision was achieved, and this was preferred because obtaining many false negatives is costlier than many false positives. This is because it is safer for the patient to be misdiagnosed than not to be diagnosed at all (Petticrew and Sowden, 2001).

Overall, from these results, it is hard to determine if the unsupervised pre-training was an improvement of the NN or not. The results are promising and, even though they are not as good as previous studies in the field, the metrics show that the results are satisfactory, as the AUC and AUPRC values are high, meaning that it is a trustable test. The low precision and accuracy values could be attributed to many factors, such as limited amount of data, too simple NN (i.e. 100 units), the unannotated dataset not being informative enough or unclear relationship between input

[2] https://radimrehurek.com/gensim/

[3] Sparse Softmax Cross Entropy measures the likelihood error in discrete classification tasks in which the classes are mutually exclusive (like in this case with HAI and non-HAI).

data and output prediction (Alpaydin, 2014; Goodfellow et al., 2016).

In any case, it can be concluded that the NN model needs to be optimised before it can be used in a real setting. Future research may focus on using other kinds of NNs such as LSTMs, testing if different word embedding algorithms are more suitable than `doc2vec` or improving the quality of the data, for example by increasing the size of the annotated dataset, adding more information to the unannotated dataset or performing other pre-processing techniques.

Acknowledgements

This project would not have come into being without the guidance of my research supervisors MSc Rebecka Weegar and Professor Hercules Dalianis from Stockholm University. I would like to express my deep gratitude for their patient guidance, enthusiastic encouragement and useful feedback on this research work.

References

E. Alpaydin. 2014. *Introduction to Machine Learning. 3rd ed*. The MIT Press, Cambridge (Massachusetts).

M. Auli, J. Gao. 2014. Decoder Integration and Expected BLEU Training for Recurrent Neural Network Language Models. In *Conference of the Association for Computational Linguistics (ACL) 2014*, 136–142.

Y. Bengio, P. Lamblin, D. Popovici, H Larochelle. 2007. Greedy Layer-Wise Training of Deep Networks. *Advances in Neural Information Processing Systems*, 19(1):153.

Centers for Disease Control and Prevention. 2015. Healthcare-associated infections.

C. Ehrentraut, H. Tanushi, H. Dalianis, J Tiedemann. 2012. Detection of Hospital Acquired Infections in sparse and noisy Swedish patient records- A machine learning approach using Nave Bayes, Support Vector Machines and C4.5. In *Proceedings of the Sixth Workshop on Analytics for Noisy Unstructured Text Data (AND)*, Mumbai, India.

C. Ehrentraut, M. Kvist, E. Sparrelid, H. Dalianis. 2014. Detecting Healthcare-Associated Infections in Electronic Health Records - Evaluation of Machine Learning and Preprocessing Techniques. In *Proceedings of the 6th International Symposium on Semantic Mining in Biomedicine (SMBM 2014)*, Aveiro, Portugal.

C. Ehrentraut, M. Ekholm, H. Tanushi, J. Tiedemann, H. Dalianis. 2016. Detecting hospital- acquired infections: A document classification approach using support vector machines and gradient tree boosting. *Health Informatics Journal*, pages 1–19.

D. Erhan, Y. Bengio, A. Courtville, PA. Manzagol, P. Vincent. 2010. Why Does Unsupervised Pre-training Help Deep Learning? *Journal of Machine Learning Research*, 11:625–660.

I. Goodfellow, Y. Bengio, A. Courtville. 2016. *Deep Learning. Web ed.* The MIT Press.

O. Jacobson and H. Dalianis. 2016. Applying deep learning on electronic health records in Swedish to predict healthcare-associated infections. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 191–195. Berlin, Germany.

T. Mikolov, S. Kombrink. 2011. Extensions of recurrent neural network language model. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5528–5531.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, pages 3111–3119.

N. McClure. 2017. *TensorFlow Machine Learning Cookbook. 1st ed.* Packt Publishing Ltd, Birmingham, UK.

M. Petticrew, A. Sowden. 2001. False-negative results in screening programs. Medical, psychological, and other implications. *International Journal of Technology Assessment in Health Care*, 17(2):164–170.

M. Ranzato, FJ. Huang, YL. Boureau, Y. LeCun. 2007. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929-1958.

H. Tanushi, M. Kvist, E. Sparrelid. 2014. Detection of healthcare-associated urinary tract infection in Swedish electronic health records. *Studies in Health Technology and Informatics*, pages 330–339.

World Health Organization. 2002. G. Ducel, J. Fabry, L. Nicolle, editors, *Prevention of hospital-acquired infections. A practical guide. 2nd ed.*.

World Health Organization. 2015. *Healthcare-associated infections - Fact sheet.*

A Pronoun Test Suite Evaluation of the English–German MT Systems at WMT 2018

Liane Guillou[1*], Christian Hardmeier[2*], Ekaterina Lapshinova-Koltunski[3*], Sharid Loáiciga[4*]

[1]School of Informatics, University of Edinburgh
[2]Department of Linguistics and Philology, Uppsala University
[3]Department of Language Science and Technology, Saarland University
[4]CLASP, University of Gothenburg
lguillou@inf.ed.ac.uk christian.hardmeier@lingfil.uu.se
e.lapshinova@mx.uni-saarland.de sharid.loaiciga@gu.se

Abstract

We evaluate the output of 16 English-to-German MT systems with respect to the translation of pronouns in the context of the WMT 2018 competition. We work with a test suite specifically designed to assess the systems quality in various fine-grained categories known to be problematic. The main evaluation scores come from a semi-automatic process, combining automatic reference matching with extensive manual annotation of uncertain cases. We find that current NMT systems are good at translating pronouns with intra-sentential reference, but the inter-sentential cases remain difficult. NMT systems are also good at the translation of event pronouns, unlike the systems in the phrase-based SMT paradigm. No single system is best at translating all types of anaphoric pronouns, suggesting unexplained random effects influencing the translation of pronouns with NMT.

1. Introduction

Data-driven machine translation (MT) systems are very good at making translation choices based on the words in the immediate neighbourhood of the word currently being generated, but aspects of translation that require keeping track of long-distance dependencies continue to pose problems. Linguistically, long-distance dependencies often arise from discourse-level phenomena such as pronominal reference, lexical cohesion, text structure, etc. Initially largely ignored, such problems have attracted increasing attention in the statistical MT community in recent years (Hardmeier, 2012; Sim Smith, 2017). One important problem that has proved to be surprisingly difficult despite extensive research is the translation of pronouns (Hardmeier et al., 2015; Guillou et al., 2016; Loáiciga et al., 2017).

Since the invention of the BLEU score (Papineni et al., 2002), the MT community has measured progress to a large extent with the help of summary scores that are easy to compute, but strongly affected by the corpus-level frequency of certain phenomena and that tend to neglect specific linguistic relations and problems that occur infrequently. The advent of neural MT (NMT) with its improved capacity for modeling more complex relationships between linguistic elements has brought an increased interest in linguistic problems perceived as difficult, which are often not captured well by metrics like BLEU. It has been suggested that test suites composed of difficult cases could provide more relevant insights in the performance of MT systems than corpus-level summary scores (Hardmeier, 2015). In this paper, we present a semi-automatic evaluation of the systems participating in the English–German news translation track of the MT shared task at the WMT 2018 conference.

The analysis was carried out with the help of an English–German adaptation of the PROTEST test suite for pronoun translation (Guillou and Hardmeier, 2016). The test suite allows us to do a fine-grained evaluation for different types of pronouns. Whilst the translation of event pronouns, which caused serious problems in earlier evaluations of statistical MT systems (Hardmeier et al., 2015; Hardmeier and Guillou, 2018), seems to be handled fairly well by modern NMT systems, we find that translating anaphoric pronouns is still difficult, especially (but not only) if the pronoun has an antecedent in a different sentence.

2. Test Suite Construction

We constructed a test suite of 200 pronoun translation examples for English–German with a focus on the English pronouns *it* and *they* and the aim of providing a set of examples that represents the different problems machine translation researchers should consider. The examples were extracted from the TED talk portion of ParCorFull (Lapshinova-Koltunski et al., 2018), an English–German parallel corpus manually annotated for full co-reference.

The selection is based on a two-level hierarchy which considers pronoun *function* at the top level, followed by other attributes of the pronouns at the more granular lower level (for anaphoric pronouns only).

The English pronoun *they* functions as an anaphoric pronoun, whereas *it* can function as either an anaphoric (1), pleonastic (2), or event reference[1] pronoun (3), with each function requiring the use of different pronouns in German.

(1) a. The infectious disease that's killed more humans than any other is malaria. **It**'s carried in the bites of infected mosquitos.
b. Jene Krankheit, die mehr Leute als jede andere umgebracht hat, ist Malaria gewesen. **Sie** wird über

[*]All authors contributed equally.

[1]*Event* reference is more commonly known as abstract anaphora or discourse deixis.

die Stiche von infizierten Moskitos übertragen.

(2) a. And **it** seemed to me that there were three levels of acceptance that needed to take place.
b. Und **es** schien, dass es drei Stufen der Akzeptanz gibt, die alle zum Tragen kommen mussten.

(3) a. But I think if we lost everyone with Down syndrome, **it** would be a catastrophic loss.
b. Aber, wenn wir alle Menschen mit Down-Syndrom verlören, wäre **das** ein katastrophaler Verlust.

At the more granular lower level, anaphoric pronouns are subdivided according to the following attributes: whether the pronoun appears in the same sentence as its antecedent (intra-sentential) or a different sentence (inter-sentential), the antecedent is a group noun, the pronoun is in subject or non-subject position (*it* only), or an instance of *they* is used as a singular pronoun (for example, to refer to a person of unknown gender). An overview of the resulting categories is provided in Table 1.

Within each category, we aim to create a balance in terms of the expected pronoun translation token. We achieve this by considering the translation of the set of possible candidates in the reference translation.

3. Evaluation Results

The semi-automatic evaluation method is a two-pass procedure. It is motivated by the observation that automatic reference-based methods can identify correct examples with relatively high precision, but low recall (Guillou and Hardmeier, 2018). The evaluation procedure relies on word alignments, which were generated automatically by running Giza++ (Och and Ney, 2003) in both directions with grow-diag-final symmetrization (Koehn et al., 2005). The word alignments for the examples in the reference translation were corrected manually.

In the first step, the candidate translations are matched against the reference translation to approve examples that we can assume to be correct with reasonable confidence. Examples in the event and pleonastic categories can be approved based on a pronoun match alone; for the anaphoric categories, we also require matching antecedent translations. Two pronoun translations are considered to match if the sets of words aligned to the pronouns have a non-empty intersection after lowercasing. For antecedent translation, the word sequences aligned to the source antecedent must be identical for an automatic match. As a special exception, no automatic matches are generated for pronoun translations containing the word *sie* alone, so that the ambiguity between third-person plural *sie* and the pronoun of polite address *Sie* can be manually resolved.

In the second step, all examples not automatically approved are loaded into a graphical analysis tool specifically designed for the PROTEST test suite (Hardmeier and Guillou, 2016). The tool presents the annotator with the source pronoun, its translation by a given system, and the previous sentence for context. In the case of anaphoric pronouns, the context includes the sentence with the antecedent and one additional sentence. The examples were split randomly over four annotators. The annotators are translator trainees at Saarland University. All of them are native speakers of German with good knowledge of English. To improve the quality of the annotations, the annotators had been trained beforehand on the output of a baseline NMT system.

Category	**–**	**+**	**total**	**correct**
Anaphoric				
intra-sent subj. *it*	5	39	44	88.6%
intra-sent non-subj. *it*	6	13	19	68.4%
inter-sent subj. *it*	13	16	29	55.2%
inter-sent non-subj. *it*	9	21	30	70.0%
intra-sent *they*	–	–	–	–
inter-sent *they*	–	–	–	–
singular *they*	–	–	–	–
group *it/they*	–	9	9	100.0%
Event reference *it*	14	68	82	82.9%
Pleonastic *it*	–	137	137	100.0%
Total	47	303	350	86.6%

Table 1: Human evaluation of automatically approved examples

The first step of our two-step procedure can only approve examples, it never rejects them automatically. As a consequence, our semi-automatic evaluation is *biased towards correctness* with respect to a fully manual evaluation. The test suite scores will therefore tend to overestimate the actual system performance.

The evaluation included 10 systems submitted to the English–German sub-task of the WMT 2018 competition and 6 anonymized online translation systems. Among the WMT submissions, all of the systems are neural models, with the Transformer (Vaswani et al., 2017) being a popular architecture choice. Implementation details can be found in the system description papers published at WMT 2018.

In total, 3,200 pronoun examples were evaluated. 1,150 examples were approved automatically and 2,050 examples were referred for manual annotation. To verify the validity of the semi-automatic method, we also solicited manual annotations for a random sample of 350 examples that had been approved automatically.

The results of the human annotation of the random sample of 350 examples automatically matched as correct are presented in Table 1. Consistently with similar results for French (Hardmeier and Guillou, 2018), 86.6% of the automatically approved examples were accepted as correct by the evaluators. However, we must highlight that the accuracy of the automatic evaluation varies substantially across categories. Whilst pronouns known to be pleonastic can be checked automatically with very good confidence, the automatic evaluation of anaphoric pronouns is much more difficult, with an evaluation accuracy as low as 55.2% in the inter-sentential subject *it* case. This reflects the general difficulty of automatic pronoun evaluation (Guillou and Hardmeier, 2018) and reinforces the positive bias discussed in the previous paragraph for these categories in particular.

The results of the semi-automatic evaluation are displayed in Table 2. For the counts in this table, we used *manual* annotations wherever possible. Automatic annotations were used only for those examples that had not been annotated manually.

	Pronouns											Antecedents
	anaphoric								event	pleonastic		
	it					they		it/they	it	it		
	intra		inter		intra	inter	sing.	group				
	subj.	non-subj.	subj.	non-subj.							**Total**	
Examples	*25*	*25*	*25*	*25*	*10*	*10*	*5*	*15*	*30*	*30*	***200***	*140*
Microsoft-Marian	18	20	12	15	9	10	2	13	29	29	**157**	132
NTT	16	18	14	16	10	10	1	8	26	29	**148**	135
UCAM	19	20	13	11	10	10	2	11	22	30	**148**	134
uedin	19	19	10	11	10	10	–	11	29	29	**148**	132
MMT-prod	20	19	11	15	10	8	–	9	25	29	**146**	137
KIT	19	18	15	11	9	9	1	6	27	30	**145**	126
online-Z	21	18	10	10	10	10	2	11	24	29	**145**	132
online-B	20	15	12	12	8	10	–	8	27	30	**142**	128
online-Y	18	17	11	12	10	9	1	8	24	30	**140**	136
JHU	12	17	8	11	8	10	3	10	24	29	**132**	119
online-F	13	16	10	11	10	10	2	7	21	28	**128**	115
LMU-nmt	10	9	10	13	7	10	1	9	28	30	**127**	125
online-A	11	9	12	16	5	10	2	5	27	30	**127**	130
online-G	10	6	15	11	2	8	2	7	23	30	**114**	119
RWTH-uns	9	5	9	8	3	8	1	7	19	29	**98**	99
LMU-uns	4	2	2	2	4	8	–	5	15	8	**50**	87
Average												
count	14.9	14.3	10.9	11.6	7.8	9.4	1.3	8.4	24.4	28.0	**130.9**	124.1
percentage	59.8	57.0	43.5	46.3	78.1	93.8	25.0	56.3	81.3	93.5	**65.4**	88.6

Table 2: Pronoun and antecedent translations marked as correct, per system

The best result was obtained by the Microsoft-Marian system, which translated 157 out of 200 pronouns correctly. It is followed by a group of 5 shared task submissions and the online-Z system that achieved scores between 145 and 148. Two of the online systems also reached scores over 140. The remaining shared task submissions are JHU with a score of 132 and LMU-nmt with a score of 127. Unsurprisingly, the unsupervised submissions are ranked last.

4. Discussion

The results of the manual evaluation vary significantly by category. In the anaphoric *it* categories, it is evident that intra-sentential anaphora are easier to handle than inter-sentential anaphora. In the intra-sentential case, the best systems produce correct translation in 70–80% of the examples, which is a fair result, but indicates that the problem is not completely solved yet. In the inter-sentential *it* categories, the average performance is below 50% despite the positive bias of our evaluation method, and even the best-performing systems are not much better. It is worth noting that no single system is best over all anaphoric categories, which suggests that the top scores achieved for this part of the test suite could be random strokes of luck. The results for pronouns in subject and non-subject positions are not very different. This contrasts with the results of Hardmeier and Guillou (2018) for English–French, where non-subject pronouns were found to be substantially harder to translate. This might be due to the fact that the direct object forms of French personal pronouns coincide with those of the definite article, a problem that does not apply to German.

The plural cases of *they* do not cause any serious problems, at least for the stronger systems, since *they* can usually be translated straightforwardly with the German pronoun *sie*. The errors occurring in these categories are often due to confusion with the pronoun of polite address *Sie* ("you"). When *they* has a singular antecedent or refers to a group, however, it is mistranslated much more frequently.

The only system that has noticeable problems with pleonastic *it* is the unsupervised LMU-uns submission. Translating event *it* seems to be more difficult, but many systems still achieve close to perfect results in this category. Similarly to the results of Hardmeier and Guillou (2018) for English–French, this suggests that NMT systems are quite good at identifying pronouns with event reference and producing appropriate translations for them.

5. Conclusions

We have presented a detailed analysis of 16 NMT systems assessing their performance in the translation of pronouns using a semi-automatic evaluation based on a balanced test suite. The results reinforce the idea that automatic evaluation scores are correlated with manual evaluation results, but they also confirm that automatic evaluation based on matching alone can give a misleading picture of the behavior of some systems. The evaluation has also reinforced that special attention should be paid to the problematic cases that are only identifiable through the careful balance of categories achieved in the test suite design. This balanced design has also made us aware of the progress made by NMT in the modeling of context for the translation of pleonastic,

event and intra-sentential anaphoric pronouns. Pleonastic pronouns are handled almost perfectly by most systems, so we suggest that future evaluations place more emphasis on the more challenging cases. Anaphoric pronouns depending on inter-sentential context remain a significant challenge. They present an ideal test case for the development of context-aware NMT systems. Research in that direction has recently gained some traction and has claimed promising results specifically for pronoun translation (Voita et al., 2018). It remains to be seen whether the development of such methods will lead to a breakthrough in the translation of inter-sentential anaphoric pronouns in the near future.

Acknowledgements

The work carried out at Uppsala University was supported by the Swedish Research Council under grants 2012-916 (to Jörg Tiedemann) and 2017-930 (to Christian Hardmeier). We thank Jörg Tiedemann for helping us fund this effort. The work carried out at The University of Edinburgh was funded by the ERC H2020 Advanced Fellowship GA 742137 SEMANTAX and a grant from The University of Edinburgh and Huawei Technologies. The manual test suite evaluation was funded by the European Association for Machine Translation. We thank our annotators Daria Hert, Georg Seiler, Alexander Schütz and Peter Schneider for their valuable work.

References

Liane Guillou and Christian Hardmeier. 2016. PROTEST: A test suite for evaluating pronouns in machine translation. In *Proceedings of the Eleventh Language Resources and Evaluation Conference*, LREC 2016, pages 636–643, Portorož, Slovenia.

Liane Guillou and Christian Hardmeier. 2018. Automatic reference-based evaluation of pronoun translation misses the point. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 000–001, Brussels, Belgium. Association for Computational Linguistics.

Liane Guillou, Christian Hardmeier, Preslav Nakov, Sara Stymne, Jörg Tiedemann, Yannick Versley, Mauro Cettolo, Bonnie Webber, and Andrei Popescu-Belis. 2016. Findings of the 2016 WMT shared task on cross-lingual pronoun prediction. In *Proceedings of the First Conference on Machine Translation*, pages 525–542, Berlin, Germany. Association for Computational Linguistics.

Christian Hardmeier and Liane Guillou. 2016. A graphical pronoun analysis tool for the protest pronoun evaluation test suite. *Baltic Journal of Modern Computing*, 4(2):318–330.

Christian Hardmeier and Liane Guillou. 2018. Pronoun translation in English–French machine translation: An analysis of error types. *ArXiv e-prints*, 1808.10196.

Christian Hardmeier, Preslav Nakov, Sara Stymne, Jörg Tiedemann, Yannick Versley, and Mauro Cettolo. 2015. Pronoun-focused MT and cross-lingual pronoun prediction: Findings of the 2015 DiscoMT shared task on pronoun translation. In *Proceedings of the Second Workshop on Discourse in Machine Translation*, pages 1–16, Lisbon, Portugal, September. Association for Computational Linguistics.

Christian Hardmeier. 2012. Discourse in statistical machine translation: A survey and a case study. *Discours*, 11.

Christian Hardmeier. 2015. On statistical machine translation and translation theory. In *Proceedings of the Second Workshop on Discourse in Machine Translation*, pages 168–172, Lisbon (Portugal), September. Association for Computational Linguistics.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *International workshop on spoken language translation*, Pittsburgh, Pennsylvania.

Ekaterina Lapshinova-Koltunski, Christian Hardmeier, and Pauline Krielke. 2018. ParCorFull: a parallel corpus annotated with full coreference. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of 11th Language Resources and Evaluation Conference*, pages 423–428, Paris, France, may. European Language Resources Association (ELRA).

Sharid Loáiciga, Sara Stymne, Preslav Nakov, Christian Hardmeier, Jörg Tiedemann, Mauro Cettolo, and Yannick Versley. 2017. Findings of the 2017 DiscoMT shared task on cross-lingual pronoun prediction. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 1–16, Copenhagen, Denmark. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29:19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, ACL 2002, pages 311–318, Philadelphia. Association for Computational Linguistics.

Karin Sim Smith. 2017. On integrating discourse in machine translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 110–121, Copenhagen, Denmark. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274. Association for Computational Linguistics.

Towards a Swedish text and speech corpus for fiction literature

Christina Tånnander and Tam Johnson
Swedish Agency for Accessible Media
Johanneshov, Sweden
christina.tannander@mtm.se tam.johnson@mtm.se

Abstract

We present an initial version of a Swedish corpus for fiction literature. The corpus consists of 40 talking books read by four different narrators and includes both text and the corresponding speech. It is intended to be used as a basic source of information for how human narrators read fiction literature aloud, in order to understand what must be done to produce acceptable narrations of fiction literature by a synthetic voice. We define the labels SPEAKER, SAIDPHRASE or NARRATIVE and apply them, manually, to text extracts of 12 pages per book. The paper describes the material and presents statistics on the proportions of dialogue in the corpus, as well as the number of shifts from NARRATIVE to SPEAKER, SPEAKER to NARRATIVE or SAIDPHRASE, and from SPEAKER to SPEAKER.

1. Introduction

This paper presents an initial Swedish text and speech corpus for fiction literature, intended to serve as a basis for source of information of how human narrators read fiction literature aloud. The corpus consists of 40 Swedish talking books produced at the Swedish Agency for Accessible Media (MTM).

1.1 The Swedish Agency for Accessible Media

The Swedish Agency for Accessible Media (Myndigheten för tillgängliga medier, https://mtm.se) is a governmental authority that produces literature in accessible formats such as Braille and talking books for people with reading disabilities. The agency produces fiction talking books, most often narrated by human narrators, university text books, of which more than 50% are produced with synthetic speech, as well as more than 100 newspapers produced with synthetic voices (Tånnander, 2018).

1.2 Dialogues in fiction and speech synthesis

In 2016, MTM conducted a quantitative survey to find out whether talking book users could accept listening to an English fiction book with synthetic speech (MTM, 2017a; Tånnander, 2018). The results showed that about 66% thought that they could have an acceptable reading experience with the synthetic voice, given that they become familiar with the voice. However, 34% of the subjects pointed out that the dialogues were read poorly, and a following qualitative survey (MTM, 2017b) showed that the users found it difficult to understand where conversations start and end, when speaker shifts occur and who the current speaker is. This reveals a need to investigate human strategies of signaling shifts between dialogue, narrative and between speakers, as well as cues to who the speaker is, and to find ways to apply these strategies on synthetic voices.

A French audiobook corpus that is currently available for similar purposes is the SynPaFlex corpus (Sini et al, 2018). It consists of 87 hours of speech read by the same narrator and has been collected for the purpose of developing models to control expressiveness in synthetic speech and also contains information about speaker ID and vocal personality ID, the latter identifying which kind of voice the narrator used for different characters.

Our long-term goal involves several steps. Firstly, dialogues must be detected. This is trivial if speaker lines are consistently marked with unambiguous quotes, for example *»speaker line«*, but it becomes trickier if the lines are unmarked. Secondly, each line must be assigned to a speaker. Attempts to automatically identify speakers in Swedish textual dialogues has shown an accuracy of about 70% for certain authors (Ek et al., 2018). Thirdly, we need to find out how these shifts between narrative, dialogue and speakers should be signaled by synthetic voices, in particular by analyzing what human Swedish narrators do and in turn apply these strategies to synthetic speech.

The current corpus constitutes a starting point in the endeavor of creating intelligible, distinct and self-explanatory readings of fiction literature by a synthetic voice.

2. Method

The selection and processing of the talking books was conducted according to the following steps: (1) selection of narrators and books, (2) mark-up and (3) text analysis.

2.1 Selection of narrators and books

Narrators and books were selected manually, applying the following principles:

- Four narrators, balanced for gender (two male and two female) who had read at least ten talking books during 2010-2018.
- Ten books of each narrator, all of which must have (a) text in XML format attached to the speech, (b) be recorded between 2010-2018, and (c) contain at least some chunks of dialogue.

2.2 Mark-up

12 pages of each book were manually labelled: four pages in the beginning, middle and end of the book, respectively. These 12 pages represent between 1.6% and 37.5% of the pages of each book, with an average percentage of 3.92%, which means that some books have a much larger proportion of mark-up than others. However, at this early stage of the text and speech corpus, we prioritized having

the same size of marked-up material per narrator, instead of a more proportional representation of each book.

The following tags were used:

- SPEAKER for direct dialogues where a speaker says something. This tag has two attributes, GENDER (female, male or unknown) and a book unique SPEAKERID, usually consisting of the speaker's name.
- SAIDPHRASE for said-phrases such as *'he said'*, including possible subsequent words until next delimiter.
- NARRATIVE for all other text chunks.

Example:

<SPEAKER_M_PETER>Hello</SPEAKER_M_PETER>,
<SAIDPHRASE>he said and left the room.</SAIDPHRASE>.
<NARRATIVE >The woman followed him.</NARRATIVE >

The reason for splitting narrative reading into said-phrases and narrative is that we are interested in looking at the prosodic features revealed by these type of phrases, as opposed to the reading of the remaining narrative.

2.3 Text analysis

Three types of analyses were performed; a text analysis calculating word types and tokens, an estimation of how much text that consists of dialogue, said-phrases and narrative, as well as the number of shifts between the labels SPEAKER, NARRATIVE and SAIDPHRASE.

The number of types and tokens were calculated for each book. A word type is here defined as a graphic word, as opposed to a lemma (Youmans, 1990). Hence, *'katt'* and *'katter'* (*'cat'* and *'cats'*) are different words. The reason is simply that the current corpus represents text read aloud, and therefore the purpose is not to measure the author's lexical variation, but rather to get a hint of the number of different phonological words in the text (though no consideration was taken to words that are written in different ways but pronounced the same, such as *'24'* and *'twenty four'*).

The manual mark-up allows us to compute the proportion of direct dialogue and narrative (including said-phrases) as a percentage of the entire text contained in the 12 pages of each book with this mark-up.

In addition, the shifts between NARRATIVE, SAIDPHRASE and SPEAKER were counted, as well as the adjacent shifts between different speakers.

3. Technical description

The talking books are each produced as file-sets consisting of MP3, XHTML and SMIL files. The MP3 files have a sampling frequency of 22,050 Hz and a bit rate of 48 Kbit/s. Text in the XHTML content files are linked to specific locations in the audio files via time stamp values stored in the SMIL files. In this way the recorded speech is synchronized to text at the paragraph and sentence level. Furthermore, the file-sets used for this corpus are derived from a source XML format. It is this structured format that is the basis for the dialogue mark-up defined above.

4. Analyses results

4.1 Word types and tokens

In total, the corpus consists of more than 400 hours of recordings, distributed among the four narrators as shown in Table 1. The table also shows types and tokens in the four book chunks. The shortest playing time of a book is 48 minutes (1,111 tokens and 4,491 types) and the longest 18 hours and 54 minutes (20,865 tokens and 158,035 types).

Narrator	Gender	Time	Tokens	Types (avg)
A	M	104:48	934,242	10,462
B	M	120:50	862,333	10,699
C	F	95:37	750,377	9,968
D	F	80:20	623,832	8,388
SUM	-	401:35	3,170,784	-
AVG/book	-	10:02	79,270	9,879

Table 1. For the ten books read by each narrator the columns show: Narrator ID, gender, total recording time (HH:MM), total number of tokens, and average number of types.

4.2 Dialogue, narrative and speaker shifts

The average proportions of direct dialogue, said-phrases and narrative per narrator are shown in Table 2, as well as the total number of shifts between a speaker and narrative, said-phrase or another speaker.

The average proportion of segments labelled as SPEAKER in the books is 25.8%, with a variation between 0 and 78.34%. This variation is explained by the fact that for two books, the extract of 12 pages did not have any representations of dialogue, even if they existed on the remaining pages of the book. The book with 78.34% dialogue is from a play, mainly consisting of dialogues and some shorter instructions for the readers/actors, such as '*silence'* or '*turning his head'*. The label SAIDPHRASE exists in the extracts of 34 of the 40 books and make up between 0 and 6.23% of the text, with an average of 1.91%. Finally, the NARRATIVE label occurs for an average of 72.49%, ranging on a book basis from 21.67 to 100%.

In Table 2, only the shifts involving SPEAKER, namely a line in a direct dialogue, are shown, though all types of shifts were computed. Since the mark-up does not include tagging of entire dialogues, we cannot say anything about how many dialogue beginnings or endings there are in the extracts. The shift from NARRATIVE to SPEAKER or from SPEAKER to NARRATIVE in Table 2, might represent the beginning and the end of a dialogue, but can also mean that there is a narrative sentence within a dialogue, as in the following example:

<SPEAKER_M_PETER>Hello.</SPEAKER_M_PETER>,
<SAID_PHRASE>he said.</SAID_PHRASE>.
<SPEAKER_F_JANE>Hello there.</SPEAKER_F_JANE>
<NARRATIVE>They stared at each other</NARRATIVE>
<SPEAKER_M_PETER>Oh my.</SPEAKER_M_PETER>

Narrator	SPEAKER (avg)	SAIDPHRASE (avg)	NARRATIVE (avg)	NARRATIVE-SPEAKER (avg)	SPEAKER-NARRATIVE (avg)	SPEAKER-SAIDPHRASE (avg)	SPEAKER-SPEAKER (avg)
A	28.50%	0.80%	70.94%	31.6	29.8	6.2	19.5
B	29.37%	4.00%	67.54%	35.6	29.1	24,5	28.1
C	19.86%	1.99%	78.37%	24.1	18	16.8	11.5
D	24.80%	1.37%	73.82%	16.3	13.8	7.4	13.4
Total average	25.80%	1.91%	72.49%	26.90	22.43	13.73	18.13

Table 2. The first column shows the four different narrators, the next three the proportions of dialogues, said-phrases and narrative in the marked-up extracts, and the final four columns the average number of shifts from SPEAKER to NARRATIVE, NARRATIVE to SPEAKER, SPEAKER to SAIDPHRASE and SPEAKER to SPEAKER.

Hence, we can only conclude that the shift from NARRATIVE to another SPEAKER occurs 1076 times in the 480 marked up pages, which computes to an average of 26.90 times/book. The corresponding numbers for SPEAKER to NARRATIVE is a total of 897, with an average of 22.43 times per book, while that for SPEAKER to SAIDPHRASE totals at 549, an average of 13.73 times per book. Finally, there are 725 examples of speaker shifts, averaging at 18.13 times per book.

5. Discussion

This approach towards a Swedish text and speech corpus for fiction literature reveals that even if we are using just a small sample of each book, in this case 12 pages, we can generate a sufficient number of examples of shifts between spoken dialogue lines, narrative and said-phrases. In other words, we believe that a manual inspection will allow us to get an idea of what our four human narrators do to signal these shifts, as well as taking into consideration how prosodic features might be mapped to said-phrases. Additional questions that we would like to raise are how the human narrators signal the gender of the speaker in the dialogue, as well as the importance of the consistency in using the same voice features for the different characters throughout the book.

A larger corpus, including a greater variety of narrators and more sections with dialogue mark-up (where also the beginning and end of each dialogue is marked), could serve as a basis for the development of models for dialogues in Swedish fiction, which in turn could be used by text-to-speech systems to provide a better listening experience of fiction. We also hope to gain access to the original sound files which have a sampling frequency of 44,100 Hz instead of the current down-sampled mp3 files of 22,050 Hz. Unfortunately, at the time of writing we are not able to release the corpus due to copyright issues.

Furthermore, the corpus can be used as a text corpus alone, allowing us to calculate for example type-token curves (Youmans, 1990), which can be used as a guide for deciding whether a book should be produced with a human or a synthetic voice.

To conclude: if we (1) automatically detect dialogues in text; (2) automatically identify speakers in these dialogues; and (3) apply human strategies of signaling dialogue/narrative shifts as well as speaker shifts in synthetic speech, and complete these steps with some human control, we have come a good way in our attempts to produce acceptable speech synthesis for fiction literature.

References

A. Ek, M. Wirén, R. Östling, K. N. Björkenstam, G. E. Grigonyt˙ & S. Gustafson Capková. 2018. Identifying Speakers and Addressees in Dialogues Extracted from Literary Fiction. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan.

MTM. 2017a. *Engelsk talsyntes: kvantitativ undersökning*. Stockholm, Sweden.

MTM. 2017b. *Engelsk talsyntes: kvalitativ undersökning*. Stockholm, Sweden.

A. Sini, D. Lolive, G. Vidal, M. Tahon, & É. Delais-Roussarie. 2018. SynPaFlex-Corpus: An Expressive French Audiobooks Corpus Dedicated to Expressive Speech Synthesis. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan.

C. Tånnander. 2018. Speech Synthesis and evaluation at MTM. *Proceedings of Fonetik* (pp. 75–80). Gothenburg: Gothenburg University.

G. Youmans. 1990. Measuring Lexical Style and Competence: The TypeToken Vocabulary Curve. *Style*, *24*(4), 584–599.

Exploring the Quality of the Digital Historical Newspaper Archive KubHist

Yvonne Adesam, Dana Dannélls, Nina Tahmasebi

Språkbanken, Språkbanken, Centre for Digital Humanities
University of Gothenburg
{yvonne.adesam/dana.dannells/nina.tahmasebi}@gu.se

Abstract

The KubHist Corpus is a massive corpus of Swedish historical newspapers, digitized by the Royal Swedish library, and available through the Språkbanken corpus infrastructure Korp (Borin et al., 2012). This paper contains a first overview of the KubHist corpus, exploring some of the difficulties with the data, such as OCR errors and lemma annotation, and discussing possible paths for improving the quality and the searchability.

1. Introduction

The past decades have seen a massive increase in digitized, historical documents that have been at the core of a range of different applications, from studies of cultural and language phenomena (Michel et al., 2011) to temporal information retrieval and extraction. The study of semantic changes, to give one example, has changed character from qualitative studies (Viberg, 1980; Vejdemo, 2017) to automatic detection via topic modeling (Lau et al., 2012), and word sense induction (Tahmasebi and Risse, 2017) to methods based on (neural) embeddings (Kulkarni et al., 2015; Basile et al., 2016). In common for the majority of the existing methods and studies is that they focus on English texts because of the vast amounts of easily available data, for example through the Google N-gram corpus.

The availability and easy access of datasets like the Google N-gram corpora, and others in full text form, like the Corpus of Historical American English (COHA), and the Penn Parsed Corpora of Historical English, draws researchers to English texts and hence, creates methods developed for the English language.

In Sweden, the amount of digital, historical texts is large compared to many other languages, but still in the shadows of that available for English. There have been few possibilities to make diachronic studies, and develop tools for historical Swedish and automatic detection of language changes. The first, large newspaper corpus, KubHist, is a good step towards this goal.

The first version of the KubHist dataset, currently available through Språkbanken, contains close to 1.1 billion tokens. Originally going under the name DigiDaily – after the project which led to the digitization of the first batch of historical newspapers, involving the Royal Swedish Library and the Swedish National Archives (`https://riksarkivet.se/digidaily`) – the corpus soon changed its name to KubHist (Kungliga bibliotekets historiska tidningar), as more material was added after the end of the project. More recently, parts of the material have been re-processed, to create data of a higher quality. Additional material has also been added. In this paper, we investigate this new KubHist corpus, which contains more than 5.5 billion tokens, and will be added to Korp after being processed.

Many of the modern methods for processing historical data rely on neural embedding methods that require large amounts of text (i.e., tokens that are automatically recognized). However, even 5.5 billion tokens is a small amount, considering that it is spread over roughly 200 years. The amount of available tokens per year ranges from 800 tokens in 1647 to 156 million tokens in 1892, see Figure 1 for an overview of the distribution of tokens over time. In addition to the low amount of data for most years, the quality of the tokens affect the results.

We know that the KubHist dataset, spanning 1645 – 1926, contains a large number of OCR errors, ranging from one misrecognized character in a word – including space, which splits a word into several tokens or joins several words into one token – to producing gibberish which is not understandable without consulting the image (and sometimes even the image is not enough). Because of this, the number of tokens is just an initial estimate, which will vary during the processing of the material. The texts have been automatically annotated. The quality of these annotations varies greatly, due to the annotation tools not being adapted to the historical language variety, bad OCR quality, and spelling variation.

The aim of this paper is to get an estimate of the quality of the texts by studying OCR errors and the annotated lemmas. These estimates will help focus our future efforts. The end goal is to automatically detect semantic change by correcting OCR errors and normalizing spelling variation and change. The improvements have a value in themselves, making these diachronic texts better suitable both for manual search and for automatic processing, within, for example, the digital humanities.

2. Basic Annotation

The newspapers in KubHist have been digitized by taking high-quality images of the pages, and then applying OCR software, see Sec. 3. for details. The resulting XML-files have then been processed by the Sparv annotation tools (`https://spraakbanken.gu.se/sparv/`). Sparv consists of a range of linguistic annotations, from tokenization to part-of-speech tagging and named entity recognition. Common for all tools in Sparv is that they were developed for modern Swedish and not the language from the KubHist time period. However, the system has a number

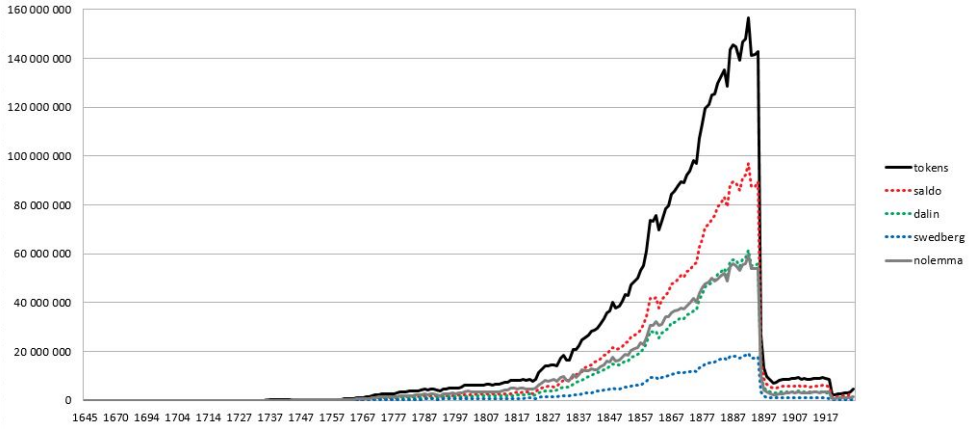


Figure 1: The number of tokens and assigned lemmas, from the different dictionaries, per year.

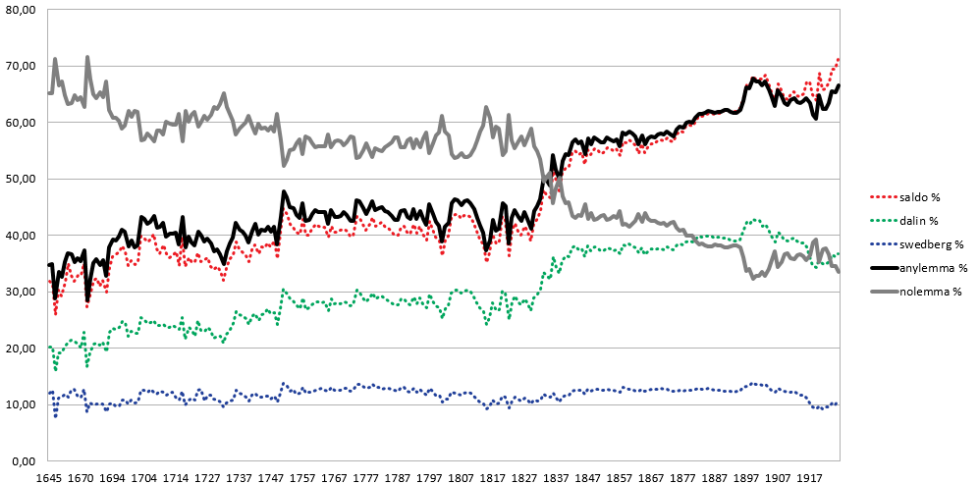


Figure 2: The number of assigned lemmas in percent, from the different dictionaries, per year.

of historical lexical resources available, and we will use them to link tokens in the text to lexicon entries, and assign lemmas. The dictionaries relevant for the texts at hand are Saldo (Borin et al., 2013) over contemporary Swedish, Dalin (1853/1855), over 19th century Swedish, and Swedberg (Holm, 2009), over 18th century Swedish (Borin and Forsberg, 2011). Apart from the lexicon itself, a morphology is needed (basically a full form lexicon) to also match inflected forms (Borin and Forsberg, 2008). The morphologies for the different dictionaries are at varying level of development.

Figure 1 shows the numbers of assigned lemmas from the different lexica, as well as the number of tokens without any assigned lemma. Although there will be errors in the lemma assignment, stemming from, e.g., OCR errors, we will focus on the tokens without lemma, to identify potential OCR errors. However, a missing lemma may also come from words missing in the lexicon (which is especially obvious in the case of names), as well as an underdeveloped morphology. In addition, we will explore the cases where we have a Dalin or Swedberg lemma, but no Saldo lemma, since we would like to increase the diachronic links between the lexica.

In Figure 2, we see the coverage of the different lexica over time, represented as percent of the number of tokens for each year. Swedberg has a fairly stable lemma assignment over time, around 12 percent. The generally low percentage comes from the fact that this resource has the smallest morphology attached to it. Saldo and Dalin follow each other over time, although Saldo has by far the largest morphology, which is seen in the gap between the two. However, after 1900, Dalin drops in coverage, which is most likely because of the spelling reform of 1906, after which the Dalin spelling no longer matches the spelling in the newspapers.

We also see that the assignment of Saldo and Dalin lemmas increases from the 1830s, resulting in more tokens having a lemma in at least one resource, than tokens having no lemma. We assume that the fonts, print, and paper quality decrease the number of OCR errors around this time.

3. OCR errors

There may be several reasons for the low quality of the digital text, after automatic OCR processing. The quality of the paper or print may be low, resulting in smudgy images for the OCR software to work with. Various font sizes, uneven text lines and a varying amount of columns, causing difficulties for the OCR software to analyze the structure of the image. As a result, e.g., points and accents are mistaken for noise, graphic or geometric symbols are interpreted as

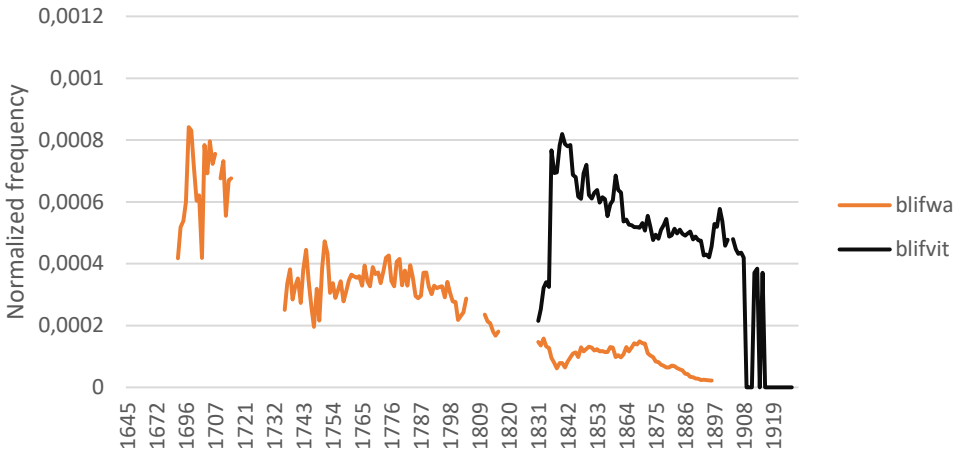


Figure 3: Two spelling variants with some overlap, but mostly complementary use.

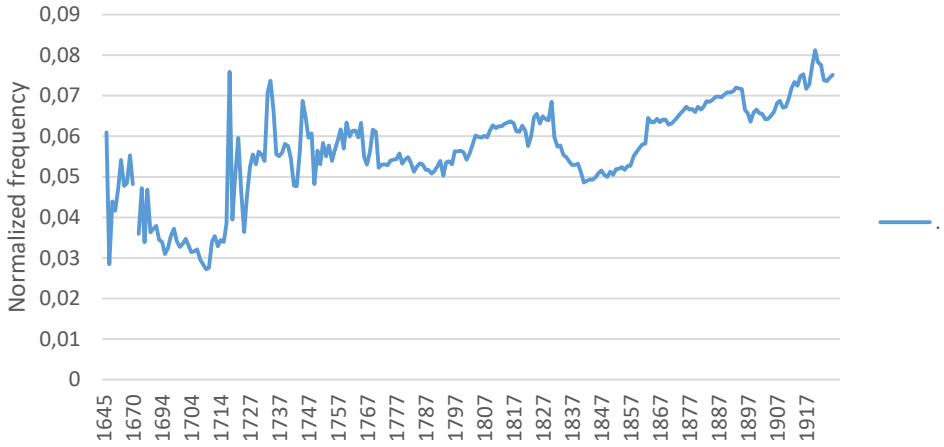


Figure 4: The frequency of '.' (period) over time. While not present in a dictionary, it should not be categorized as spelling variation or OCR error at a general level.

text, and characters are interpreted as symbols. Another difficulty is the mixture of font types, most notably blackletter and roman typefaces, which requires that the OCR software is properly trained on different types of old fonts and languages.

The KubHist material has been processed by the commercial ABBYY Finereader OCR software, which is known to achieve high OCR accuracy, but unfortunately does not process our material with sufficient quality. This becomes apparent when we study the rate of lemma-assignment. Although there are several reasons for the annotation tools not being able to assign lemmas to tokens in the texts, a low rate of lemma assignment may point to, e.g., blackletter articles. When we explore the 349.608 OCR processed newspaper editions, we find that 27% have a lemma-rate of 50% or lower. Only 3% have a lemma-rate of above 80%. (It should be noted that this does not say anything about the quality of lemma-assignment, it just states that a number of tokens were identified as forms of words in the lexicon by the annotation tools.)

In an initial experiment we examined the top 500 most common tokens that did not receive a lemma, categorizing them according to 7 attributes. We found that around 75% should not receive a lemma, as they were instances of numbers or punctuation. Less than 3% contained OCR errors (although, we would not expect many to show up among the most frequent words), and under 4% required some kind of processing as they were spelling variants which the tools did not recognize. However, as these top 500 words were explored as word types, in isolation, around 10% could also not be categorized out of context. Overall, although numbers and punctuation may contain a large amount of OCR errors, this shows that we should not be aiming for 100% lemma coverage, but that the desired upper bound is much lower (unless numbers and punctuation are also included in the lexicon).

For comparison, we examined two digitized historical texts that have been manually transcribed and processed by our annotation tools, where we can assume that there are no OCR errors. One contains law text from 1734, with close to 100.000 tokens, and around 35% of the tokens have no lemma. The other contains judicial protocols, with around 120.000 tokens, and almost 60% of the tokens have not been assigned a lemma. For both of these texts, the

Dalin and Swedberg dictionaries (but currently not Saldo) have been used for lemma assignment. Looking at modern news text, in Göteborgsposten of 2013 with 16.870.000 tokens, a little over 20% of the tokens are missing a lemma. Texts with more variation, such as the 2017 Bloggmix (various Swedish blogs), contains almost 1.670.000 tokens, and close to 22% of these tokens have not been assigned a lemma. For these modern corpora, Saldo has been used for lemma assignment.

Thus, a reasonable upper bound for lemma assignment for modern Swedish, using a lexical resource like Saldo, is closer to 80%. For historical texts, the variation is larger, and the upper bound is quite a bit lower than for modern texts.

4. Spelling Variation

We split the KubHist dataset into 50-year bins and explore the most frequent words that were not assigned a lemma. Our hypothesis is that words that appear in many bins among the most frequent words, are unlikely to be OCR errors. Instead we expect words that are OCR errors to be less frequent, unless they are consistent with font errors. A word like *massor* ('many', 'masses') could translate to one of "niassor", "iiiessor" etc, and its frequency should be distributed over multiple possible errors, rather than concentrated to one given form.

When looking at the most frequent non-lemma words in all bins, we find that these are different types of punctuation (*!"' ()*,-.:;?/»*) and numbers (*1-9*, *14*), as well as single letters (*M*, *a*, *m*, *n*, *r*, *t*), and a few words (*ägt*, *nied*). Among those that were frequent in only one bin we find names (*Londén*, *Maji*, *Borgholm*), uncommon short-lived abbreviations (*k.*, *K.*), and possible OCR errors (*ocb/oeh* → *och*, *näget* → *något*). Important to remember is that the first 50-year bin has very little text from only a few sources (we have only a couple of newspapers available) which means that a name like *Borgholm* could be e.g. the name of a journalist, and not universally important. This remains to be investigated.

Among the words that are not assigned a lemma and appear in three bins, i.e., three 50-year periods, we have words that are common spelling variants, such as *öfwer*, *warit*, *blifwa*, *äfven*, *hwilka*, *blifvit*, *hvilka*, *hafwa*. Interestingly, some of these seem to hand over to each other, like in the case of *blifwa* and *blifvit* in Figure 3. The latter has a normalized form *blifva* that is present in Dalin, but due to an incomplete morphological description, the past tense of *blifvit* is not captured. Their frequency seems complementary. Observe that years without a frequency corresponds to an absolute frequency of below 50 occurrences. In the case of a frequently occurring character without an entry in the dictionary, '.' (period), Figure 4, we see that the frequency is much more consistent across years.

In future work, we intend to use these characteristics to attempt to automatically categorize words without lemma assignment as either OCR errors (which we expect to have a low, but consistent frequency), spelling variants (with a higher frequency focused around a specific period in time), or common characters not included in dictionaries (punctuation, numbers, etc).

References

Pierpaolo Basile, Annalina Caputo, Roberta Luisi, and Giovanni Semeraro. 2016. Diachronic analysis of the Italian language exploiting Google Ngram. In *Italian Conf. on Comp. Linguistics*.

Lars Borin and Markus Forsberg. 2008. Something old, something new: A computational morphological description of Old Swedish. In *LREC Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2008)*, pages 9–16, Marrakech.

Lars Borin and Markus Forsberg. 2011. A diachronic computational lexical resource for 800 years of Swedish. In Caroline Sporleder, Antal van den Bosch, and Kalliopi Zervanou, editors, *Language technology for cultural heritage*, pages 41–61, Berlin. Springer.

Lars Borin, Markus Forsberg, and Johan Roxendal. 2012. Korp the corpus infrastructure of Språkbanken. In *Proceedings of LREC 2012*, Istanbul. ELRA.

Lars Borin, Markus Forsberg, and Lennart Lönngren. 2013. SALDO: a touch of yin to WordNet's yang. *Language Resources and Evaluation*, 47(4):1191–1211.

Anders Fredrik Dalin. 1853/1855. *Ordbok öfver svenska språket*, volume I–II. Stockholm, Sweden.

Lars Holm, editor. 2009. *Jesper Swedberg: Swensk Ordabok*. Skara stiftshistoriska sällskaps skriftserie. Stiftelsen för utgivande av Skaramissalet, Skara.

Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2015. Statistically significant detection of linguistic change. In *WWW 2015*.

Jey Han Lau, Paul Cook, Diana McCarthy, David Newman, and Timothy Baldwin. 2012. Word sense induction for novel sense detection. In *EACL 2012*, pages 591–601.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray, The Google Books Team, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez Lieberman Aiden. 2011. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182.

Nina Tahmasebi and Thomas Risse. 2017. Finding individual word sense changes and their delay in appearance. In *RANLP 2017*.

Susanne Vejdemo. 2017. *Triangulating Perspectives on Lexical Replacement: From Predictive Statistical Models to Descriptive Color Linguistics*. Ph.D. thesis, Stockholm University.

Åke Viberg. 1980. *Studier i kontrastiv lexikologi: perceptionsverb*. Stockholms Universitet, Inst. för lingvistik.

A Challenge Set for English-Swedish Machine Translation

Lars Ahrenberg

Department of Computer and Information Science
Linkping University
lars.ahrenberg@liu.se

Abstract

This paper presents a project aimed at creating a challenge set for machine translation from English to Swedish. A challenge set is a test suite where sentences or short text snippets with their translations have been selected for purposes of evaluation. The current version contains 202 cases covering various translation problems in the direction from English to Swedish.

1. Introduction

Evaluation is a long-standing issue in natural-language processing, not least in machine translation. While the focus in recent years has been on metrics that can be computed automatically, such as BLEU (Papineni et al., 2002) or TER (Snover et al., 2006), they are not very informative. A potential user may be more interested in knowing the strengths and weaknesses of a given system. What can it do well? When is it likely to produce incorrect translations?

A common approach to more informative evaluations is error analysis (Vilar et al., 2006; Stymne and Ahrenberg, 2012). With a not too big and interpretable error taxonomy a user can get a good picture of what kind of mistakes a system is making when applied to a given text. A drawback with error analysis, though, is that the properties of the analysed text(s) are generally unknown so that we don't get information on what the system did right or on the frequency of constructions that sometimes give rise to errors.

Test suites may be seen as an alternative or complement to error analysis. King and Falkedal (King and Falkedal, 1990) discuss pros and cons of test suites for machine translation evaluation, and suggest that they can be valuable in spite of some pitfalls. One such list may be targeted at source language coverage, while another may be targeted at specific translation problems for the language pair in question, in particular at constructions where the two languages show 'mismatches'. They also argue that selection of inputs should be corpus-based.

When the paper by King and Falkedal was published, the capacity of a machine translation system was far below the capacities of present online systems. New technologies, such as Neural MT, are generally quite capable but also opaque and sometimes give errors that are hard to explain and describe. For this reason, (Isabelle, Cherry, and Foster, 2017) advocate a "challenge set approach" to evaluation of modern systems as a way to probe their capabilities. A challenge set, as the name suggests, should focus on difficult cases but the cases should be categorized so that the system output can be described and quantified in understandable terms. This paper presents a first version of a challenge set for English-Swedish machine translation.

A more ambitious approach to the use of test suites for machine translation evaluation is taken in the QT21 project (Burchardt et al., 2017; Macketanz et al., 2018). The ultimate goal is described as to "represent all phenomena relevant for translation" (Burchardt et al., 2017, p. 164) and provide for (semi-)automatic evaluation (Macketanz et al., 2018). Currently, their German-English test suite contains some 5,000 segments categorised into 15 major categories and some 120 different phenomena. The test suite is not published with the explicit reason "[t]o prevent overfitting or cheating"! .

2. The challenge set approach

In this work I decided to follow the challenge set approach centered around the notion of divergence or mismatch. A divergence is present in a translation if some construction in the source has been translated by a non-isomorphic construction. (Isabelle, Cherry, and Foster, 2017) suggests that a challenge set should be based on forced divergences, i.e., cases where the target language does not have an isomorphic construction, either because it does not exist at all in the language, or because the linguistic context is such that it cannot be used.

In (Isabelle, Cherry, and Foster, 2017) the divergences are divided into three major classes: morpho-syntactic, lexico-syntactic and (pure) syntactic ones. In (Isabelle and Kuhn, 2018) a fourth category, purely lexical divergences, was added. These categories are quite general, and not very informative in themselves. However, for the purposes of this paper, they are retained, and are defined as follows:

- **Morpho-syntactic divergence**. The divergence involves a morphological feature that is either not present in the source language, or, if it is, must change value in the target language sentence. A case in point for English-Swedish translation is gender agreement on determiners, pronouns, and adjectives. See Table 1 for an example.

- **Lexico-syntactic divergence**. The divergence involves a change in syntactic structure, such as complement structures, when a lexical item is translated by its typical synonym in the target language. For example, the English verb *want* is often constructed with an object NP and an infinitive VP (*want x to protest*) which is not available for the Swedish synonym *vill*, which instead requires a subordinate finite clause beginning with the subjunction *att*: *vill att x protesterar*.

SRC	The table she bought was **cheap**.
SYS	Bordet hon köpte var billig.
QUE	*Does the Swedish word translating 'cheap' have the proper form?*
SUG	billigt
ANS	YES NO NA

Table 1: A sentence with its focus question and suggestion.

- **Purely syntactic divergence**. The divergence involves a construction with no isomorphic counterpart in the target language. An example is the necessity to place a finite verb in the second position of a Swedish translation, although the English source verb may be in third or fourth position.

- **Purely lexical divergence**. These concern differences in selection of lexical items, including support verbs, prepositions and idioms. A case in point is the English verb *put* which usually requires a Swedish translation with a more specific sense.

An important aspect of the challenge set approach is that only one phenomenon for every example is evaluated. Every input sentence is supplied with a question, that focuses attention to some part of the source sentence, and that can be answered by a clear 'yes' or 'no'. For an illustration, see again Table 1.

Evaluation of a system with a challenge set is straightforward. The translations returned from a system are put into a form and each one is put together with its source sentence and the focus question. The human evaluators will then answer the question by yes or no. The performance of the system is captured by computing the share of correct translations in each category.

A challenge set can be used to compare several systems at one occasion or to compare different versions of the same system.

3. The English-Swedish Challenge Set

3.1 Design changes

We have made some minor changes to the design used by (Isabelle, Cherry, and Foster, 2017). They give a reference translation for each source sentence. As the question is focusing on a single aspect of the source, we believe that a complete reference translation may be too normative. Instead, the evaluator is given one or more suggestions for good translations of the focused part (SUG in Table 1), and, if known, responses that should be considered errors.

As in the English-French set every example carries a finer description of what divergence it is supposed to illustrate. In addition, the examples have been structured in pairs, with one member being judged a little more difficult to translate than the other. This added difficulty may have various sources, for instance, a longer distance between a targeted phrase and its governor, or the use of rarer words.

While the aim is to obtain a clear yes- or no-answer, this may not always be possible. For this reason the English-Swedish set includes a third alternative, NA, for 'not applicable'. This alternative can be used if the system somehow manages to circumvent the problem associated with the focused part, or if the evaluator cannot decide.

Category	Examples
Morpho-syntactic	48
Agreement in NP	10
ADJ-agreement in predication	14
Noun compounding	8
Pronoun coreference	6
Other	10
Lexico-syntactic	62
Sense-distinguishing context	30
NP-to-VP complements	8
Wh-phrases	8
Explicitation	8
Double object	4
Clauses with *fail to*	4
Purely syntactic	62
Word order	24
do-support	20
Inalienable possession	8
Clausal conjuncts	6
Tag questions	4
Purely Lexical	30
Sense specification	10
Idioms	20

Table 2: An overview of the data.

3.2 Contents

The current English-Swedish set contains 202 example sentences. The distribution on categories and phenomena is shown in Table 2.

Some sentences have been taken from the English-French challenge set, as they give rise to the same translation difficulty when translating into Swedish as they do for translating into French. Other sentences are made up but often based on sentences found in corpora that can be searched online such as the COCA corpus (Davies, 2018), the BYU-BNC (Davies, 2018), and the English-Swedish parallel UD corpora such as LinES and PUD (Nivre et al., 2018).

3.3 Evaluation

A thorough evaluation has not yet been undertaken, but we have made two pilot studies to get indicative answers to the following questions: (1) Are the sentences really challenging for current systems, and (2) Will different evaluators agree in their judgements? The first question was tested by randomly selecting 20 sentences from the full set and have

Systems	Yes	No	NA	Accuracy
Sys 1	10	9	1	0.50
Sys 2	9	10	1	0.45
Sys 3	9	11	-	0.45
Sys 4	10	9	1	0.50

Table 3: A pilot system evaluation using twenty randomly generated challenge sentences. 'Yes' means the translation is judged correct in the focused aspect, 'No' that it is not. Judgements by author.

Items	User1	User2	User3	Author
Q 1	NO	YES	NO	NO
Q 2	NO	YES	NO	NO
Q 3	YES	YES	YES	YES
Q 4	YES	YES	YES	NA
Q 5	NO	YES	YES	YES
Q 6	NO	NO	NO	NO
Q 7	YES	YES	YES	NO
Q 8	YES	YES	YES	YES
Q 9	YES	YES	NO	NO
Q10	YES	NO	NO	NO
Q11	YES	NA	YES	YES
Q12	YES	NO	NO	NO
Q13	NO	NO	NO	NO
Q14	NO	NO	NO	NO
Q15	YES	YES	YES	YES
Q16	YES	YES	NO	YES
Q17	YES	YES	YES	YES
Q18	YES	YES	YES	YES
Q19	YES	YES	YES	YES
Q20	YES	YES	YES	YES

Table 4: A pilot user evaluation.

them translated by four different online systems. The output has been judged by the author and is shown in Table 3. Judgements on one of the systems have also been provided by three other evaluators, yielding a higher accuracy (see Table 4). Still, the results clearly indicate that the label 'challenge set' is indeed appropriate.

To test the second question the output from one of the systems was given to three other people with Swedish as their mother tongue. They were given a web form and very little instruction to perform the task. The results are shown in Table 4. There was full agreement only on half of the items which means that the agreements are not as strong as could be hoped. Using majority voting 17 judgements can be seen to agree with those of the author. This indicates that evaluators require detailed instruction, and that it may be useful to provide instances of translations that should be judged as incorrect. Lists of such answers may also pave the way for automatic scoring. Table 5 shows some of the examples causing disagreement.

3.4 Availability

The challenge set will be available via a common repository such as GitHub. Contributions, in the form of new examples and reviews are very welcome.

References

A. Burchardt, V. Macketanz, J. Dehdari, G. Heigold, J-T. Peter and P. Williams. 2017. A Linguistic Evaluation of Rule-Based, Phrase-Based, and Neural MT Engines. *Prague Bulletin of Mathematical Linguistics*, 108(1):159–170.

M. Davies. 2018a. BYU-BNC. A Corpus based on the British National Corpus from Oxford University Press. https://corpus.byu.edu/bnc.

M. Davies. 2018b. The Corpus of Contemporary American English (COCA): 560 million words, 1990-present. https://corpus.byu.edu/coca.

P. Isabelle, C. Cherry, and G. Foster. 2017. A Challenge Set Approach to Evaluating Machine Translation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, (EMNLP '17), Copenhagen, Denmark, pages 2486–2496.

P. Isabelle and R. Kuhn. A Challenge Set for French → English Machine Translation. ArXiv:1806.02725v1 cs.CL.

M. King and K. Falkedal. 1990. Using Test Suites in Evaluation of Machine Translation Systems. In Proceedings of the 13th Conference on Computational Linguistics (COLING-90) - Volume 2, Helsinki, Finland, pages 211–216. doi = 10.3115/997939.997976.

V. Macketanz, A. Renlong, A. Burchardt and H. Uszkoreit. 2018. TQ-AutoTest An Automated Test Suite for (Machine) Translation Quality. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC'18), May, 7-12, Miyasaki, Japan. European Language Resources Association (ELRA).

J. Nivre et al. 2018. Universal Dependencies 2.2. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. http://hdl.handle.net/11234/1-2837.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, July 2002, pages 311–318. doi = 10.3115/1073083.1073135.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla and J. Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In Proceedings of Association for Machine Translation in the Americas (AMTA'06).

S. Stymne and L. Ahrenberg. 2012. On the practice of error analysis for machine translation evaluation. In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12, May 23-25, Istanbul, Turkey.

D. Vilar, J. Xu, L.F. D'Haro and H. Ney. 2006. Error Analysis of Machine Translation Output. In Proceedings of LREC06, Genoa, Italy, pages 697–702.

SRC	*[A] terrifying black lion crossed the road.*
SYS	*en skrämmande svart lejon korsade vägen.*
QUE	Does the translation of the marked determiner agree with its head noun?
ANS	1 YES 3 NO - NA
SRC	*Only two of the players seemed to be [ready].*
SYS	*bara två av spelarna verkade vara redo.*
QUE	Does the translation of the marked word agree with its head noun?
ANS	3 YES - NO 1 NA
SRC	*As always, the bastard [failed to] respond.*
SYS	*som alltid misslyckades den jäveln att svara.*
QUE	Is the meaning of the marked verb correctly rendered in the Swedish translation?
ANS	3 YES 1 NO - NA
SRC	*They told you to [put] the bottle on the table, didn't they?*
SYS	*de sa att du skulle lägga flaskan på bordet, eller hur?*
QUE	Is the meaning of the marked verb correctly rendered in the Swedish translation?
ANS	2 YES 2 NO - NA

Table 5: Responses to items Q1, Q4, Q7, and Q9.

Parameter Sharing in Multilingual Dependency Parsing

Miryam de Lhoneux

Department of Linguistics and Philology
Uppsala University

Abstract

The consensus is increasing that, in neural dependency parsing, sharing parameters between languages can be beneficial. However, 1) there is no consensus on what parameters to share and 2) sharing parameters is a strategy that has so far mostly been shown to be beneficial when working with related languages and it is unclear if it is beneficial to share parameters between unrelated languages. We present an evaluation of 27 different parameter sharing strategies across 10 languages, representing five pairs of related languages, each pair from a different language family. We evaluate those strategies when training parsers on the related pairs as well as when training parsers with the same languages mixed into unrelated pairs. We find that sharing transition classifier (typically a Multilayer Perceptron (MLP)) parameters helps in the related as well as in the unrelated case, whereas the usefulness of sharing parameters of Long Short-Term Memory networks (LSTM) representing words and/or characters is mostly beneficial in the related case. Based on these findings, we propose as future work a universal parsing architecture where the MLP is shared across all languages and word and character representations can be shared across language families when appropriate.

1. Introduction

The idea of sharing parameters between parsers of related languages goes back to early work in cross-lingual adaptation (Zeman and Resnik, 2008), and the idea has recently received a lot of interest in the context of neural dependency parsers (Duong et al., 2015; Ammar et al., 2016; Smith et al., 2018). Modern neural dependency parsers, however, use different sets of parameters for representation and scoring, and it is not clear what parameters it is best to share. In addition, the consensus is increasing that it is helpful to share parameters across related languages but there is little research showing the benefits of sharing parameters across unrelated languages in the context of neural dependency parsing, although Lynn et al. (2014) have shown that Indonesian was surprisingly particularly useful for Irish in the context of a statistical parser.

The Universal Dependencies (UD) project (Nivre et al., 2016), which is seeking to harmonize the annotation of dependency treebanks across languages, has seen a steady increase in languages that have a treebank in a common standard. Many of these languages are low resource and have small UD treebanks. It seems interesting to find out ways to leverage the wealth of information contained in these treebanks, especially for low resource languages.

2. Parameter Sharing in UUParser

UUParser (de Lhoneux et al., 2017a,b) consists of three sets of parameters; the parameters of the character-based LSTM, those of the word-based LSTM, and the parameters of the MLP that predicts transitions. The character-based LSTM produces representations for the word-based LSTM, which produces representations for the MLP. The Uppsala parser is a transition-based parser (Kiperwasser and Goldberg, 2016), adapted to the Universal Dependencies (UD) scheme,[1] and using the arc-hybrid transition system from Kuhlmann et al. (2011) extended with a SWAP transition and a static-dynamic oracle, as described in de Lhoneux et al. (2017b). The SWAP transition is used to generate non-projective dependency trees (Nivre, 2009).

Since our parser has three basic sets of model parameters, we consider sharing all combinations of those three sets. We also introduce two ways of sharing, namely, with or without the addition of a vector representing the language. This language embedding enables the model, in theory, to learn what to share between the two languages in question. Since for all three model parameter sets, we now have three options – not sharing, sharing, or sharing in the context of a language embedding – we are left with $3^3 = 27$ parameter sharing strategies; see Table 2.

We refer the reader to de Lhoneux et al. (2018) for a more extensive description of the parser and the different parameter strategies.

3. Experiments

Datasets The dataset characteristics are listed in Table 1. For all 10 languages, we use treebanks from the Universal Dependencies project. To keep the results comparable across language pairs, we down-sample the training set to the size of the smallest of our languages, Hebrew: we randomly sample 5000 sentences for each training set.

ISO	Lang	Tokens	Family	Word order
ar	Arabic	208,932	Semitic	VSO
he	Hebrew	161,685	Semitic	SVO
et	Estonian	60,393	Finnic	SVO
fi	Finnish	67,258	Finnic	SVO
hr	Croatian	109,965	Slavic	SVO
ru	Russian	90,170	Slavic	SVO
it	Italian	113,825	Romance	SVO
es	Spanish	154,844	Romance	SVO
nl	Dutch	75,796	Germanic	No dom. order
no	Norwegian	76,622	Germanic	SVO

Table 1: Dataset characteristics

[1] http://universaldependencies.org/

RELATED LANGUAGES

Model	C	W	S	ar	he	es	it	et	fi	nl	no	hr	ru	AV
MONO				76.3	80.2	83.7	83.3	70.4	70.8	77.3	80.8	76.8	82.3	78.2
LANGUAGE-BEST				76.6	80.6	84.4	84.8	72.8	72.9	79.6	82.1	78.0	82.9	79.5
BEST	✗	✓	ID	76.3	80.3	84.2	84.5	72.1	72.5	78.8	81.4	77.6	82.8	79.1
CHAR	✓	✗	✗	76.4	80.3	84.3	84.0	72.3	71.0	78.3	81.3	77.0	82.3	78.7
WORD	✗	✓	✗	76.3	79.9	83.9	84.4	72.4	71.3	77.4	80.7	76.9	82.5	78.6
STATE	✗	✗	✓	76.6	80.3	84.0	83.7	71.5	72.9	78.3	81.5	77.4	82.8	78.9
ALL	✓	✓	✓	76.2	80.1	84.0	84.2	72.1	71.4	78.7	81.1	77.0	82.5	78.7
SOFT	ID	ID	ID	76.3	79.9	84.1	84.4	72.1	71.3	79.6	81.4	77.1	82.5	78.9

UNRELATED LANGUAGES

Model	C	W	S	he	no	fi	hr	ru	es	it	et	nl	ar	average
MONO				80.2	80.8	70.8	76.8	82.3	83.7	83.3	70.4	77.3	76.3	78.2
LANGUAGE-BEST				80.5	81.5	71.9	77.6	82.9	84.0	84.3	72.5	78.7	76.5	78.9
BEST	✗	✗	✓	80.3	81.5	71.9	77.6	82.7	84.0	83.8	72.5	78.7	76.3	78.9
WORST	ID	ID	✗	79.8	80.6	69.2	76.7	81.4	83.8	83.2	69.4	76.6	76.0	77.7
CHAR	✓	✗	✗	80.1	80.9	71.4	76.8	82.9	83.9	84.3	70.9	78.0	76.5	78.6
WORD	✗	✓	✗	79.6	80.9	71.9	76.9	82.2	83.7	83.8	70.9	77.0	76.4	78.3
ALL	✓	✓	✓	80.5	80.9	69.8	76.6	82.3	83.7	84.0	70.6	77.4	76.2	78.2
SOFT	ID	ID	ID	79.8	80.5	70.1	76.6	82.1	83.9	83.8	70.6	77.2	76.3	78.1

Table 2: Performance on development data (LAS; in %) across select sharing strategies. MONO is our single-task baseline; LANGUAGE-BEST is using the best sharing strategy for each language (as evaluated on development data); BEST and WORST are the overall best and worst sharing strategies across languages; CHAR shares only the character-based LSTM parameters; WORD shares only the word-based LSTM parameters; ALL shares all parameters. ✓ refers to hard sharing, ID refers to soft sharing, using an embedding of the language ID and ✗ refers to not sharing.

Implementation details A flexible implementation of parameter strategies for UUParser was implemented in Dynet.[2] The code is publicly available.[3]

4. Results and discussion

A subset of our results on development sets are presented in Table 2. We refer the reader to the supplementary material of de Lhoneux et al. (2018) for the comprehensive tables of results. Our main observations when looking at results on related language pairs are: **(i)** that, generally, and as observed in previous work, *multi-task learning helps*: all different sharing strategies are on average better than the monolingual baselines, with minor (0.16 LAS points) to major (0.86 LAS points) average improvements; and **(ii)** that sharing the MLP seems to be overall a better strategy than not sharing it: the 10 best strategies share the MLP. Whereas the usefulness of sharing the MLP seems to be quite robust across language pairs, the usefulness of sharing word and character parameters seems more dependent on the language pairs. This reflects the linguistic intuition that character- and word-level LSTMs are highly sensitive to phonological and morphosyntactic differences such as word order, whereas the MLP learns to predict less idiosyncratic, hierarchical relations from relatively abstract representations of parser configurations.

Looking at results on unrelated language pairs, as expected, there is much less to be gained from sharing parameters. However, it is possible to improve the monolingual baseline by sharing some of the parameters. In general, sharing the MLP is still a helpful thing to do. It is most helpful to share the MLP and optionally one of the two other sets of parameters. Results are close to the monolingual baseline when everything is shared. Sharing word and character parameters but not the MLP hurts accuracy compared to the monolingual baseline.

[2] https://github.com/clab/dynet

[3] https://github.com/coastalcph/uuparser

5. Selective Sharing in Multilingual Parsing

Naseem et al. (2012) proposed to selectively share subsets of parameters of a parsing model across languages in the context of a probabilistic parser. This idea has not been explored in the context of neural dependency parsing, as far as we are aware. Since sharing the MLP seems to be a useful thing to do irrespective of language relatedness, we propose to construct a multilingual parser with many languages where the MLP is shared across all languages. This can be done via hard sharing or could be done via soft sharing with a language embedding like previously. We could alternatively or additionally construct a *language family* embedding for this soft sharing. Parameters of words and characters could then be shared within language families, when useful.

Acknowledgments

Thanks are due to the co-authors of the paper of the main work presented in this abstract: Johannes Bjerva, Isabelle Augenstein and Anders Søgaard. I also thank Joakim Nivre, Sara Stymne and Aaron Smith for discussions about this work.

References

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. More languages, one parser. In *TACL*.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low Resource Dependency Parsing: Cross-lingual Parameter Sharing in a Neural Network Parser. In *Proceedings of ACL*.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations. *TACL*, 4:313–327.

Marco Kuhlmann, Carlos Gómez-Rodríguez, and Giorgio Satta. 2011. Dynamic Programming Algorithms for Transition-Based Dependency Parsers. In *Proceedings of ACL*, pages 673–682, Portland, Oregon, USA.

Miryam de Lhoneux, Johannes Bjerva, Isabelle Augenstein, and Anders Søgaard. 2018. Parameter sharing between dependency parsers for related languages. In *Proceedings of EMNLP*.

Miryam de Lhoneux, Yan Shao, Ali Basirat, Eliyahu Kiperwasser, Sara Stymne, Yoav Goldberg, and Joakim Nivre. 2017a. From raw text to universal dependencies - look, no tags! In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 207–217, Vancouver, Canada.

Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2017b. Arc-Hybrid Non-Projective Dependency Parsing with a Static-Dynamic Oracle. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 99–104, Pisa, Italy.

Teresa Lynn, Jennifer Foster, Mark Dras, and Lamia Tounsi. 2014. Cross-lingual transfer parsing for low-resourced languages: An irish case study. In *Proceedings of the First Celtic Language Technology Workshop*, pages 41–49.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 629–637. Association for Computational Linguistics.

Joakim Nivre. 2009. Non-Projective Dependency Parsing in Expected Linear Time. In *Proceedings of ACL*, pages 351–359, Suntec, Singapore.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*.

Aaron Smith, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 2018. 82 Treebanks, 34 Models: Universal Dependency Parsing with Multi-Treebank Models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*.

Daniel Zeman and Philip Resnik. 2008. Cross-Language Parser Adaptation between Related Languages. In *IJCNLP*.

Language Technology and Early Signs of Cognitive Decline - Current Status of a Multimodal and Multidisciplinary Approach

Dimitrios Kokkinakis[1], Kristina Lundholm Fors[1], Kathleen C. Fraser[1,3], Charalambos Themistocleous[1,4], Marie Eckerström[2], Greta Horn[1]

[1]Department of Swedish, University of Gothenburg, Sweden
[2]Department of Psychiatry and Neurochemistry, University of Gothenburg, Sweden
[3]National Research Council Canada, Ottawa, Canada
[4]Department of Neurology, School of Medicine, Johns Hopkins University, Baltimore, USA
[1]first.last@svenska.gu.se; [2]first.last@neuro.gu.se

Abstract

The number of people with cognitive impairments, e.g. various types of dementia, has grown steadily on a global scale. To date, nearly all therapy interventions have failed to show significant benefits, perhaps because detection of the severity of the brain damage was made too late to be reversed with drug treatments. However, long before the clinical onset of symptoms of dementia, patients exhibit deficits in their oral and written communication and visual short-term memory, signs that can be objectively measured and serve as evidence to predict poor cognitive health in later life. The aim of this paper is to present a snapshot of our on-going experimental and analytical studies in this area that *could* lead to significant complementary knowledge for early detection of dementia. We aim to identify important linguistic markers that can be used as a complementary, early diagnostic, prognostic or screening tool for neurodegenerative pathologies such as Alzheimer's disease.

1. Introduction

Many types of dementia, and Alzheimer's disease (AD) in particular, are characterized by a decline in cognitive skills, memory, language and executive function which seriously affects people's everyday activities. Early diagnosis of dementia has important clinical significance and impact on society, considering the fact that the total estimated worldwide cost of dementia in 2015 was 818 billion US$ and it is estimated that by 2018, dementia will become a "trillion dollar disease" (Prince, 2015). In this paper, we present a summary of our current multidisciplinary, experimental and analytical studies on how multi-modal data resources and a set of language related measures can be used for the development, experimentation and evaluation of classification algorithms to be used for identifying early linguistic symptoms of cognitive decline in the elderly. Early detection of dementia is important for a number of reasons, such as giving the person access to interventions, medications and currently available treatment strategies to delay the progression of the disease; participate in (cost effective) clinical trials, and allowing the individual and families time to prepare e.g. to handle financial and legal issues.

2. Population and ethical considerations

All collected samples are produced by Swedish speakers, recruited from the ongoing Gothenburg MCI study (Wallin et al., 2016), after obtaining written consent approved by the local ethics committee (206-16, 2016 and T021-18, 2018). Table 1 shows some demographic characteristics of the population. Here, HC refers to Healthy Controls, SCI to subjects with Subjective Cognitive Impairment (a pre-symptomatic and predominantly benign condition) and MCI to subjects with Mild Cognitive Impairment (a decline in cognitive abilities that is associated with an increased risk of developing dementia).

	HC (35)	**SCI (23)**	**MCI (31)**
Sex (M/F)	13/22	9/14	15/16
Age (years)	68 (7.3)	66.3 (6.9)	70.1 (5.6)
Education (years)	13.3(3.4)	16.1(2.1)	14.1(3.6)
MMSE (max 30)	29.6(0.6)	29.5(0.9)	28.2(1.4)

Table 1: Demographic information; the MMSE (Mini Mental State Exam) is a general screening test of cognitive status and has a maximum score of 30.

3. Tasks

We collected data in two phases (2016 and 2018). At both occasions we let participants describe the Cookie Theft picture from the Boston Diagnostic Aphasia Examination (Goodglass et al., 2001) to acquire spoken language material. Another speech task involves reading aloud a short text from the International Reading Speed Texts collection, IReST (Trauzettel-Klosinski et al., 2012), presented on a computer screen. One more IREST text is read silently; and in both cases we combine reading with eye-tracking recording. After each reading task, participants answer five multiple choice questions about what they have read. For tracking eye movements we use EyeLink 1000 Desktop Mount with monocular eye tracking with head stabilization and a real-time sample access of 1000Hz. During the first collection phase (2016) we manually performed verbatim transcriptions of the audio recordings, while in 2018 we intend to use a newly developed speech-to-text system (Themistocleous and Kokkinakis, 2018) which can automate the transcription step.

We decided to record in two phases since we also want to analyze whether there are longitudinal differences between the two audio and eye-tracking recordings, and at which level and magnitude. The second phase also includes three

new tasks, namely:

- a semantic verbal fluency task, where participants have to produce as many words as possible from a category (animals) in a given time (60 seconds). This task tests executive function and semantic memory; cf. (Wolters, et al., 2016).
- a complex planning task that tests the ability to identify, organize and carry out (complex) steps and elements that are required to achieve a goal cf. (Fleming, 2014).
- a map task: a spontaneous speech production/semi-structured conversation in which the participants are encouraged to talk about a predefined, cooperative task oriented topic; cf. (Anderson, et al., 1991).

4. Experiments

4.1 Eye-tracking

Machine learning analysis of eye-tracking data for the detection of MCI is described in (Fraser et al., 2017). The comparison of two experimental configurations (reading aloud vs. reading silently), as well as two methods of combining information from the two trials (concatenation vs. merging) could distinguish participants with and without cognitive impairment with 86% accuracy in the best case. Notably, tracking eye movements while the participant reads silently provides more diagnostic information than when reading aloud. Merging data from the two trials led to an increase in classification accuracy. Eye-tracking holds promise as a method for detecting the earliest stages of cognitive decline. Compared to state-of-the-art, (Biondi et al., 2017) reported 89,8% accuracy by using a set of features with AD patients. The results reported in (Fraser et al., 2017) use exactly the same feature set but with MCI patients.

4.2 Eye-voice span

We investigated the process of reading aloud, by exploring the eye-voice span in subjects with and without cognitive impairment. The eye-voice span is a measurement of the temporal and spatial organization between the eye and the voice, and it is affected by for example working memory and automaticity, but also by the familiarity and length of words. The aim of the study (Lundholm Fors et al., 2018b) was to identify potential differences in the reading processes and evaluate whether these differences can be used to discriminate between the two groups. In previous work, differences between eye movements when reading in healthy controls and subjects with cognitive impairments have been identified, and it has been shown that subjects with Alzheimers disease show impairments when reading aloud, specifically with regards to speech and articulation rate. We performed a quantitative and qualitative analysis of the reading process in the subjects, focusing both on general measures of eye-voice span, but also specifically on instances of hesitation and mistakes in the speech, and the correlated eye movements. We found that participants with cognitive impairment had a significantly shorter eye-voice span than healthy controls.

4.3 Syntactic analysis of transcriptions

The syntactic complexity of transcribed data of the Cookie Theft picture was reported in (Lundholm Fors et al., 2018a). Using a random forest classifier (18 features) we achieved a mean F-score of 0.64 for distinguishing MCI and SCI groups; 0.63 of distinguishing the MCI and HC groups and only 0.45 when distinguishing between the SCI and HC groups (an expected result given that the SCI participants perform as well as the controls on all neuropsychological tests). Our results indicated that syntactic features are moderately successful at distinguishing the participant groups. While none of the features differed significantly between the groups, there are some trends; namely an increase in the number of false starts, an increase in the proportion of main clauses where the main verb is nonfinite, and a reduction in the proportion of main clauses where the verb is finite in the MCI group.

4.4 Acoustic properties, speech segments and prosody

Speech motor control and the complex coordination of articulatory movements for the production of speech sounds is the output of linguistic processes that takes place in the brain. In this process, cortical structures determine the association of meanings to phonological representations. When aspects of this process become impaired due to neural degeneration, we expect that the impairment will be manifested in speech production; the latter is accessible from the acoustic properties of the corresponding speech signals. Thus by understanding the acoustics of speech, we may identify aspects of cognitive impairment that affect cortical areas and the associated linguistic processes.

We study the effects of MCI on both the abstract realization of phonetic targets and on speech dynamics. Targets manifest invariant properties of speech production, whereas dynamics explain acoustic patterns that change in time. Specifically, we have focused at the segmental and suprasegmental level of speech. At the segmental level, we have analyzed the production of vowels and focused on vowels' formant frequencies and vowel duration. Vowel formants are a good candidate for identifying physiological properties of speakers (Themistocleous, 2017). In (Themistocleous et al., 2018a), we compared the vowels of MCI and HC participants and showed that participants with MCI produce longer vowels than healthy controls, which indicates an overall slower speech in MCI participants. We also found that male MCI participants produce longer vowels than female MCI and HC participants. Although this phenomenon may correspond to the sociophonetic manifestation of gender in Swedish speech, it may also designate other gender specific properties in the realization of MCI and thus it requires more research.

Prosody is an important linguistic structure that binds constituents of speech into groups (e.g. syllables, prosodic words); it manifests different melodies with respect to speech acts; it designates the boundaries of phrases; and it marks constituents as more prominent or less prominent (Themistocleous, 2014). Speech melody also manifests physical and emotional aspects of speech. To this end, we have analyzed the fundamental frequency (F0), which is the acoustic manifestation of intonation and the temporal as-

pects of speech productions.

In a current study, we trained deep neural networks on acoustic features and achieved a high classification accuracy of MCI and HC, which is close to 75% (Themistocleous et al., 2018b). Studies such as these are promising because they bring us closer to understanding the phonemic and phonetic realization of MCI and AD speech and their corresponding neurophysiological and neuropsychological properties. We have also investigated pause length and articulation rate in participants with MCI and healthy controls, and found that participants with MCI tend to produce longer pauses and present with a lower articulation rate (Lundholm Fors et al., 2018c), which is congruent with previous research about persons with Alzheimer's disease.

4.5 Multilinguality

Clinical language samples are hard to come by due to the sensitive nature of the data. Data collection is expensive, time-consuming, and limited by various factors such as the need to respect ethical guidelines and participant consent when sharing and storing data. In (Fraser et al., 2018) we have considered how we can use external data to boost the performance of automatic and machine learning methods when faced with the challenge of small data. We analyze the information content (Croisile, et al., 1996) in narrative speech samples from individuals with MCI in both English and Swedish, using a combination of supervised and unsupervised learning. Information units were extracted using topic models trained on word embeddings in monolingual and multilingual spaces. Results showed that the multilingual approach leads to significantly better classification accuracies than training on the target language alone. Ultimately, we were able to distinguish MCI speakers from healthy older adults with accuracies of up to 72% (Swedish) and 63% (English) on the basis of information content alone.

4.6 Multimodality

Figuring out the best way to combine and learn from all available data is an important step in successfully applying machine learning in the clinical domain. We have started to experiment with methods for combining information from the different modalities (i.e. features from the audio recordings, text transcripts, eye-tracking scanpaths, and responses to comprehension questions) to improve the detection of MCI. In current work, we are exploring questions such as: Is it more effective to combine features from all modalities in an early fusion paradigm, or to combine information from each mode and task in a hierarchical structure? Do we obtain similar information from, for example, speech features extracted from the Cookie Theft task and speech features extracted from the reading task? And do predictions based on individual task performance correlate with the standardized neuropsychological test scores? Preliminary results suggest that we can improve both accuracy and interpretability by combining information from the different speech and language tasks.

5. Future work

The ultimate goal of this research is to create an automated differential diagnostic tool, which will enable the differentiation of MCI from conditions with similar symptoms. Such a system will require more data from a larger population, yet our current findings, based on data collected during phase one, are encouraging and do provide a promising step towards this purpose. This is not a trivial enterprise since MCI is a complex and heterogeneous stage (Bäckman, et al., 2005), and performance on neuropsychiatric tests of persons with MCI overlap greatly with the performance of healthy controls. Still, international research suggests that clues about the neural degeneration process could manifest years before a diagnosis can be established.

As previously outlined, data collection is performed in two phases. The second phase (during 2018) is an identical repetition of all the tests performed during the first collection phase in 2016. However, during phase two we introduced three new tests since during the analysis of the data collected during phase one, it became apparent that the data we have available and analyzed would need to be supplemented in order to achieve an even higher level of predictive value. In future work we would like to increase the number of features and to e.g. compare spoken and written Cookie Theft descriptions (Kokkinakis et al., 2018). We also plan to incorporate the syntactic features with measures of semantics, information content, discourse-level processing (Toledo, et al., 2018), and acoustic/phonetic production to gain a more complete picture of speech in MCI (see also Section 4.4). We hope that the addition of the new data, collected during 2018 (not analyzed at the time this paper was written), can provide us with more dialogue-like data and not just monologue as in the first data collection phase.

We also plan to perform more cross-lingual studies of prosodic, acoustic and other linguistic features and, also, to examine any potential correlations between the predictions and the available standardized neuropsychological test scores and specific language tasks conducted in the Gothenburg MCI study such as the Boston Naming Test (Kaplan et al., 1983).

Acknowledgements

This work is supported by *Riksbankens Jubileumsfond - The Swedish Foundation for Humanities and Social Sciences*, through the grant agreement no: NHS 14-1761:1.

References

A. Anderson, et al. 1991. The HCRC Map Task Corpus. *Language and Speech*, 34, pp. 351–366.

J. Biondi, G. Fernandez, S. Castro, and O. Agamenonni. 2017. Eye-movement behavior identification for AD diagnosis *arXiv:1702.00837.*

L. Bäckman, et al. 2005. Cognitive impairment in preclinical Alzheimers disease: A meta-analysis. *Neuropsychology.* 19(4): 520—531

B. Croisile, et al. 1996. Comparative Study of Oral and Written Picture Description in Patients with Alzheimers Disease. *Brain Lang.* 53(1):1—19.

V.B. Fleming. 2014. Early Detection of Cognitive-Linguistic Change Associated With Mild Cognitive Impairment. *Communication Disorders Quarterly*. 35:3, pp. 146–157. Sage.

K.C. Fraser, K. Lundholm Fors, D. Kokkinakis and A. Nordlund. 2017. An analysis of eye-movements during reading for the detection of mild cognitive impairment. *Conference on Empirical Methods in Natural Language Processing (EMNLP).*, pp. 1027–1037, Copenhagen, Denmark c 2017 ACL.

K.C. Fraser, K. Lundholm Fors and D. Kokkinakis. 2018. Multilingual word embeddings for the assessment of narrative speech in mild cognitive impairment. *J of Computer Speech and Language*. Elsevier, doi.org/10.1016/j.csl.2018.07.005

H. Goodglass, E. Kaplan and B. Barresi. 2001. *Boston Diagnostic Aphasia Examination (3rd ed.).* Philadelphia: Lippencott, Williams & Wilkins.

E. Kaplan, H. Goodglass and S. Weintraub. 1983. *Boston Naming Test* Philadelphia: Lea & Febiger.

D. Kokkinakis, K. Lundholm Fors, K.C. Fraser and A. Nordlund. 2018. A Swedish Cookie-Theft Corpus *11th edition of the Language Resources and Evaluation Conference (LREC)*, pp. 1252—1258, Miyazaki, Japan.

K. Lundholm Fors, K.C. Fraser and D. Kokkinakis. 2018a. Automated Syntactic Analysis of Language Abilities in Persons with Mild and Subjective Cognitive Impairment. *Studies in health technology and informatics.*, vol 247, pp. 705–709, Gothenburg, Sweden.

K. Lundholm Fors, K.C. Fraser and D. Kokkinakis. 2018b. Eye-voice span in adults with mild cognitive impairment (MCI) and healthy controls. *10th European Congress of Speech and Language Therapy (CPLOL)*, Cascais, Portugal.

M. Prince et al. 2015. World Alzheimer Report 2015 - The Global Impact of Dementia - An analysis of prevalence, incidence, cost and trends. *Alzheimers Disease International (ADI)*, 34(4): 555–596. London.

C. Themistocleous. 2014. Edge-tone effects and prosodic domain effects on final lengthening. *Linguistic Variation*, 14:1, 129–160.

C. Themistocleous. 2017. Classifying linguistic and dialectal information from vowel acoustic parameters. *Speech Communication* 94, 13 – 22. DOI: 10.1016/j.specom.2017.05.003.

C. Themistocleous and D. Kokkinakis. 2018. THEMIS-SV: Automatic classification of language disorders from speech signals. *4th European Stroke Organisation Conference*, Gothenburg, Sweden.

C. Themistocleous, M. Eckerström and D. Kokkinakis. 2018b. Identification of Mild Cognitive Impairment from Speech in Swedish using Deep Sequential Neural Networks *Frontiers in Neurology.*, *under review*.

C. Themistocleous, M. Eckerström D. Kokkinakis, K.C. Fraser and K. Lundholm Fors. 2018a. Effects of cognitive Impairment on vowel duration. *9th Tutorial & Research Workshop on Experimental Linguistics (Exling2018)*, 105–108, Paris, France.

K. Lundholm Fors, K.C. Fraser, C. Themistocleous and D. Kokkinakis. 2018c. Prosodic Features As Potential Markers of Linguistic and Cognitive Deterioration in Mild Cognitive Impairment. Alzheimer's Association International Conference Chicago, U.S.A.

C.M. Toledo, et al. 2018. Analysis of macrolinguistic aspects of narratives from individuals with Alzheimers disease, mild cognitive impairment, and no cognitive impairment *Alzheimer's & Dementia: Diagnosis, Assessment & Disease Monitoring.* 10: 31–40. Elsevier

S. Trauzettel-Klosinski, et al. 2012. Standardized Assessment of Reading Performance: The New International Reading Speed Texts IReST. *Investigative Opthalmology & Visual Science.*, 53(9):5452–61.

K. Wallin, et al. 2016. The Gothenburg MCI study: Design and distribution of Alzheimers disease and subcortical vascular disease diagnoses from baseline to 6-year follow-up. *J of Cerebral Blood Flow & Metabolism.*, 36(1): 114–131. Sage.

M. Wolters, et al. 2016. Prosodic and Linguistic Analysis of Semantic Fluency Data: A Window into Speech Production and Cognition *Interspeech.*, pp. 2085–2089, San Francisco, USA.

The Eukalyptus Treebank of Written Swedish

Yvonne Adesam, Gerlof Bouma, Richard Johansson, Lars Borin, Markus Forsberg

Språkbanken
Department of Swedish
University of Gothenburg
firstname.lastname@gu.se

1. Introduction

Treebanks – texts with added information about parts-of-speech and syntactic structures – are instrumental for developing various annotation tools, such as part-of-speech taggers and syntactic parsers, and valuable for empirical language research. Sweden has a long history of creating treebanks, starting with the 1970s 'MAMBA' treebank (Teleman, 1974), which has been reused for later resources (Nivre et al., 2006; Nivre et al., 2008; de Marneffe et al., 2014). The latter also includes some material not based on MAMBA. The Stockholm–Umeå Corpus (SUC) (Ejerhed et al., 1992), consisting of about a million tokens with manually checked base forms, part-of-speech tags and morphological information, has been the de facto standard Swedish tagged resource.

We now release a newly developed treebank, the Eukalyptus treebank of written Swedish. Its development was motivated by the need for a freely available treebank (unlike SUC), with more and newer texts, and an annotation scheme in line with a modern view on descriptive Swedish grammar. The newly developed annotation scheme will also be employed in the annotation tools of Språkbanken (Borin et al., 2016), and applied to the billion word corpora available through Språkbanken (https://spraakbanken.gu.se/korp).

2. The Eukalyptus Treebank

The rest of this paper will give a brief overview of the first release of the Eukalyptus treebank.

2.1 The Texts

The texts chosen for the Eukalyptus treebank are public domain, and contain around 100.000 tokens. These are equally distributed over five types of text.

The first subcorpus was taken from Europarl (Koehn, 2002), a corpus of European parliament proceedings. This part is interesting as it contains normalized spoken language, as well as translations, and also allows for future extension of exploring the same text in different languages. The second subcorpus consists of blog texts, taken from the SIC corpus (Östling, 2013). Blog texts exhibit a variety of language phenomena, often associated with informal language. The third was taken from Wikipedia (sv.wikipedia.org), and contains articles on 16 different topics. The fourth contains excerpts from four public domain novels (las-en-bok.com). The fifth and final subcorpus is made up of articles from the newspaper Arbetaren (arbetaren.se), as well as newsletters from the Government Offices of Sweden (regeringen.se) and Digisam (digisam.se), a Swedish collaboration for digitization of cultural heritage.

None of the subcorpora is intended to represent a genre on their own, however, together they show a wide variety of written language. They range from professionally edited to more personal texts, from formal to informal, and informative to entertaining.

2.2 Segmentation

SUC follows a rather strict tokenization scheme where space separates tokens (although abbreviations such as *t ex* 'e.g.' are tokenized as *t_ex*). We, however, allow tokens to contain spaces, as in *Mont Blanc-tunneln* 'Mont Blanc tunnel.DEF'. In addition, our treatment of multi-word units (Section 2.4) reduces the importance of tokenization for the syntactic annotation layer.

Sentence segmentation mainly follows the orthographic hints given by the authors. Especially in the blog texts, additional criteria like perceived syntactic coherence were needed to supplement this strategy.

The 100 000 tokens are distributed over close to 5 800 sentences. This gives an average of just over 17 tokens per sentence, although this varies between the text types, with an average of under 14 for the novels and an average of 24 for the Europarl sentences.

2.3 Annotation

The data has been manually annotated with parts-of-speech, morphological features, syntactic structure, and sense information. The part-of-speech tagset is loosely based on the SUC tagset (Ejerhed et al., 1992) but adapted to make it more in line with *Svenska Akademiens grammatik* (Teleman et al., 1999). The syntactic structure contains both phrases and functions (Adesam et al., 2015a). Phrases (possibly discontinuous) follow rules restricting what type of lexical material may head them, linking the part-of-speech categories to phrase categories through projection rules.

The annotation has been manually corrected on the basis of semi-automatic error detection. We intend to address remaining inconsistencies in future releases of the treebank.

The syntactic structure now contains close to 56 000 nonterminals (phrases), with more than 138 000 edges (tree-formed dependencies), and almost 11 000 secondary edges (various kinds of shared dependencies). In addition, around 70 000 tokens, mainly content words, have been annotated with sense identifiers (Johansson et al., 2016) using the sense

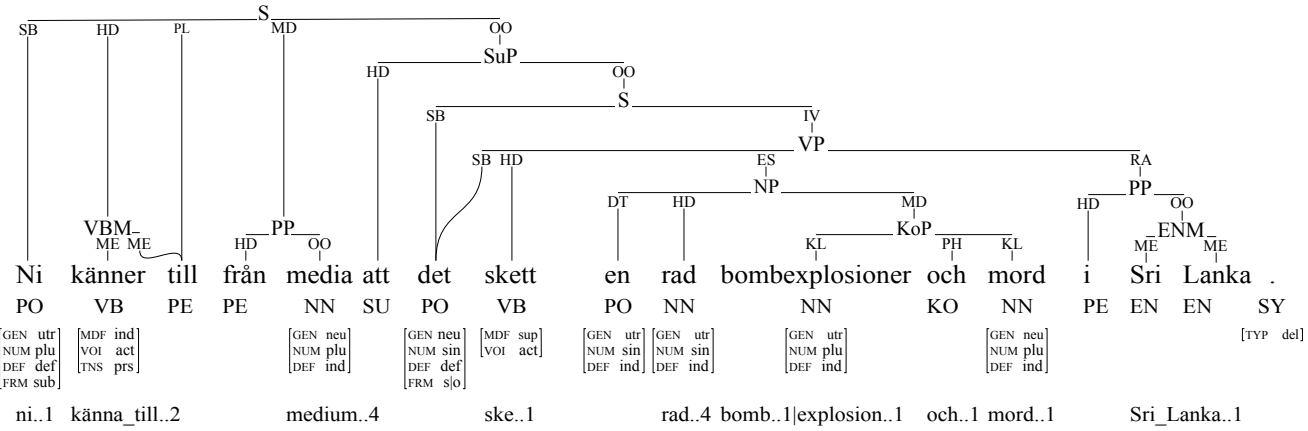


Figure 1: An annotated tree.

inventory defined by the SALDO lexicon (Borin et al., 2013). An example of a fully annotated sentence is given in Fig. 1.

2.4 Multi-word Units

The integration of multi-word units in the lexical and syntactic annotation levels has received considerable attention in the Eukalyptus annotation scheme (Adesam et al., 2015b). Wherever possible, multi-word units are analyzed syntactically as regular phrases with an additional multi-word part-of-speech node indicating their special status (*känner till* in Fig. 1). For multi-word units that are not easily incorporated into 'regular' syntax, we allow a flat annotation under the multi-word part-of-speech node (*Sri Lanka* in Fig. 1). In either case the multi-word node serves as an anchor for a word sense label.

3. Conclusions

We have presented the first release of the Eukalyptus treebank of written Swedish, made available through Språkbanken under a creative-commons license (CC-BY). Future releases will include more quality checks, as well as mappings to the SUC tag set and the Universal Dependency format.

Acknowledgements

The Eukalyptus treebank was developed within the Koala project, funded by Riksbankens Jubilemsfond, grant number In13-0320:1.

References

Yvonne Adesam, Gerlof Bouma, and Richard Johansson. 2015a. Defining the Eukalyptus forest – the Koala treebank of Swedish. In *Proceedings of NODALIDA*, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

Yvonne Adesam, Gerlof Bouma, and Richard Johansson. 2015b. Multiwords, word senses and multiword senses in the eukalyptus treebank of written Swedish. In *Proceedings of TLT*, Warsaw, Poland.

Lars Borin, Markus Forsberg, and Lennart Lönngren. 2013. SALDO: a touch of yin to WordNet's yang. *Language Resources and Evaluation*, 47(4):1191–1211.

Lars Borin, Markus Forsberg, Martin Hammarstedt, Dan Rosén, Roland Schäfer, and Anne Schumacher. 2016. Sparv: Språkbanken's corpus annotation pipeline infrastructure. In *Proceedings of SLTC*.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher Manning. 2014. Universal stanford dependencies: A cross-linguistic typology. In *Proceedings of LREC*.

Eva Ejerhed, Gunnel Källgren, Ola Wennstedt, and Magnus Åström. 1992. The linguistic annotation system of the Stockholm-Umeå corpus project - description and guidelines. Technical report, Department of Linguistics, Umeå University.

Richard Johansson, Yvonne Adesam, Gerlof Bouma, and Karin Hedberg. 2016. A multi-domain corpus of Swedish word sense annotation. In *Proceedings of LREC*, Portorož, Slovenia.

Philipp Koehn. 2002. Europarl: A multilingual corpus for evaluation of machine translation.

Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In *Proceedings of LREC*.

Joakim Nivre, Beáta Megyesi, Sofia Gustafson-Capková, Filip Salomonsson, and Bengt Dahlqvist. 2008. Cultivating a Swedish treebank. In *Resourceful Language Technology: Festschrift in Honor of Anna Sågvall Hein*. Uppsala University, Department of Linguistics and Philology.

Robert Östling. 2013. Stagger: an open-source part of speech tagger for Swedish. *Northern European Journal of Language Technology*, 3:1–18.

Ulf Teleman, Staffan Hellberg, and Erik Andersson. 1999. *Svenska Akademiens Grammatik*. Svenska Akademien, Stockholm.

Ulf Teleman. 1974. *Manual för grammatisk beskrivning av talad och skriven svenska*. Studentlitteratur, Lund.

The Interplay Between Loss Functions and Structural Restrictions in Semantic Dependency Parsing

Robin Kurtz, Marco Kuhlmann

Department of Computer and Information Science
Linköping University
robin.kurtz@liu.se, marco.kuhlmann@liu.se

Abstract

Semantic dependency parsing, the task of parsing to bilexical directed acyclic graphs representing the semantic structure of a sentence, has recently gained traction due to the ability of these graphs to express language phenomena that cannot be modelled with more restrictive tree structures. The general line of attack on semantic dependency parsing has been to adapt methods and ideas from the more mature field of syntactic parsing, in particular the design of effective inference algorithms for structurally restricted search spaces, and the use of neural networks as the learning component. In this paper we study the interplay between structural restrictions and network loss functions. We uncover a problem that arises when using the usual structured hinge loss in combination with a structural constraint, and propose a modified loss function to address this problem.

1. Introduction

Semantic dependency parsing is the task of mapping a sentence into a formal representation of its meaning in the form of a bilexical directed acylic graph, rather than a tree as in syntactic parsing. The added expressivity of graphs allows for an intuitive representation of relational semantics and analyses of argument sharing, coordination, quantification, and others (Oepen et al., 2014; Oepen et al., 2015).

Semantic dependency parsing is algorithmically more challenging than syntactic dependency parsing because the search space opened up by removing the tree constraint is much larger. Recent work has therefore focused on designing algorithms that are expressive enough to cover the data, and yet restricted enough to support efficient inference. While straightforward restrictions of the search space like that to general directed acyclic graphs lead to intractable parsing (Schluter, 2014), polynomial-time decoding algorithms have been proposed for more complex structural restrictions, including the noncrossing constraint (Kuhlmann and Jonsson, 2015), bounded treewidth (Gildea et al., 2017), and the one-endpoint-crossing property (Kummerfeld and Klein, 2017; Cao et al., 2017; Kurtz and Kuhlmann, 2017).

The state of the art in semantic dependency parsing is defined by systems powered by neural networks. Recurrent neural networks (RNNs) in particular have proven themselves to be effective, being able to naturally encode context information. However, as we show in this paper, combining neural networks with a structurally restricted parsing algorithm is not always straightforward. We compare the learning behaviour of two algorithms, the noncrossing algorithm of Kuhlmann and Jonsson (2015) and a minimally restricted algorithm used in recent work (Zhang et al., 2017; Dozat and Manning, 2018), when these algorithms are combined with different loss functions. We find that, while the restricted algorithm (but not the unrestricted one) benefits from a structured loss function, it is important to bound the loss that can be suffered from dependency arcs that are disallowed under the structural constraint in order to prevent the learning algorithm from diverging.

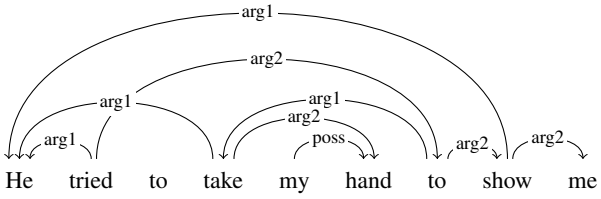


Figure 1: A sample dependency graph from the SDP dataset (Flickinger et al., 2016, DM #41526060). Note that this graph violates the noncrossing constraint, as the arcs *take* → *He* and *tried* → *to* cross each other.

2. Background

Given a natural language sentence $x = x_1, \ldots, x_n$, we define a *dependency graph* for x as an arc-labelled directed acyclic graph whose vertices correspond, one-to-one, to the words x_i. Placing the vertices of a dependency graph on a line in the plane following the left-to-right ordering of the sentence, we draw the arcs as semi-circles in the half-plane above the line, using arrows to denote the direction from head to dependent. Each arc of the dependency graph has a label taken from a finite set, describing the relation of the head and the dependent vertices. An example of a dependency graph is shown in Figure 1. In this graph, the subject of the sentence, *He*, is the primary argument (arg1) for the three different verbs *tried*, *take* and *show*, showcasing the structure's ability to assign one and the same actor to multiple actions. The example graph is taken from the standard data set for semantic dependency parsing (SDP), released by Flickinger et al. (2016), which contains graphs in three different target representations or 'flavours', respectively derived from DeepBank (DM, Oepen and Lønning (2006; Ivanova et al. (2012)), the predicate–argument structures computed by the Enju parser (PAS, Miyao (2006)), and the tectogrammatical layer of the Prague Dependency Treebank (PSD, Hajic et al. (2012)). We use the SDP data set for all experiments reported in this paper.

Generalizing the approach of McDonald et al. (2005), semantic dependency parsing can be cast as *maximum subgraph parsing*, where the predicted graph $\hat{y}$ for the sentence x is chosen as that graph y from a set $Y(x)$ of candidate graphs which maximises the scoring function S:

$$\hat{y} = \arg\max_{y \in Y(x)} S(x, y) \quad (1)$$

The scoring function S is computed via a sum of scores for local substructures, which in our case are single arcs. This is known as the *arc-factored model*.

3. Parsing Setup

In the general framework just described, a parser consists of essentially two components: a learning component that learns the arc-specific scores, and an inference algorithm or decoder that solves the optimization problem (1) for a set of candidate graphs $Y(x)$. We now describe the specific instantiations for these choices that we study in this paper.

Decoding Algorithms We compare two different inference algorithms: the structurally restricted algorithm of Kuhlmann and Jonsson (2015) and an essentially unrestricted algorithm very similar to the one recently used by Zhang et al. (2017) and Dozat and Manning (2018). The algorithm of Kuhlmann and Jonsson (2015) finds the highest-scoring *noncrossing* dependency graph for the input sentence x, where a graph is called noncrossing if it does not contain any crossing arcs. (Two arcs are said to cross when their semi-circles intersect at an internal point.) Note that the example graph in Figure 1 violates the noncrossing constraint. The unrestricted algorithm simply adds all arcs with a positive score to the graph, excluding only arcs with self-loops, and choosing the arc with the highest score for when two arcs share the same start- and endpoints.

Learning Component For the learning component we use the same neural network structure as Kiperwasser and Goldberg (2016) for graph-based dependency tree parsing. Embeddings for tokens and part-of-speech (POS) tags are concatenated and serve as inputs for a bidirectional long-short term memory network (BiLSTM) with two layers, where the outputs of the first BiLSTM are used as inputs for the second layer. We replace the regular LSTM cells with a variant using coupled input and forget gates, and peephole connections, which has shown to yield good results. Every potential arc is scored using a multi-layer perceptron (MLP) with one hidden layer producing scalar outputs. These outputs are then collected in a score matrix which is used as the input for the decoder. Labels are scored using a separate MLP. Both the arc-scoring and the label-scoring network share the same underlying RNN structure.

4. Experiments and Results

The purpose of our experiments is to study the interplay between the structural restriction in the decoding algorithm and the loss function of the network. In all experiments, the label-scoring module is trained on the gold-standard arcs, parallel to the arc-scoring module; the losses for both modules are then simply added to one overall loss.

Word embedding	100
POS tag embedding	25
hidden units in arc-MLP	100
hidden units in label-MLP	100
BiLSTM Layers	2
BiLSTM dimensions (hidden/output)	125/125
word dropout	0.25

Table 1: Hyper-parameters for the network and training.

The implementation of the parsers was done in Python 3, using DyNet 2.04 (Neubig et al., 2017), minimally modifiying the code of Kiperwasser and Goldberg (2016). In each experiment, for each flavour of the SDP data we train models using the Adam optimizer (Kingma and Ba, 2014) for 20 epochs using hyperparameters as in Table 1 and DyNet's default parameters if not stated otherwise. For each epoch we evaluate on the development set, choosing the best model according to labelled F-score, which is then used for final evaluation on the in-domain and out-of domain test sets. Each experiment is repeated three times; the reported scores are the averaged results over the three runs.

4.1 Structural Hinge Loss

Perhaps the most obvious choice for the loss function for the arc-scoring module is a structured hinge loss defined in terms of the unlabelled graph $\hat{y}$ predicted for the sentence x. We follow Peng et al. (2017) and minimise a slightly modified loss function

$$L(x, \hat{y}; y) = \max_{\hat{y} \in Y(x)} \{S(x, \hat{y}) + c(y, \hat{y})\} - S(x, y) \quad (2)$$

where $c(y, \hat{y})$ is the weighted (unlabelled) Hamming distance between the gold-standard graph y and the predicted graph $\hat{y}$. The goal with using this loss function is to score the gold graph y higher than the best-scoring incorrect graph. Using the weighted Hamming distance we encourage recall over precision, weighting false negatives (gold arcs that were not predicted by the parser) by 0.6 and false positives (arcs falsely predicted) by 0.4.

With this loss function, the results for the noncrossing decoder are considerably worse than those for the unrestricted decoder (see Table 2, column 'hinge'). In fact, when looking at the intermediate results on the development set, we can see that the loss function does not approach zero but instead diverges towards negative infinity.

Ignoring the Hamming distance, the loss (2) consists of two sums: the sum of scores for arcs in the predicted graph, subtracted with the sum of scores for arcs in the gold graph. True positives (correctly predicted arcs) cancel each other out between the two sums, leaving behind the sums of scores for false positives and false negatives. The unrestricted decoder simply adds all arcs with positive scores, meaning that false postives will have scores greater and false negatives scores less than zero. During training these are then decreased and increased, respectively, leading to more true positives that cancel each other out and thus a loss function approaching its minimum at zero. However, in the case of the noncrossing decoder there are false negatives that are not added to the predicted graph due to them crossing other arcs, regardless of their scores. These arcs, and their crossing

System	Data	DM			PAS			PSD		
		hinge	hinge′	cross	hinge	hinge′	cross	hinge	hinge′	cross
unrestricted	id	89.1	89.1	87.9	91.5	91.5	90.2	76.6	76.6	74.8
	ood	83.9	83.9	82.2	87.1	87.1	85.8	74.0	74.0	72.7
noncrossing	id	85.5	88.1	87.3	85.3	90.1	89.5	73.6	75.3	74.1
	ood	80.4	82.9	81.8	80.4	85.9	85.2	71.3	73.0	72.3
Dozat et al.	id			91.4			93.9			79.1
	ood			86.9			90.8			77.5

Table 2: Averaged F-Scores for the in-domain and out-of-domain test sets for the two algorithms paired with the unmodified and the modified hinge loss and the cross entropy loss functions. For comparison we add the scores Dozat and Manning (2018) reported for their most basic system.

counterparts, continually have their scores increased during training, leading to more and more negative losses and unnecessary updates, disrupting the overall learning process.

4.2 Modified Hinge Loss

In order to avoid the unnecessary updates that arise due to the structural constraint, we first took a very simplistic approach and forcefully restored the original function's intended effect by clipping all loss values below zero. This modification improves the results by stopping learning whenever the loss would become negative due to zero gradients. At the same time however we lose the opportunity to learn from false positives, when the scores of false negatives outweigh the scores of false positives. Instead, we decided to use an upper bound for scores of false negatives that prevents updates only for those arcs, whenever the bound is reached. This results in a hinge loss where the hinge is not at zero, but rather is dependent on the upper score limit and the amount of arcs eliminated by the constraint. We set this upper limit at 3, which is the upper end of the interval of the arc scores learned by the unrestricted decoder. Our modification increases the performance for the noncrossing decoder (see Table 2, column 'hinge′'), albeit its results still lag behind the unrestriced algorithm, which we attribute to the decrease in coverage. (Less than two thirds of the graphs in the data set are noncrossing.)

4.3 Binary Cross Entropy Loss

For comparison we also implement the approach of Dozat and Manning (2018), who obtain strong results by computing the loss for every possible arc, not only the arcs in the output of the decoder, rendering the decoder obsolete during training. (They still use the decoder at test time.) More specifically, they define a binary cross entropy loss on the arcs and a softmax cross entropy loss on the labels. The results for the noncrossing parser are higher with this loss function than with the unmodified structural hinge loss, showing that training the weights ignoring the structural constraints avoids the previously described disruptive training behaviour (see Table 2, column 'cross'). Given that these results are still significantly lower than with the modified hinge loss, this could mean that it is still useful, in the presence of a structural constraint, to base the loss computation on the actual output. It is however worth noting that the network of Dozat and Manning (2018) is considerably larger than the one used in our experiments.

5. Conclusion

In this paper we have shown that when parsing to semantic dependency graphs using neural networks, enforcing a structural constraint has consequences for the choice of an appropriate loss function. If the structural constraint is to be used during training, the structured hinge loss gave the best results, but only after having been modified in order to handle arcs ruled out by the structural constraint. While models using the structurally informed loss outperform models using the binary cross entropy loss of Dozat and Manning (2018), which ignores the constraint, a significant benefit of the latter systems is that they do not have to decode during training, allowing them to more efficiently train the network and therefore also to increase its size. When aiming for performance, the question how to combine both the structurally independent with the structurally informed loss, needs to be further explored.

References

Junjie Cao, Sheng Huang, Weiwei Sun, and Xiaojun Wan. 2017. Parsing to 1-Endpoint-Crossing, Pagenumber-2 Graphs. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1:2110–2120.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but More Accurate Semantic Dependency Parsing. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2:484–490.

Dan Flickinger, Jan Hajič, Angelina Ivanova, Marco Kuhlmann, Yusuke Miyao, Stephan Oepen, and Daniel Zeman. 2016. SDP 2014 & 2015: Broad Coverage Semantic Dependency Parsing LDC2016T10.

Daniel Gildea, Giorgio Satta, and Xiaochang Peng. 2017. Cache Transition Systems for Graph Parsing.

Jan Hajic, Eva Hajicová, Jarmila Panevová, Petr Sgall, Ondřej Bojar, Silvie Cinková, Eva Fućiková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiŕi Semecký, Jana Sindlerová, Jan Štěpánek, Josef Toman, Zdenka Uresová, and Zdenek Zabokrtský. 2012. Announcing Prague Czech-English Dependency Treebank 2.0. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3153–3160, İstanbul, Turkey. European Language Resources Association.

Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan

Flickinger. 2012. Who Did What to Whom?: A Contrastive Study of Syntacto-semantic Dependencies. In *Proceedings of the Sixth Linguistic Annotation Workshop*, LAW VI '12, pages 2–11, Stroudsburg, PA, USA. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations. *Transactions of the Association of Computational Linguistics*, 4(1):313–327.

Marco Kuhlmann and Peter Jonsson. 2015. Parsing to Noncrossing Dependency Graphs. *Transactions of the Association of Computational Linguistics*, 3(1):559–570.

Jonathan K. Kummerfeld and Dan Klein. 2017. Parsing with Traces: An $O(n^4)$ Algorithm and a Structural Representation. *Transactions of the Association for Computational Linguistics*, 5.

Robin Kurtz and Marco Kuhlmann. 2017. Exploiting Structure in Parsing to 1-Endpoint-Crossing Graphs. *Proceedings of the 15th International Conference on Parsing Technologies*, pages 78–87.

Ryan T. McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-Projective Dependency Parsing using Spanning Tree Algorithms. In *HLT/EMNLP*, pages 523–530. The Association for Computational Linguistics.

Yusuke Miyao. 2006. *From Linguistic Theory to Syntactic Analysis: Corpus-Oriented Grammar Development and Feature Forest Model*. Ph.D. thesis.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, Paul Michel, Yusuke Oda, Matthew Richardson, Naomi Saphra, Swabha Swayamdipta, and Pengcheng Yin. 2017. DyNet: The Dynamic Neural Network Toolkit. *arXiv preprint arXiv:1701.03980*.

Stephan Oepen and Jan Tore Lønning. 2006. Discriminant-Based MRS Banking. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC-2006)*, Genoa, Italy, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L06-1214.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Yi Zhang. 2014. SemEval 2014 Task 8: Broad-Coverage Semantic Dependency Parsing. *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinkova, Dan Flickinger, Jan Hajic, and Zdenka Uresova. 2015. SemEval 2015 Task 18: Broad-Coverage Semantic Dependency Parsing. *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 915–926.

Hao Peng, Sam Thomson, and Noah A. Smith. 2017. Deep Multitask Learning for Semantic Dependency Parsing. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1:2037–2048.

Natalie Schluter. 2014. On maximum spanning DAG algorithms for semantic DAG parsing. *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pages 61–65.

Xingxing Zhang, Jianpeng Cheng, and Mirella Lapata. 2017. Dependency Parsing as Head Selection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 665–676, Valencia, Spain. Association for Computational Linguistics.

Modular Mechanistic Networks for Computational Modelling of Spatial Descriptions

Simon Dobnik[1] **and John D. Kelleher**[2]

[1]CLASP and FLOV, University of Gothenburg, Sweden
simon.dobnik@gu.se

[2]School of Computing, Dublin Institute of Technology, Ireland
john.d.kelleher@dit.ie

Abstract

We argue current deep learning approaches to modelling of spatial language in generating image captions have shortcomings and that the multiplicity of factors that influence spatial language invites a modular approach where the solution can be built in a piece-wise manner and then integrated. We call this approach where deep learning is assisted with domain knowledge expressed as modules that are trained on data a top-down or mechanistic approach to otherwise a bottom-up phenomenological approach.

In recent years deep learning approaches have made significant breakthroughs. An exciting aspect of deep learning is learning inter/multi-modal representations from data that includes discrete information (e.g. words) and continuous representations (e.g. word embeddings and visual features), such as those used in automatic image captioning systems. A number of shortcomings with current deep learning architectures have been identified with respect to their application to spatial language such as "the chair is to the left and close to the table" or "go down the corridor until the large painting on your right, then turn left". For example, in (Kelleher and Dobnik, 2017) we argue that contemporary image captioning networks have been configured in a way that they capture visual properties of objects ("what" in terms of (Landau and Jackendoff, 1993; Landau, 2016)) rather than spatial relations between them ("where"). Consequently, within the captions generated by these systems the relation between the preposition and the object is not grounded in geometric representation of space but only in the linguistic sequences through the decoder language model where the co-occurrence of particular words in a sequence is estimated.[1] This is because neural networks are typically used as generalised learning mechanisms that learn with as little supervision through architecture design as possible. We call this data-driven approach a *bottom-up* or *phenomenological approach*. The problem is that the chosen architecture may not be optimal for every aspect of the cognitive representations that we want to learn.

We do not argue that language model is not informative for predicting spatial relations since these are not just about geometric space. In addition to (i) scene geometry (Logan and Sadler, 1996; Dobnik and Åstbom, 2017) they also rely on (ii) perspective and perceptual context (Kelleher and Kruijff, 2005; Dobnik et al., 2015), (iii) functional world knowledge about dynamic kinematic routines of objects (Coventry et al., 2001), and (iv) interaction between agents through language and dialogue and with the environment through perception (Clark, 1996; Fernández et al., 2011; Schutte et al., 2017; Dobnik and de Graaf, 2017). In (Dobnik and Kelleher, 2013; Dobnik and Kelleher, 2014; Dobnik et al., 2018) we show that a language model is useful in predicting functional relations between objects. The system can learn something about object interaction without visually observing these objects and such knowledge is used as background knowledge when generating and interpreting spatial descriptions. The information expressed in a language model or visual features of the scene is therefore just one of the modalities that must be taken into account. This provides a challenge for computational modelling of spatial descriptions because (i) it is difficult to provide and integrate that kind of knowledge and (ii) its contextual underspecification. A computational system taking into account these meaning components in the context would be able to understand and generate better, more human-like, spatial descriptions and engage in more efficient communication in the domain of situated agents and humans. Furthermore, it could exploit the synergies between different knowledge sources to compensate missing knowledge in one source from another (Steels and Loetzsch, 2009; Skočaj et al., 2011; Schutte et al., 2017).

We argue (Dobnik and Kelleher, 2017) that the multiplicity of factors that influence spatial language invites a modular approach where the solution can be built in a piece-wise manner and then integrated (Feldman et al., 1988; Feldman, 1989; Regier, 1996; Andreas et al., 2016; Johnson et al., 2017). We call this approach where deep learning is assisted with domain knowledge expressed as modules that are trained on data a *top-down* or *mechanistic approach*. One challenge to spatial language is the lack of an overarching theory explaining how these different factors should be integrated but (Herskovits, 1987) and (Coventry and Garrod, 2005) appear to be promising candidates. Early work on neural networks includes some examples of neural models that could provide a basis for the design of specific modules. For example, (Regier, 1996) captures geometric factors and paths of motion. The system in (Coventry et al.,

[1]The over-reliance of deep learning models on the language model has been criticised recently for example, in relation to visual question answering and an attempts have been made to make the systems give a greater weight to images in predicting the caption, for example by balancing different answers in datasets (Agrawal et al., 2017).

2005) processes dynamic visual scenes containing three objects: a teapot pouring water into a cup and the network learns to optimise, for each temporal snapshot of a scene, the appropriateness score of a spatial description obtained in subject experiments. The idea behind these experiments is that descriptions such as *over* and *above* are sensitive to a different degree of geometric and functional properties of a scene, the latter arising from the functional interactions between objects. The model is split into three modules: (i) a vision processing module that deals with detection of objects from image sequences using an attention mechanism, (ii) an Elman recurrent network that learns the dynamics of the attended objects in the scene over time, and (iii) a dual feed-forward vision and language network to which representations from the hidden layer of the Elman network are fed and which learns how to predict the appropriateness score of each description for each temporal configuration of objects. Each module of this network is dedicated to a particular task: (i) to recognition of objects, (ii) to follow motion of attended objects in time and (iii) to integration of the attended object locations with language to predict the appropriateness score, factors that have been identified to be relevant for computational modelling of spatial language and cognition in previous experimental work (Coventry et al., 2001). The example shows the effectiveness of representing networks as modules and their possibility of joint training where individual modules constrain each other.

The model could be extended in several ways. For example, contemporary CNNs and RNNs could be used which have become standard in neural modelling of vision and language due to their state-of-the-art performance. Secondly, the approach is trained on a small dataset of artificially generated images of a single interactive configuration of three objects. An open question is how the model scales on a large corpus of image descriptions (Krishna et al., 2017) where considerable noise is added: the appearance and location of objects may be distorted by the angle at which the image is taken. Furthermore, there are no complete temporal sequences of objects and the corpora typically does not contain human judgement scores on how appropriate a description is given an image. Finally, (Coventry et al., 2005)'s model integrates three modalities used in spatial cognition, but as we have seen there are several others. An important aspect is grounded linguistic interaction and adaptation between agents. For example, (Lazaridou et al., 2016) describe a system where two networks are trained to perform referential games (dialogue games performed over some visual scene) between two agents. In this context, the agents develop their own language interactively. An open research question is whether parameters such frame of reference intended by the speaker of a description could also be learned this way.

References

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2017. Don't just assume; look and answer: Overcoming priors for visual question answering. *arXiv*, arXiv:1712.00377 [cs.CV].

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Learning to compose neural networks for question answering. In *Proceedings of NAACL-HLT 2016*, pages 1545–1554, San Diego, California, June 12-17. Association for Computational Linguistics.

Herbert H. Clark. 1996. *Using language*. Cambridge University Press, Cambridge.

Kenny Coventry and Simon Garrod. 2005. Spatial prepositions and the functional geometric framework. towards a classification of extra-geometric influences. In Laura Anne Carlson and Emile van der Zee, editors, *Functional features in language and space: insights from perception, categorization, and development*, volume 2, pages 149–162. Oxford University Press.

Kenny R. Coventry, Mercè Prat-Sala, and Lynn Richards. 2001. The interplay between geometry and function in the apprehension of Over, Under, Above and Below. *Journal of Memory and Language*, 44(3):376–398.

Kenny R. Coventry, Angelo Cangelosi, Rohanna Rajapakse, Alison Bacon, Stephen Newstead, Dan Joyce, and Lynn V. Richards. 2005. Spatial prepositions and vague quantifiers: Implementing the functional geometric framework. In Christian Freksa, Markus Knauff, Bernd Krieg-Brückner, Bernhard Nebel, and Thomas Barkowsky, editors, *Spatial Cognition IV. Reasoning, Action, Interaction*, volume 3343 of *Lecture Notes in Computer Science*, pages 98–110. Springer Berlin Heidelberg.

Simon Dobnik and Amelie Åstbom. 2017. (Perceptual) grounding as interaction. In Volha Petukhova and Ye Tian, editors, *Proceedings of Saardial – Semdial 2017: The 21st Workshop on the Semantics and Pragmatics of Dialogue*, pages 17–26, Saarbrücken, Germany, August 15–17.

Simon Dobnik and Erik de Graaf. 2017. KILLE: a framework for situated agents for learning language through interaction. In Jörg Tiedemann and Nina Tahmasebi, editors, *Proceedings of the 21st Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 162–171, Gothenburg, Sweden, 22–24 May. Northern European Association for Language Technology (NEALT), Association for Computational Linguistics.

Simon Dobnik and John D. Kelleher. 2013. Towards an automatic identification of functional and geometric spatial prepositions. In *Proceedings of PRE-CogSsci 2013: Production of referring expressions – bridging the gap between cognitive and computational approaches to reference*, pages 1–6, Berlin, Germany, 31 July.

Simon Dobnik and John D. Kelleher. 2014. Exploration of functional semantics of prepositions from corpora of descriptions of visual scenes. In *Proceedings of the Third V&L Net Workshop on Vision and Language*, pages 33–37, Dublin, Ireland, August. Dublin City University and the Association for Computational Linguistics.

Simon Dobnik and John D. Kelleher. 2017. Modular mechanistic networks: On bridging mechanistic and phenomenological models with deep neural networks in natural language processing. In Simon Dobnik and Shalom Lappin, editors, *Proceedings of the Conference on Logic and Machine Learning in Natural Language (LaML 2017), Gothenburg, 12–13 June 2017*, volume 1 of *CLASP Papers in Computational Linguistics*, pages

1–11, Gothenburg, Sweden, November. Department of Philosophy, Linguistics and Theory of Science (FLOV), University of Gothenburg, CLASP, Centre for Language and Studies in Probability.

Simon Dobnik, Christine Howes, and John D. Kelleher. 2015. Changing perspective: Local alignment of reference frames in dialogue. In Christine Howes and Staffan Larsson, editors, *Proceedings of goDIAL – Semdial 2015: The 19th Workshop on the Semantics and Pragmatics of Dialogue*, pages 24–32, Gothenburg, Sweden, 24–26th August.

Simon Dobnik, Mehdi Ghanimifard, and John D. Kelleher. 2018. Exploring the functional and geometric bias of spatial relations using neural language models. In *Proceedings of the First International Workshop on Spatial Language Understanding (SpLU 2018) at 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*, pages 1–11, New Orleans, Louisiana, USA, June 1–6. Association for Computational Linguistics.

J. A. Feldman, M. A. Fanty, and N. H. Goodard. 1988. Computing with structured neural networks. *Computer*, 21(3):91–103, March.

Jerome A. Feldman. 1989. Structured neural networks in nature and in computer science. In Rolf Eckmiller and Christoph v.d. Malsburg, editors, *Neural Computers*, pages 17–21. Springer, Berlin, Heidelberg.

Raquel Fernández, Staffan Larsson, Robin Cooper, Jonathan Ginzburg, and David Schlangen. 2011. Reciprocal learning via dialogue interaction: Challenges and prospects. In *Proceedings of the IJCAI 2011 Workshop on Agents Learning Interactively from Human Teachers (ALIHT)*, Barcelona, Catalonia, Spain.

Annette Herskovits. 1987. *Language and Spatial Cognition*. Cambridge University Press, New York, NY, USA.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Fei-Fei Li, C. Lawrence Zitnick, and Ross B. Girshick. 2017. Inferring and executing programs for visual reasoning. *arXiv*, arXiv:1705.03633v1 [cs.CV]:1–13.

John D. Kelleher and Simon Dobnik. 2017. What is not where: the challenge of integrating spatial representations into deep learning architectures. In *CLASP Papers in Computational Linguistics: Proceedings of the Conference on Logic and Machine Learning in Natural Language (LaML 2017)*, volume 1, pages 41–52, Gothenburg, Sweden, 12–13 June.

John D. Kelleher and Geert-Jan M. Kruijff. 2005. A context-dependent algorithm for generating locative expressions in physically situated environments. In Graham Wilcock, Kristiina Jokinen, Chris Mellish, and Ehud Reiter, editors, *Proceedings of the Tenth European Workshop on Natural Language Generation (ENLG-05)*, pages 1–7, Aberdeen, Scotland, August 8–10. Association for Computational Linguistics.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, May.

Barbara Landau and Ray Jackendoff. 1993. "what" and "where" in spatial language and spatial cognition. *Behavioral and Brain Sciences*, 16(2):217–238, 255–265.

Barbara Landau. 2016. Update on "what" and "where" in spatial language: A new division of labor for spatial terms. *Cognitive Science*, 41(2):321–350.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2016. Multi-agent cooperation and the emergence of (natural) language. *arXiv*, arXiv:1612.07182v2 [cs.CL]:1–11.

Gordon D. Logan and Daniel D. Sadler. 1996. A computational analysis of the apprehension of spatial relations. In Paul Bloom, Mary A. Peterson, Lynn Nadel, and Merrill F. Garrett, editors, *Language and Space*, pages 493–530. MIT Press, Cambridge, MA.

Terry Regier. 1996. *The human semantic potential: spatial language and constrained connectionism*. MIT Press, Cambridge, Massachusetts, London, England.

Niels Schutte, Brian Mac Namee, and John D. Kelleher. 2017. Robot perception errors and human resolution strategies in situated human–robot dialogue. *Advanced Robotics*, 31(5):243–257.

Danijel Skočaj, Matej Kristan, Alen Vrečko, Marko Mahnič, Miroslav Janíček, Geert-Jan M. Kruijff, Marc Hanheide, Nick Hawes, Thomas Keller, Michael Zillich, and Kai Zhou. 2011. A system for interactive learning in dialogue with a tutor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2011*, San Francisco, CA, USA, 25-30 September.

Luc Steels and Martin Loetzsch. 2009. Perspective alignment in spatial language. In Kenny R. Coventry, Thora Tenbrink, and John. A. Bateman, editors, *Spatial Language and Dialogue*. Oxford University Press.

Probability or change in probability?

Cheikh Bamba Dione & Christer Johansson
Dept. of Linguistic, Literary and Aesthetic Studies
University of Bergen, Norway
Dione.Bamba@uib.no, Christer.Johansson@uib.no

Abstract

Research has had a long-standing interest in estimation of objective word frequencies and transition probabilities between linguistic symbols at various linguistic levels such as letter, phoneme and word transitions, as well as transitions between parts-of-speech tags and syntactic nodes. Such probabilities have been shown to be of much practical value for developing natural language processing tools.
However, there are other problems in linguistics where the use of frequency estimates has not been so successful. For example, if we want to find out the antecedent of a pronoun the most frequent option is that a candidate word is not an antecedent. This makes it difficult to base a process on probabilistic choice. It is also the case that people do not have good intuitions for probabilities and often use some alternative logic for comparing alternatives, as is the case for the well-known *Conjunction Fallacy*, where people often value the probability of two outcomes as more 'probable' than just one of the outcomes, if that outcome is more salient, or *better supported* by context.
We (i.e. humans) often understand that a context is motivated by a reason. We are inherently more interested in *change in probability* rather than objective probability, and we will therefore select any choice that increases probability most. One example is adding 'banana' to a choice between 'kiwi' and 'wiki', which will select for 'kiwi' even if 'wiki' still is by far the most 'likely' choice from frequency estimates. This will be investigated in this article, with examples for compounding (or not) of words, and for selecting the literal or idiomatic meaning of multiword expressions. We assume that adding context words will affect relative probability. The effect is mediated by how strongly context correlates with the items under investigation, and evaluated by relative effect size estimated from deviance from statistical independence.

1. Introduction & Background

One of the simplest and most basic rules of probability, called the conjunction rule, states that the probability (p) of two events occurring together (in *conjunction*) cannot exceed the probability of either one occurring alone. Formally, for two events A and B, we have p (A ∩ B) =< p(A) and p(A ∩ B) =< p(B). To test whether decision-makers abide by the conjunction rule, Tversky and Kahneman (1983) asked subjects to rank the likelihoods of certain conclusions that can be drawn from hypothetical personality sketches of fictitious individuals. Subjects were given the following personality profile and asked to identify which of the two alternatives below the profile was more probable (Tversky and Kahneman, 1983, p. 297).

> *Linda is 31 years old, single, outspoken, and very bright. She majored in philosophy. As a student, she was deeply concerned with issues of discrimination and social justice, and also participated in anti-nuclear demonstrations.*
>
> (a) Linda is a bank teller.
>
> (b) Linda is a bank teller and is active in the feminist movement.

Most of the respondents (85 %) indicated that (b) is more likely than (a), thus violating the conjunction rule. By assigning higher probability to the conjunction than its constituents, most of the respondents were subject to conjunction fallacy, and this fallacy indicates how we select relevant information from a linguistic context by using change in probability due to context.

Testing for an association between context words (*single, bright, outspoken, philosophy, concerned, discrimination, and demonstrations*) and either *feminist* or *teller* (as in bank teller) using the word2vec model (Mikolov et al., 2013) found that all were more positively correlated with *'feminist'*, except *'bright'*, which was minimally more associated with *'teller'*. *'Outspoken'* had the highest correlation with feminist.
An alternative explanation of the Linda effect could therefore be that the context highly activates the term 'feminist', but deactivates 'bank teller'.

We will argue that this effect of context is present in many situations where we judge the most *'likely'* alternative, for example judging compounding (Johansson 2017) and the idiomaticity of multi-word expressions (Dione & Johansson 2018). We use a measure of deviation from statistical independence (Johansson 2017) to mark statistical perplexity.

2. Previous work

In previous works, various tasks were handled successfully by looking at the pointwise effect size (*'serendipity'*, cf. Johansson 2017, Dione & Johansson 2018). Johansson (2017) used the serendipity measure to investigate compounding (and decompounding) when words are written together (e.g. *'toothbrush'*) or apart (e.g. *'apple sauce'*). Serendipity relies on effect size rather than statistical significance, and signals that particular observations or events deviate from expectations of statistical independence. Unlike statistical significance, serendipity is insensitive to the sample size. Furthermore, it may more closely mirror some human intuitions on the preferred alternative.

The *odds ratio* is similar to the serendipity measure in that it also reacts to information provided by context words. For example, the odds ratio in the kiwi/wiki example of Johansson (2017) shows that knowing *'banana'* increases the chance for *'kiwi'* more than 6 times, while it almost halves the chance that it is *'wiki',* which was always the most frequent choice. This means that it would be tempting to select *kiwi,* even if it is objectively an unlikely choice. This is similar to the *Conjunction Fallacy* in that it reflects a more or less innate tendency to value *change in probability* higher than the absolute probability.

In related work, Dione and Johansson (2018) present a model that investigates the role of context words for selecting between the literal and idiomatic meanings of *multiword expressions* (MWE). The model combines frequency estimates and serendipity to identify and mark patterns as either over- or under-represented compared to statistical independency. It became clear that the results has an advantage stemming from *term specificity,* as modeled by the Google search engine, since more specific context words had a tendency to give higher, rather than lower, frequencies compared to a baseline without context words if they were positively correlated with other words in the query. The statistical effect size thus cooperates with term specificity through frequency estimates delivered by the Google search engine. Thus correlations and anti-correlations of word patterns in contexts help to select more relevant documents.

The *Linda-effect* discussed above shows that "documents about feminists are rarely also about bank tellers, and vice versa, so they are highly anti-correlated" (Dione and Johansson, 2018, p. 157). If we consider the personality profile given above as a search query (which describes Linda as a (feminist) activist), such a query would heavily select documents where the term *'bank teller'* is relatively rare and *'feminist'* relatively frequent. One of the context words used in the personality profile (or query) is *outspoken*. Using the serendipity measure, Dione and Johansson (2018) could show that *'bank teller'* in the context of *outspoken* is considered less expected and should thus be avoided, and *feminist + 'bank teller'* is more frequent than expected in that context and is chosen because its chance of occurrence is higher compared to baseline.

Context words in a paragraph of text can be selected automatically based on word similarity functions as provided by publicly available resources, for example the Gensim project (Řehůřek and Sojka, 2010). One such resource is the *word2vec* function (Mikolov et al., 2013), a toolkit enabling the training and use of pre-trained word embeddings. Word2vec makes use of two different learning models: the Continuous Bag-of-Words (CBOW) model and the Continuous Skip-Gram model. CBOW aims to predict the *center word* w_0 based on its surrounding words within a text window. For instance, for a window size WS=3, the surrounding words of w0 are w-3;w-2;w-1;w1;w2;w3. The prediction of the centre word is measured by summing vectors of the surrounding words. In contrast, the Skip-Gram model takes a centre word and a window of context (neighbor) words and tries to predict *context words* within the window size for each centre word. Both of these models are shallow neural networks, which map word(s) to a target vector, which may be one or more words. In passing, it can be noted that the word2vec model (ibid.) is completely local in the sense that it does not rely on global statistics. Rather, it iteratively runs through the corpus, considers a couple of words at a time and tries to predict the centre word from its surrounding words (in the CBOW model) or the surrounding words from the centre word (in the Skip-Gram model).

A similarity function can partly replace the estimated Google frequency for the task of selecting how an expression fits with a context, and enable us to compare patterns in different contexts. The contribution of the search engine is coverage, and staying closer to attested examples. Neural networks are prone to either over-generalisations or over-learning (failing to find relevant patterns due to low match between training samples and test samples).

As an example we will go through a German expression *'ins Wasser gefallen'*, which can literally mean 'fell into water' or be used idiomatically to indicate that something failed to happen.

3. Discussion: When a child falls into water the inauguration might be cancelled

We introduce a mathematical model for how context can be used to estimate perplexity and expectation. Multiword expressions show many good examples.

A multiword expression has a context. Through a search engine it is possible to estimate how many documents contain the expression and how many also contain the context. Context words are content words that occur in close proximity and also have high term specificity, i.e., they are words that do not occur in most documents.

One example is the German multiword expression *"ins Wasser fallen"*, which literally means *"fall in water"*, but it is also used to state that something failed to happen.

Table 1 compares estimated document frequencies (June 6, 2018) for two patterns "*Das Kind ist ins Wasser gefallen*" (The child has fallen into the water) and "*Die Eröffnung ist in Wasser gefallen*" (The opening is cancelled) and what happens to frequencies when we add context words that are congruent with the literal meaning (*baden*, *schwimmen*, *ertrinken*; to bathe, swim and drown).
In parentheses, we see the serendipity measure, which indicates that '*Eröffnung*' is *over-represented* without context and *under-represented* (-9.59) with the context. Similarity can be estimated through word2vec (Mikolov et al. 2013, Řehůřek & Sojka 2010), but one restriction we have used is to only use similarity between pairs of words.

X	X+MWE	+CONTEXT
das Kind	10200 (-4.77)	22800 (**3.17**)
die Eröffnung	7320 (**14.44**)	3580 (-9.59)

Table 1. Estimated document frequencies

As an example, *'Kind'* and *'Wasser'* have a similarity of 0.3487, while the similarity between *'Eröffnung'* and *'Wasser'* is 0.2165. Similarity is consistently lower for *'Eröffnung'* compared to *'Kind'* for all the context words (*Wasser, fallen, baden, schwimmen, ertrinken*). The opposite is often true for contexts of events that can fail to happen, where context similar to for example *'Eröffnung'* is not similar to *'Wasser'* or *'fallen'*. However, as Katz & Giesbrecht (2006:17) noted "[...] in the newspaper genre, highly idiomatic expressions [...] were often used in their idiomatic sense [...] particularly frequently in contexts in which elements of the literal meaning were also present".

We argue that estimating document frequencies for patterns gives other possibilities than a general similarity function, such as the possibility to use longer patterns and estimating the effect of context words. We have noticed that frequencies are often higher when we add positively associated context words, which is similar to how people react to the *Conjunction Fallacy*, which shows that both we as humans and search engines are sensitive to context and assume that context is there for a reason.

We have noticed that higher term specificity may lead search engines to retrieve more documents. One example is shown in Table 1, where *adding* context retrieved more documents for *'Kind'* (but not for *'Eröffnung'* which is anti-correlated with the context).
This is of course sensible for a search engine to use when ranking the documents to be retrieved, and we consider this *'a feature not a bug'*. However, it raises the question if there is an objectively optimal way to include term specificity for retrieving an optimal set of documents that are *about* the search query. It is likely that all speakers will have differing intuitions, based on their experience, but the coarse patterns will be caught by search engine mechanisms.

4. Future Research

We are investigating if the use of search engines may be replaced by similarity through word2vec, with the aim of linking reasoning and similarity judgements. Recent research (Trueblood et al. 2014) also considers this possibility, but uses the framework of quantum probability theory. In their model, judging the conjunction means to project all relevant features into the subspaces from a common state vector. The Linda effect may then arise from sequentially evaluating first the most likely predicate (i.e., feminism) and only after that evaluate the second predicate (i.e., bank teller).

References

C.B. Dione and C. Johansson. 2018. Modeling Non-Compositional Expressions using a Search Engine. In *proceedings of the 8th International Conference on Awareness Science and Technology (iCAST)*. IEEE, Fukuoka, Japan. https://doi.org/10.29007/4jl9

C. Johansson. 2017. A word or two? In V. Rosén & K. De Smedt (Eds.), *The very model of a modern linguist — in honor of Helge Dyvik,* (pp. 112–126), Bergen Language and Linguistics Studies, 8(1). http://dx.doi.org/10.15845/bells.v8i1

G. Katz and E. Giesbrecht, 2006. Automatic identification of non-compositional multi-word expressions using latent semantic analysis, in *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*. Association for Computational Linguistics, pp. 12–19.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, vol. abs/1301.3781, 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

A. Tversky and D. Kahneman. 1983. Extensional versus intuitive reasoning: The conjunction fallacy in probability judgement. *Psychological Review*, vol. 90, no. 4, pp. 293–315.

R. Řehůřek and P. Sojka. 2010. Software Framework for Topic Modeling with Large Corpora, in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50. [Online]. Available: http://is.muni.cz/publication/884893/en

J. S. Trueblood, E. M. Pothos and J. R. Busemeyer. 2014. Quantum probability theory as a common framework for reasoning and similarity. *Frontiers in Psychology,* 5(322). https://doi.org/10.3389/fpsyg.2014.00322

Profiling Domain Specificity of Specialized Web Corpora using Burstiness Explorations and Open Issues

Marina Santini, Wiktor Strandqvist, Arne Jönsson

RISE Research Institutes of Sweden
marina.santini@ri.se, wiktor.strandqvist@gmail.com, arne.jonsson@ri.se

Abstract

In this paper we describe an approach to profile the domain specificity of specialized web corpora in Swedish. The proposed approach is based on burstiness. Burstiness is a statistical measure that identifies words with uneven distribution across the documents of a corpus. We apply burstiness to two medical web corpora that have different size and different domain granularity. Results are promising and show that burstiness is an appropriate measure to profile the domain specificity when matched against reference lists (gold standards) that represent the target domains. However, further research is needed to find adequate evaluation metrics, less empirical cut-off points and more principled gold standard design.

1. Introduction

Web corpora are valuable textual resources widely exploited in Language Technology. Leveraging on the web for corpus creation is a well-established idea because bootstrapping corpora from the web is fast and inexpensive. While texts in traditional corpora are hand-picked from several media and agreed upon by a number of experts, web corpora are built with documents available on the web at the time of corpus bootstapping. Traditional corpora are carefully curated and annotated to preserve the original traits of the selected texts, while web corpora can be noisy in several respects, e.g. they might contain damaged characters, problematic symbols, inconsistent punctuation or ungrammatical texts. In short, traditional corpora and web corpora represent different approaches to corpus construction and use. Arguably, traditional corpora and web corpora are complementary and allow for a wide spectrum of possible linguistic, empirical and computational studies and experiments. The unique and unprecedented potential of web corpora is that they can promptly and inexpensively account for virtually any domain, topic, genre, register, sublanguage, style and emotional connotation, since the web itself is a mine of linguistic and textual varieties.

While bootstrapping a web corpus is common practice (many tools exist, either based on crawling or on search engine queries), the validation of web corpora is still a grey area. With the investigations described in this paper, we would like to contribute to the discussion by adding a new perspective to web corpus evaluation. Normally, corpora can be assessed according to several parameters, for instance corpus balance, corpus representativeness, corpus quality, corpus size, and similar. In this complex scenario, we single out one aspect, namely domain specificity, and test whether a statistical measure like burstiness can help profile and quantify it given a reference domain. The long-term goal is to find a suitable metric that would help assess whether one corpus is more domain-specific than another corpus. This information would speed up any post-editing of specialized web corpora by reducing manual intervention.

Here ”domain” is defined as the ”subject field” or ”area” in which a web document is used. Domain specificity, a.k.a domainhood (Santini et al., 2018), refers to the domain representativeness of a corpus. For instance, a high frequency of medical terms is a sign that a corpus is a specialized medical corpus. However, a domain might have different granularities. As pointed out by Lippincott et al. (2011) ”[w]hile variation at a coarser domain level such as between newswire and biomedical text is well-studied and known to affect the portability of NLP systems, there is a need to develop an awareness of subdomain variation when considering the practical use of language processing applications”. Previous experiments showed that burstiness is a promising measure for the profiling and quantification of domain specificity (Santini et al., 2018). Burstiness is attractive for three main reasons. First, it helps identify words that are frequent in certain documents, but that are unevenly distributed in the corpus as a whole. This characterization is suitable for many specialized web corpora, where domain-specific terms are discussed in some of the documents, but not in all of them. Second, it is a measure based on word frequencies, so it requires very little pre-processing and can be applied to any language. Third, it is easy to understand and implement, since: ”Burstiness is like the mean but it ignores documents with no instances” (Church and Gale, 1995).

2. Previous Work

The importance of a quantitative evaluation of corpora has been stressed for a long time (Kilgarriff, 2001). Although many researchers have worked on the design and assessment of web corpora, no standard metrics have been agreed upon to date.

Currently, research is available on the evaluation of general-purpose web corpora. For instance, Schäfer et al. (2013) focus on the quality of texts, Ciaramita and Baroni (2006) on the representativeness of a web corpus when compared to a traditional corpus, Eckart et al. (2012) highlight the importance of standardized preprocessing steps, and Kilgarriff et al. (2014) show how to evaluate a web corpus for a specific task, namely a collocation dictionary.

Corpora can be assessed according to several criteria.

Domain, genre, style, register, medium, etc. are well-known aspects that affect corpus representativeness. Here we focus on the quality of "domain" and explore ways to profile and quantify domain-specific web corpora. Our aim is somewhat similar to SPARTAN, a technique for constructing specialized corpora from the web by systematically analysing website contents (Wong et al., 2011). However, our purpose is not to analyze the domain-specificity of individual websites as a whole, rather we focus on web pages about chronic diseases retrieved from several web sites by search engines. In recent experiments (Santini et al., 2018), we presented a case study where we explored the effectiveness of different measures - namely the Mann-Withney-Wilcoxon Test, Kendall correlation coefficient, Kullback–Leibler divergence, log-likelihood and burstiness - to assess domainhood. Our findings indicated that burstiness was the most suitable measure to single out domain-specific words. In the next sections, we apply burstiness to two medical web corpora of different size and different domain granularity.

3. Specialized Web Corpora and Domain Granularity

Since "words are not selected at random" (Kilgarriff, 2005), we assume that the content words included in a corpus represent its content and domain. The corpora that we describe below both belong to the medical domain, but they have been built with slightly different target domains and domain granularity (see Section 3.1). The target domains are represented by reference lists (see Section 3.2).

3.1 Same Domain, Different Granularities

We rely on two web corpora of Swedish texts, namely *eCare_ch_sv_01* and *eCare_uc_sv_02*. Both corpora are components of the eCare web corpus. *eCare_ch_sv_01* is about chronic diseases, while *eCare_uc_sv_02* was built with terminology automatically extracted from the E-care@home's project use cases, i.e. narratives that describe chronic diseases that affect the elderly.

eCare_ch_sv_01 was built using 155 terms listed in SNOMED CT, Swedish edition indicating chronic diseases as seeds. The 155 terms were selected from a much longer list of chronic diseases compiled by a domain expert and they represent a restricted and fine-grained domain (Santini et al., 2017). The size of this corpus is approx. 700 000 words. This corpus was used in the experiments presented in Santini et al. (2018).

eCare_uc_sv_02 was created more recently using seed terms automatically extracted from the use cases of the E-care@home project. These use cases describe the chronic ailments that affect the elderly and the recommended treatments. The size of this corpus is approx. 7 million words (6 942 193 tokens). *eCare_uc_sv_02* is, thus, about 10 times larger than *eCare_ch_sv_01* and we use it here for the first time.

Both web corpora are supposed to represent the domain of chronic diseases but with different domain granularities and different corpus sizes. We assume that the domain granularity is more fine-grained in *eCare_ch_sv_01* and coarser in *eCare_uc_sv_02* because of the way the corpora have been bootstrapped. In this study, "fine-grained domain" means a very specialized domain where the seeds to bootstrap the corpus are specialized medical terms, e.g. "artrit" (en: arthritis), while "coarse-domain" refers to a corpus that has been bootstrapped both with specialized medical terms and polysemous words that are often related with diseases, e.g. "dos" (en: dosage) or "akut" (en: acute). The domain-granularity is implicitly incorporated in the gold standards (see Section 3.2). Both web corpora were bootstrapped and downloaded with BootCat (Baroni and Bernardini, 2004), which is currently based on Bing or Google. Using regular search engines (like Google, Yahoo or Bing) and seeds to build a corpus is handy, but it also has some caveats that depend on the design or distortion of the underlying search engine (Wong et al., 2011). These caveats affect the content of web corpora since it might happen that irrelevant documents are included in the collection, especially when searching for very specialized terms. Since manual and qualitative inspections are often prohibitive, the automatic assessment of the domain specificity of a corpus bootstrapped from the web is potentially very useful.

3.2 Corpus Seeds and Gold Standards

What is the best way to represent a target domain? This question is complex and arguably the ideal solution depends on the purpose of an application. Here we take a basic approach and represent the target domains as reference lists (gold standards) that contain the term seeds used to bootstrap the corpora. It makes sense to use domain-specific terms both for bootstrapping a web corpus and for evaluating its domainhood because the terms used as seeds (source terms) should be found in non-trivial proportions to be sure that the corpus is domain-representative. Here we present two different approaches to gold standard construction. The gold standard used to profile and evaluate *eCare_ch_sv_01* is made *only* of specialized medical terms, while the gold standard automatically extracted from use cases contains also polysemous words, such as "attack" (en: attack), "extrem" (en: extreme), "fet" (en: fat). The gold standards contain tokenized term seeds, without duplicates. This means that terms like "kronisk anemi" (en: chronic anemia) and "kronisk artrit" (en: chronic arthritis), in the gold standard are represented by three entries, namely "kronisk", "anemi" and "artrit". Both these lists and the top-ranked bursty words were stemmed, stopwords and numbers were removed using the R package Quanteda, without applying any customization to the stoplist and to the stemmer.

The two web corpora are evaluated against two gold standards. More specifically, *gold_eCare_ch_sv_01* represents the target domain of *eCare_ch_sv_01* and contains 164 unigrams, while the target domain of *eCare_uc_sv_02* is represented by *gold_eCare_uc_sv_02* that contains 248 unigrams.

4. Burstiness

Burstiness indicates "how peaked a word's usage is over a particular corpus of documents" (Pierrehumbert, 2012) and helps identify words that are important in certain documents, but that are "unevenly distributed in the corpus as

a whole" (Irvine and Callison-Burch, 2017). While bursty words are feared and filtered out when assessing general-purpose corpora (Sharoff, 2017), we think that they could give a good indication of domain specificity in some kind of web corpora, like the eCare corpus.
Several burstiness formulas exist. Here we use the formula from Church and Gale (1995), including the modification proposed by Irvine and Callison-Burch (2017) (i.e. the use of relative frequencies rather than absolute frequencies), namely:

$$B_w = \frac{\sum_{d_i \in D} rf_{w_{d_i}}}{df_w} \quad (1)$$

where *rf* refers to the relative frequency of word *w* in a document, and *df* is the number of documents in which the word *w* appears. Relative frequencies are raw frequencies normalized by document length. In other words, burstiness is given by the sum of the all the relative frequencies of word *w* in the documents of the corpus divided by the number of documents containing the word. Burstiness is essentially the mean of a word in a corpus normalized by the number of documents where the word appears, and ignoring the documents where the word does not appear (Church and Gale, 1995; Katz, 1996).

Burstiness differs from measures like TF (Term Frequency) – which is simply the frequency of occurrence of a word normalized by document length – and TF*IDF where the TF is normalized by IDF (Inverse Document Frequency), which takes the log of the total number of documents in a corpus (irrespective of the presence or absence of the word *w*) divided by the number of documents containing the word *w*. If compared with more traditional profiling measures, such as log-likelihood, burstiness is a "self-contained" measure, because it does not need a reference corpus to be calculated, and the top-ranked bursty words can be easily matched against a gold standard representing the target domain.

5. Experiments

Burstiness was calculated separately for *eCare_ch_sv_01* and for *eCare_uc_sv_02*. For each corpus, we sorted the burstiness values by decreasing order and we took the top 2105 bursty words for *eCare_ch_sv_01* (Santini et al., 2018) and the top 21028 bursty words for *eCare_uc_sv_02* (since *eCare_uc_sv_02* is about 10 times larger than *eCare_ch_sv_01*) and matched them against the two gold standards that were described in Section 3.2. We used several metrics to assess the results, namely: intersection, percentage, precision@, Jaccard and Dice coefficients. For precision@ we use two cut-off points, i.e. 2105 for *eCare_ch_sv_01* and 21028 for *eCare_uc_sv_02*.

Table 1: Assessment of bursty words against gold standards

	Inter	%	Precision@	Jaccard	Dice
ch_sv_01	93	58.1%	0.0359	0.0427	0.0819
uc_sv_02	183	73.7%	0.0111	0.0086	0.0172

Results are shown in Table 1, which reports the intersection between the top-ranked scores and the gold standard (col.2), percentage (col. 3), precision@ (col. 4), Jaccard coefficient (col.5), and Dice coefficient (col. 6). The size of the intersection and the percentage give an intuitive understanding of the overlap between the top-ranked bursty words and the target domains stored in the gold standards. The intersections show a promising 58.1% for *eCare_ch_sv_01* and 73.6% for *eCare_uc_sv_02*. It is also encouraging to note that burstiness seems to be robust to corpus size variation since we observe that the number of domain-specific words identified increases with the size of the corpus rather than dropping. Apparently, the values of precision@ and those of the two coefficients do not make justice to the magnitude of the overlap since their calculation takes into account the number of unmatched items, which in our case are many because the gold standards are much shorter than the lists of top-ranked bursty words.

5.1 Discussion

Results show that burstiness and the extent to which words with a higher burstiness overlap with gold standards (i.e. reference lists comprising domain-specific vocabulary) can be used to profile and quantify the domain specificity of a (web) corpus. As stated earlier, the burstiness of a word indicates to what extent its frequency is unevenly distributed across documents within a specialized web corpus. This characterization fits very well the web corpora used in these experiments where domain-specific medical terms appear only in some documents. We find these results promising because burstiness has the potential to "discover" and bring to the surface words that are important and domain specific, but that are distributed unevenly across a corpus. Many bursty words match the gold standards. This is encouraging because burstiness seems to capture the way in which content is distributed in this kind of web corpora. In this situation, a measure like perplexity, an evaluation metric used to evaluate language models and often also to assess domain adaptation in NLP tasks, could give misleading results, because of the number of "unpredictable" bursty words.
We observe that an intersection of 93 words out of the 160 unigrams listed in *gold_eCare_ch_sv_01* (58.1%) indicates that about 8% of the 2015 top-ranked bursty words belong to the fine-grained domain of 155 SNOMED CT chronic diseases. An intersection of 183 words out to the 248 unigrams listed in *gold_eCare_uc_sv_02* (73.7%) indicates that about 1.2% of the 21028 top-ranked bursty words belong to the coarse-grained domain extracted from eCare use cases. At this stage of research we do not make any assumption about the minimum size of intersection that would account for a certain domain granularity, since we need further investigations to find a more principled approach to assess the relation between the size of the corpus, the length of the gold standards, and the cut-off points.

5.2 Open Issues

Research on the quantification of domain granularity of corpora bootstrapped from the web is still at the outset and several issues need to be further discussed and investigated.
Domain granularity: in this study, we put forwards two

working definitions, namely "fine-grained domain" means bootstrapped with specialized medical terms, and "coarse-grained domain" means bootstrapped with both specialized medical terms and more general words.
Evaluation: the quantification using the intersection and percentage is more intuitive than precision@, Jaccard and Dice coefficients. However, further experimentation is needed to establish a balanced and principled relation between the size of the corpus, the length of the gold standards, and the cut-off points.
Cut-off points: the decision about the cut-off points was based on a rule of thumb, but in the future we would rather find more theoretically-grounded threshold settings, for example, the statistical significance of the burstiness scores.
Gold standards: the design of the gold standards is exploratory rather than principled. Discussion with domain experts is ongoing.
Last but not least, in these experiments we focus on lexical items because words are easy to pre-process. However, domain specificity certainly includes other aspects, such as special syntactic constructs, stance or sublanguage variations.

6. Conclusion and Future Work

In this paper, we explored whether burstiness is a suitable measure to profile and quantify domain specificity both for small and large specialized web corpora with different domain granularities. Results show that burstiness gives a good indication of the domainhood. We find these results promising because burstiness has the potential to discover terms that are domain specific, but that are not evenly distributed in a corpus and could easily be ignored by other statistical measures.
However, some open issues need to be further investigated, such as the need for more appropriate evaluation metrics, the quest of less empirical cut-off points, and a more principled design of the gold standards.
We are currently planning several follow-up studies that include comparative experiments between burstiness, perplexity, TF, TF*IDF and topic models on several (web) corpora characterized by different word frequency distributions (e.g. poisson mixtures). In the future, we plan to use burstiness not only to assess domainhood, but also for document indexing, terminology induction and for removing irrelevant documents from a web corpus.

Acknowledgements

We thank the reviewers for their useful comments. This research was supported by E-care@home, a "SIDUS – Strong Distributed Research Environment" project, funded by the Swedish Knowledge Foundation. Project website: http://ecareathome.se/ Lists of seeds, gold standards and other material is available at: http://santini.se/eCareCorpus/home.htm

References

M. Baroni and S. Bernardini. 2004. Bootcat: Bootstrapping corpora and terms from the web. In *LREC*.

Kenneth W Church and William A Gale. 1995. Poisson mixtures. *Natural Language Engineering*, 1(2):163–190.

M. Ciaramita and M. Baroni. 2006. A figure of merit for the evaluation of web-corpus randomness. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*.

Thomas Eckart, Uwe Quasthoff, and Dirk Goldhahn. 2012. The influence of corpus quality on statistical measurements on language resources. In *LREC*, pages 2318–2321. Citeseer.

A. Irvine and C. Callison-Burch. 2017. A comprehensive analysis of bilingual lexicon induction. *Computational Linguistics*, 43(2):273–310.

S. M Katz. 1996. Distribution of content words and phrases in text and language modelling. *Natural Language Engineering*, 2(1):15–59.

Adam Kilgarriff, Pavel Rychlỳ, Milos Jakubicek, Vojtech Kovár, Vit Baisa, and Lucia Kocincová. 2014. Extrinsic corpus evaluation with a collocation dictionary task. In *LREC*, pages 545–552.

A. Kilgarriff. 2001. Comparing corpora. *International journal of corpus linguistics*, 6(1):97–133.

A. Kilgarriff. 2005. Language is never, ever, ever, random. *Corpus linguistics and linguistic theory*, 1(2):263–276.

T. Lippincott, D. Ó Séaghdha, and A. Korhonen. 2011. Exploring subdomain variation in biomedical language. *BMC bioinformatics*, 12(1):212.

J. Pierrehumbert. 2012. Burstiness of verbs and derived nouns. In *Shall We Play the Festschrift Game?*, pages 99–115. Springer.

M. Santini, A. Jönsson, M. Nyström, and M. Alireza. 2017. A web corpus for ecare: Collection, lay annotation and learning. first results. In *Proceedings of the 2nd International Workshop on Language Technologies and Applications (LTA17)*. FedCSIS.

M. Santini, W. Strandqvist, M. Nyström, M. Alirezai, and A. Jönsson. 2018. Can we quantify domainhood? exploring measures to assess domain-specificity in web corpora. In *International Conference on Database and Expert Systems Applications*, pages 207–217. Springer.

R. Schäfer, A. Barbaresi, and F. Bildhauer. 2013. The good, the bad, and the hazy: Design decisions in web corpus construction. In *8th Web as Corpus Workshop*, pages pp–7.

S. Sharoff. 2017. Know thy corpus! Exploring frequency distributions in large corpora. In Mona Diab and Aline Villavicencio, editors, *Essays in Honor of Adam Kilgarriff*. Text, Speech and Language Technology Series, Springer.

W. Wong, W. Liu, and M. Bennamoun. 2011. Constructing specialised corpora through analysing domain representativeness of websites. *Language resources and evaluation*, 45(2):209–241.

Comparing LSTM and FOFE-based Architectures for Named Entity Recognition

Marcus Klang, Pierre Nugues

Department of Computer Science
Lund University, S-221 00 Lund
Marcus.Klang@cs.lth.se, Pierre.Nugues@cs.lth.se

Abstract

LSTM architectures (Hochreiter and Schmidhuber, 1997) have become standard to recognize named entities (NER) in text (Lample et al., 2016; Chiu and Nichols, 2016). Nonetheless, Zhang et al. (2015) recently proposed an approach based on *fixed-size ordinally forgetting encoding* (FOFE) to translate variable-length contexts into fixed-length features. This encoding method can be used with feed-forward neural networks and, despite its simplicity, reach accuracy rates matching those of LTSMs in NER tasks (Xu et al., 2017). However, the figures reported in the NER articles are difficult to compare precisely as the experiments often use external resources such as gazetteers and corpora. In this paper, we describe an experimental setup, where we reimplemented the two core algorithms, to level the differences in initial conditions. This allowed us to measure more precisely the accuracy of both architectures and to report what we believe are unbiased results on English and Swedish datasets.

1. Introduction

Named entity recognition (NER) aims at identifying all the names of persons, organizations, geographic locations, as well as numeric expressions in a text. This is a relatively old task of NLP that has applications in multiples fields such as information extraction, knowledge extraction, product recommendation, and question answering. Named entity recognition is also usually the first step of named entity linking, where the mentions of named entities, once recognized, are disambiguated and linked to unique identifiers (Ji and Nothman, 2016; Ji et al., 2017).

Over the time, NER has used scores of techniques starting from hand-written rules, to decision trees, support vector machines, logistic regression, and now deep neural networks. The diversity of applications and datasets makes it difficult to compare the algorithms and systems. Researchers in the field quickly realized it and the committee of the message understanding conferences (MUC) first defined procedures for a systematic evaluation of NER performance (Grishman and Sundheim, 1996). The CoNLL 2002 and 2003 conferences (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) further developed them and provided standardized annotations, multilingual datasets, and evaluation scripts, that are still references today.

In spite of a continuous research, designing a perfect domain-independent NER is still an unmet goal. New ideas and architectures make that the state-of-the-art is improving every year. However, the figures reported in the NER articles are difficult to compare precisely as the experiments often involve external resources such as gazetteers and non-published corpora.

In this paper, we describe an experimental setup, where we reimplemented two of the best reported algorithms and where we defined identical initial conditions. This allowed us to measure more precisely the accuracy of both architectures and to report what we believe are unbiased results on English and Swedish datasets.

2. Previous Work

NER has been addressed by many techniques. Participants in the MUC conferences, such as FASTUS, used extensively gazetteers and regular expressions to extract the mentions (Appelt et al., 1993). The CoNLL conferences started to distribute annotated corpora that enabled participants to train classifiers such as logistic regression, decision trees, perceptrons, often organized as ensembles. For a review of early systems from 1991 to 2006, see Nadeau and Sekine (2007).

With the advent of deep learning, long short-term memory architectures (LSTM) (Hochreiter and Schmidhuber, 1997) have become standard to recognize named entities. Out of the 24 teams participating in the trilingual entity disambiguation and linking task (EDL) of TAC 2017, 7 used bidirectional LSTMs – with varying degrees of success (Ji et al., 2017).

Chiu and Nichols (2016) reported a score of 91.62 on the CoNLL 2003 test set with LSTM and convolutional neural networks (CNN) on character embeddings using the development set and the training set to build their model; Lample et al. (2016) used LSTM and conditional random fields (CRF) and reached 90.94 on the same test set; Ma and Hovy (2016) combined LSTM, CNN, and CRF and obtained 91.21.

Parallel to the LSTM achievements, Zhang et al. (2015) recently proposed an approach based on *fixed-size ordinally forgetting encoding* (FOFE) to translate variable-length contexts into fixed-length features. This encoding method can be used with feed-forward neural networks and, despite its simplicity, reach accuracy rates matching those of LTSMs in NER tasks (Xu et al., 2017).

All the reported performance figures are now close and may be subject to initialization conditions of random seeds. See Reimers and Gurevych (2017) for a discussion on their validity. In addition, all the experiments are carried out on the same data sets, again and again, which may, in the long run, entail some data leaks.

In this paper, we report experiments we have done with reimplementations of two of the most accurate NER taggers on English, to be sure we could reproduce the figures and that we applied to the Swedish Stockholm-Umeå corpus (SUC) (Ejerhed et al., 1992).

3. Datasets and Annotations

Annotated datasets. As datasets, we used the English corpus of CoNLL 2003, OntoNotes, and SUC, that bracket the named entities with semantic categories such as location, person, organization, etc. The corpora use either IOB v1 or v2 as annotation tagsets. We converted the annotation to IOBES, where S is for single-tag named entities, B, for begin, E, for end, I, for inside, and O for outside. For the bracketed example from CoNLL:

> Promising 10th-ranked [$_{MISC}$ American $_{MISC}$] [$_{PER}$ Chanda Rubin $_{PER}$] has pulled out of the [$_{MISC}$ U.S. Open Tennis Championships $_{MISC}$] with a wrist injury, tournament officials announced.

the annotation yields:

> Promising/O 10th-ranked/O American/S-MISC Chanda/B-PER Rubin/E-PER has/O pulled/O out/O of/O the/O U.S./B-MISC Open/I-MISC Tennis/I-MISC Championships/E-MISC with/O a/O wrist/O injury/O ,/O tournament/O officials/O announced/O ./O

The CoNLL 2003 dataset is derived from the Reuters corpus (RCV).

Word embeddings. For English, we used the pre-trained Glove 6B embeddings (Pennington et al., 2014) and the lower-cased 100 to 300 dimension variants. In addition, we trained our own cased and lowercased embeddings using the Word2vec algorithm provided by the Gensim library (Řehůřek and Sojka, 2010). For Swedish, we used Swectors (Fallgren et al., 2016) and we trained Swedish embeddings from the Swedish Culturomics Gigaword Corpus (Eide et al., 2016).

4. Systems

We implemented two systems: one based on FOFE, which is an extension to that of Klang et al. (2017) and Dib (2018) and the second one on LSTM, taking up the work of Chiu and Nichols (2016).

4.1 FOFE

The FOFE model can be seen as a weighted bag-of-words (BoW). Following the notation of Xu et al. (2017), given a vocabulary V, where each word is encoded with a one-hot encoded vector and $S = w_1, w_2, w_3, ..., w_n$, an arbitrary sequence of words, where e_n is the one-hot encoded vector of the nth word in S, the encoding of each partial sequence z_n is defined as:

$$z_n = \begin{cases} 0, & \text{if n = 0} \\ \alpha \cdot z_{n-1} + e_n, & \text{otherwise,} \end{cases} \quad (1)$$

where the α constant is a weight/forgetting factor which is picked such as $0 \leq \alpha < 1$. The result of the encoding is a vector of dimension $|V|$, whatever the size of the segment.

Features. The neural network uses both word and character-level features. The word features extend over parts of the sentence, while character features are only applied to the focus words: The candidates for a potential entity.

Word-level Features. The word-level features use bags of words to represent the focus words and FOFE to model the focus words as well as their left and right contexts. As context, we used all the surrounding words up to a maximum distance. The beginning and end of sentence are explicitly modeled with `BOS` and `EOS` tokens, which have been added to the vocabulary list.

Each word feature is used twice, both in raw text and normalized lower-case text. The FOFE features are used twice, both with and without the focus words. For the FOFE-encoded features, we used $\alpha = 0.5$. The complete list of features is then the following:

- Bag of words of the focus words;
- FOFE of the sentence: starting from the left, excluding the focus words; starting from the left, including the focus words; starting from the right, excluding the focus words; and starting from the right, including the focus words.

This means that, in total, the system input consists of 10 different feature vectors, where five are generated from the raw text, and five generated from the lowercase text.

Character-Level Features. The character-level features only model the focus words from left to right and right to left. We used two different types of character features: One that models each character and one that only models the first character of each word. We applied the FOFE encoding again as it enabled us to weight the characters and model their order. For these features, we used $\alpha = 0.8$. Higher choice of alpha for character features matches the original implementation. Our hypothesis is, using a higher alpha for the FOFE encoded character features increases its likelihood to remain salient during training.

Training. NER datasets are traditionally unbalanced with regards to the negative outside class. To produce enough positive examples to fit the model, we balanced every mini-batch, so that it contains a constant and adjustable ratio of positive and negative classes. The size of an epoch is defined by the number of mini-batches we can fill with the smallest class repeated T times.

4.2 LSTM

The LSTM model uses the sequential input directly, which does not require any preprocessing. We feed the network with the input sentences. Before training as a performance optimization, we sorted all the sentences by length and we then divided them into mini-batches. This reduces the amount of masking, and thereby wasteful computations as the majority of mini-batches will be of fixed length.

We use the same set of input features as Chiu and Nichols (2016):

- Word-level, the matching word-embedding for the input word or the unknown word embedding if the word is not in our vocabulary.
- Word-character level, all the characters per word are mapped to embeddings trained with the model. We extracted the alphabet manually and the language is specific.
- Word-case feature, per word class mapping such as lower, upper, title, digits etc.

Architecture. The word-character level features are passed through a convolution layer with a kernel of size 3 and a max-pooling layer with a window matching the maximum word length, resulting in a fixed-width character feature.

We tested LSTM cell sizes of dimension 100 and 200, our character embedding set at 30, and a maximum word length at 52. Dropout was set to 50% for recurrent LSTM connections, character feature and before the output layer. We observed that the output dropout had the greatest influence on the results.

All the word and character features are then concatenated per word and fed to a single BILSTM layer consisting internally of two independent LSTM cells which represent the forward and backward passes. The BILSTM output is the concatenation of both passes. We computed the tag scores for the BILSTM-CNN model using softmax from a single dense layer. The BILSTM-CNN-CRF model replaces the dense softmax layer with a CRF layer.

We used a negative log likelihood as loss function for the BILSTM-CNN-CRF model and categorical crossentropy for BILSTM-CNN.

5. Experimental Setup

We implemented all the models using Keras and Tensorflow as its backend. Early stopping was performed on all the models with a patience ranging from 5 to 10 depending on model; the parameters from the best epoch were selected for the resulting classifier. The word-embeddings were preinitialized without any preprocessing or normalization. In addition, we froze them during training but in a future work we may enable training. All the models used the Nadam optimizer.

Hyperparameters. We carried out a minimal hyperparameter search for BILSTM variants as usable parameters could be found in previous work. However, we could not use FOFE parameters as they produced poor results for us. We performed a smaller hyperparameter search on the CoNLL 2003 dataset to find more optimal parameters.

Evaluation. All the models produce IOBv2 annotations, IOBES is postprocessed by simple rules into correct IOBv2 tags. The annotated datasets were evaluated using conlleval from the CoNLL 2003 task, using tab delimiter instead of space, this because SUC3 has tokens with spaces in them.

SUC3 is evaluated on the 4 statistically significant classes instead of all 9: PERSON, PLACE, INST and MISC. The MISC is the combination of the remaining 5. Ontonotes 5 is evaluated on PERSON, GPE, ORG, NORP, LOC and MISC using the same principle as SUC. Following (Chiu and Nichols, 2016), we excluded the New Testaments portion from Ontonotes 5 as it lacks goldstandard annotations for NER.

For crossvalidation, we indexed all the sentences of the full dataset and we randomly split the index into 10 folds; this created 10 sets of indices. For each fold, we used one of them as test set and the rest as training set. For the training part, we used a 90/10% split to create a validation part which is used to determine when to stop training. Finally, we combined the predictions of the test part in each fold, 10 of them, into one dataset which we evaluated to produce the final score.

6. Results

BILSTM models outperform FOFE-CNN, as can be seen in Table 1. We trained FOFE-CNN models on Ontonotes 5 and SUC 3 with similar settings as the CoNLL 2003 dataset, these parameters produced subpar models which were not comparable without a new hyperparameter search.

Character features are important, as can be seen in Table 3 with more substantial improvements for lowercase embeddings. CRF improves the result for most embeddings and larger networks appear to have mixed results.

Acknowledgements

This research was supported by Vetenskapsrådet, the Swedish research council, under the *Det digitaliserade samhället* program.

References

Douglas Appelt, Jerry Hobbs, John Bear, David Israel, Megumi Kameyama, and Mabry Tyson. 1993. SRI: Description of the JV-FASTUS system used for MUC-5. In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore*, pages 221–235, San Francisco, August. Morgan Kaufmann.

Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the ACL*, 4:357–370.

Firas Dib. 2018. A multilingual named entity recognition system based on fixed ordinally-forgetting encoding. Master's thesis, Lund University, Lund.

Stian Rødven Eide, Nina Tahmasebi, and Lars Borin. 2016. The Swedish culturomics gigaword corpus: A one billion word Swedish reference dataset for NLP. In *From Digitization to Knowledge 2016: Resources and Methods for Semantic Processing of Digital Works/Texts, Proceedings of the Workshop*, volume 126, pages 8–12, Krakow.

Eva Ejerhed, Gunnel Källgren, Ola Wennstedt, and Magnus Åström. 1992. The linguistic annotation system of the Stockholm-Umeå corpus project. Technical Report 33, Department of General Linguistics, University of Umeå.

Per Fallgren, Jesper Segeblad, and Marco Kuhlmann. 2016. Towards a standard dataset of swedish word vectors. In *Sixth Swedish Language Technology Conference (SLTC), Umeå 17-18 nov 2016*.

CoNLL03 Test	Glove 6B 100d			Glove 6B 200d			Glove 6B 300d			RCV 256d		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BILSTM 100d	83.64	83.82	83.73	83.27	83.46	83.37	82.85	82.81	82.83	85.51	87.66	86.57
BILSTM-CNN 100d	86.87	90.00	88.41	85.39	88.77	87.05	84.75	88.74	86.70	86.25	89.09	87.65
BILSTM-CNN-CRF 100d	89.25	90.10	89.67	88.63	89.20	88.92	88.61	88.56	88.59	89.25	89.41	**89.33**
BILSTM-CNN-CRF 200d	89.76	90.44	**90.10**	89.08	89.82	**89.45**	88.80	88.83	**88.81**	88.92	88.92	89.07
FOFE-CNN	79.87	84.17	81.97	82.26	83.11	82.68	83.64	83.16	83.40	87.91	87.80	87.86

Table 1: CoNLL03 Test results

Ontonotes5 Test	Glove 6B 100d			Glove 6B 200d			Glove 6B 300d		
	P	R	F1	P	R	F1	P	R	F1
BILSTM 100d	81.32	82.46	81.89	82.65	80.03	81.32	81.49	80.09	80.79
BILSTM-CNN 100d	84.20	86.91	85.53	82.59	85.84	84.19	82.92	86.40	**84.62**
BILSTM-CNN-CRF 100d	82.70	86.50	84.56	83.91	84.10	84.00	83.13	84.84	83.97
BILSTM-CNN-CRF 200d	84.62	86.79	**85.69**	86.43	86.00	**86.22**	81.86	84.50	83.16

Table 2: Ontonotes 5 Test results.

SUC3 Test	Swectors 300d			Gigawords 256d			Gigawords Lcase 256d		
	P	R	F1	P	R	F1	P	R	F1
BILSTM 100d	81.82	64.07	71.87	81.91	77.98	79.90	79.14	75.99	77.53
BILSTM-CNN 100d	79.72	74.83	77.20	82.33	81.79	82.06	82.23	80.46	81.34
BILSTM-CNN-CRF 100d	82.55	78.31	80.37	86.13	83.28	**84.68**	82.53	82.12	**82.32**
BILSTM-CNN-CRF 200d	85.09	77.48	**81.11**	81.63	79.47	80.54	82.15	80.79	81.47
SUC3 10-fold crossvalidation									
BILSTM-CNN-CRF 100d	82.07	80.22	81.14	85.22	83.62	84.41	84.31	84.32	84.31

Table 3: SUC 3 Test results

Ralph Grishman and Beth Sundheim. 1996. Message understanding conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, pages 466–471.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Heng Ji and Joel Nothman. 2016. Overview of TAC-KBP2016 Tri-lingual EDL and Its Impact on End-to-End KBP. In *Proceedings of the Ninth Text Analysis Conference (TAC 2016)*, Gaithersburg.

Heng Ji, Xiaoman Pan, Boliang Zhang, Joel Nothman, James Mayfield, Paul McNamee, and Cash Costello. 2017. Overview of TAC-KBP2017 13 Languages Entity Discovery and Linking. In *Proceedings of the Tenth Text Analysis Conference (TAC 2017)*, Gaithersburg.

Marcus Klang, Firas Dib, and Pierre Nugues. 2017. Overview of the ugglan entity discovery and linking system. In *Proceedings of the Tenth Text Analysis Conference (TAC 2017)*, Gaithersburg, Maryland, November.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *CoRR*, abs/1603.01360.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the ACL (Volume 1)*, pages 1064–1074.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Linguisticæ Investigationes*, 30(1):3–26.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on EMNLP*, pages 1532–1543, Doha.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta.

Nils Reimers and Iryna Gurevych. 2017. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on EMNLP*, pages 338–348, Copenhagen.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147, Edmonton.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158, Taipei.

Mingbin Xu, Hui Jiang, and Sedtawut Watcharawittayakul. 2017. A local detection approach for named entity recognition and mention detection. In *Proceedings of the 55th Annual Meeting of the ACL (Volume 1)*, pages 1237–1247, Vancouver.

ShiLiang Zhang, Hui Jiang, MingBin Xu, JunFeng Hou, and LiRong Dai. 2015. The fixed-size ordinally-forgetting encoding method for neural network language models. In *Proceedings of the 53rd Annual Meeting of the ACL and the 7th IJCNLP (Volume 2)*, pages 495–500.

A Component based Approach to Measuring Text Complexity

Simon Jönsson, Evelina Rennes, Johan Falkenjack, Arne Jönsson

Linköping University, Linköping, Sweden
simjo241@student.liu.se, evelina.rennes|johan.falkenjack|arne.jonsson@liu.se

Abstract

We present results from assessing text complexity based on a factorisation of text property measures into components. The components are evaluated by investigating their ability to classify texts belonging to different genres. Our results show that the text complexity components correctly classify texts belonging to specific genres, given that the genres adhere to certain textual conventions. We also propose a radar chart visualisation to communicate component based text complexity.

1. Introduction

Recent years' development of speed and accuracy of text analysis tools has made new text features available for readability assessment. For instance, phrase structure parsing has been used to find the average number of sub-clauses, verb phrases, noun phrases and average tree depth (Schwarm and Ostendorf, 2005). For Swedish, Heimann Mühlenbock (2013), Falkenjack and Jönsson (2014), and Falkenjack et al. (2013) have addressed such data driven text complexity assessment.

2. Text complexity measures

For the study presented in this paper we use the publicly available toolkit TECST (Falkenjack et al., 2017) and the text complexity analysis module SCREAM (Heimann Mühlenbock, 2013; Falkenjack et al., 2013). As of today, SCREAM calculates 119 features of text complexity that roughly can be divided into the following categories:

Shallow features are features that can be extracted after tokenisation by simply counting words and characters. Shallow features include mean word length and mean sentence length.

Lexical features are based on categorical word frequencies extracted after lemmatisation and calculated using the basic Swedish vocabulary SweVoc (Heimann Mühlenbock, 2013). They are further divided into groups such as everyday use and communication.

Morpho-syntactic features concern a morphology based analysis of the text. The analysis relies on previously part-of-speech annotated text. Measures include a number of part-of-speech tags and ratio of content words.

Syntactic features are features that can be estimated after syntactic parsing of the text. Features include a number of dependency distance measures.

Text quality metrics include measures that traditionally are used to measure readability.

Several studies have explored how text complexity measures can be combined and clustered in different ways to be more comprehensive and easier to understand, c.f. (Falkenjack et al., 2016). One way of conducting clustering is through factor analysis, allowing large amounts of variables to be combined into fewer clusters or factors. Biber (1988) conducted such analyses in order to find the factors that distinguish spoken language from written language. Through a principal factor analysis, 67 features were reduced to 7 factors.

Our study is inspired by Biber's three step analysis. The first step is to decide on a method for analysis. The method used by Biber (1988) is Principal Factor Analysis (PFA, also known as common factor analysis). Another method of factor analysis is Principal Component Analysis (PCA). A fundamental difference between PFA and PCA is that PFA does not account for all the variance, only the variance that is shared between variables (Biber, 1988). Henry (1979) and Lee et al. (2012) are two examples of studies that used PCA in terms of combining linguistic features into fewer components.

The second step is to decide on how many factors to extract. This can be done by analysing a screen plot and determine where additional factors do not contribute to the overall analysis (Biber, 1988). It is also possible to analyse a table to see how much variance each factor explains and how much the factors explain together. A third way of determining the number of factors to be extracted is through parallel analysis (O'Connor, 2000). The analysis is a way to test how many eigenvalues that are statistically significant.

The third step is to choose what type of rotation that should be used. Biber (1988) chooses an oblique structure, Promax, which allows for more correlations, even minor ones, among the factors.

3. Corpus

The text material used in our studies comprises texts from the SUC corpus (Ejerhed et al., 2006). In the experiments we want to investigate the ability to distinguish different text domains, or genres, using text complexity measures factorised into components as suggested by Biber (1988). There is a theoretical distinction between the concepts of genre and domain. Here domain refers to the shared general topic of a group of texts. For instance, "Fashion", "Leisure", "Business", "Sport", "Medicine" or "Education" are examples of broad domains. Genre is a more abstract concept. It characterises text varieties on the basis of conventionalised textual patterns. For instance, an *academic*

paper obeys to textual conventions that differ from the textual conventions of a *tweet*; a *letter* complies to conventions that are different from the conventions of an *interview*. *Academic papers*, *tweets*, *letters*, *interviews* are examples of genres. For more details see (Falkenjack et al., 2016). If we apply this distinction to the nine top genres included in the SUC, we end up with six "proper" genres, see Table 1.

Table 1: The six proper SUC genres used in our study

Genre	Size
Press Reportage (**A**)	269
Press Editorial (**B**)	70
Press Review (**C**)	127
Biographies/Essays (**G**)	27
Learning/Scientific Writing (**J**)	86
Imaginative Prose (**K**)	130

4. Procedure

Similar to Biber (1988), a factor analysis was conducted in order to group linguistic features. The method used here was a Principal component analysis (PCA). Features which either did not have any values or were already represented by other features by having one-to-one correlations were excluded from the feature set.

Through a parallel analysis, the number of clusters to extract from the PCA was elicited (O'Connor, 2000). The method compares raw data, principal component eigenvalues that correspond to the actual data, with random data eigenvalues. If the first value, raw data, is larger than the 95th percentile, it is considered a significant eigenvalue and is included. The extracted number of significant eigenvalues is the number of components extracted through the PCA.

With a Promax, oblique structure, the PCA was done on the set of data containing the remaining linguistic features, each with a total of 1040 data points, where each data point represents results from analyses as described above.

Using the obtained components we investigate their ability to classify the SUC genres. We are using a 18×15 *softmax* neural network with linear activation function. Since SUC has the issue of uneven amount of genre representatives we sample the data as a tensor, Batches $\times$ Samples $\times$ Components (where a batch is a 10×6 matrix of sampled measures of SUC texts), to attempt solving this issue. Genre **G** has fewer data points than the rest of the genres, giving a limited training sample. Classifying a genre that is underrepresented gives a vague model and therefore genre **G** is excluded.

Two of the components obtained can be seen in Table 2. Components were obtained by quantitatively analysing correlation between features and removing features such that we obtain maximal classification. The correlation cut-off was $|0.8|$ where we found local optimum of classification rate 84.0%.

5. Results

From the parallel analysis, a total of 28 eigenvalues were elicited that were used as number of components to be extracted. A total of 93 features remained in the data set after removing 19 features, features with either a prediction of 0, no result at all, already subsumed by other features with a one to one correlation, or not having a predictability higher than 0.65 (.503 - .646).

Table 2: Example of extracted components

Comp.	Feature	Weight	Explanation
1	pos_PN	.816	Pronouns
	pos_NN	-.808	Nouns
	nrValue	-.807	Nominal ratio
	avgNoSyllables	-.730	Average number of syllables
	dep_PA	-.729	Complement of preposition
	dep_ET	-.714	Other nominal post-modifier
	dep_MS	.612	Macrosyntagm
	ratioSweVocC	.607	SweVoc lemmas fundamental for communication
	dep_IO	.573	Indirect object
	pos_AB	.572	Adverb
	dep_SS	.525	Other subject
	dep_DT	-.524	Determiner
	avgPrepComp	-.522	Average number of prepositional complements per sentence in the document
	pos_PS	.487	Possessive pronoun
	dep_NA	.473	Negation adverbial
	dep_MA	.446	Attitude adverbial
	dep_I	.425	Question mark
	pos_RG	-.407	Cardinal number
	dep_AA	.400	Other adverbial
	dep_F	.388	Coordination at main clause level
	dep_PL	.382	Verb particle
	dep_OO	.365	Direct object
	pos_HA	.322	WH-adverb
	dep_AT	-.302	Nominal (adjectival) pre-modifier
	ratioSweVocTotal	.301	Unique, per lemma, SweVoc words in the sentence.
2	pos_PM	-.858	Proper noun
	dep_HD	-.788	Head
	lexicalDensity	.710	Lexical density
	ratioSweVocTotal	.706	Unique, per lemma, SweVoc words in the sentence.
	ratioSweVocH	.573	SweVoc other highly frequent lemmas (category H)
	ratioSweVocC	.544	SweVoc lemmas fundamental for communication
	dep_SS	.429	Other subject
	dep_AN	-.393	Apposition
	ratioSweVocD	.356	SweVoc lemmas for everyday use (category D)
	ratioVerbalRoots	.347	The ratio of sentences with a verbal root
	pos_NN	.332	Noun

An analysis using the Kaiser-Meyer-Olkin Measure of Sampling Adequacy (.595) and Bartlett's Test of Sphericity shows the validity of PCA to interpret the data set ($p < .05$)

The variables chosen for each component had a magnitude over 0.3 and under -0.3. The total variance explained by the 28 components is 60.5%, of which the first component explains 8% on its own.

The results from classification using the neural network is presented in Table 3.

Table 3: F1-Scores for the components

Genre	F1
Press Reportage (**A**)	0.814
Press Editorial (**B**)	0.793
Press Review (**C**)	0.831
Learning/Scientific Writing (**J**)	0.826
Imaginative Prose (**K**)	0.9324

We note that the F1-scores of respective genre are fairly

consistent, except genre **B**, which has a significantly lower score and genre **K** which has a significantly higher score. The former might be due to the properties of genres who has a Press origin being similar in some textual sense. Whereas Imaginative Prose, **K**, might differ from the rest of the genres in a text complexity sense, which makes it easier for the classifier to distinguish the genre. Analogously the classifier might have difficulties distinguishing Press related data points, to some extent.

Table 4: Confusion matrix for the components. Each genre has been classified 150 times.

	A	B	C	J	K
A	120	6	9	8	7
B	11	111	8	15	5
C	8	4	125	9	4
J	4	8	7	128	3
K	2	1	2	0	145

To further analyse the classification results, Table 4 presents the resulting confusion matrix. From Table 4 we note that genres **A, B, C, J** have many False Positives (FP) and many False Negatives (FN), whereas genre **K** only have strong FN, which means that the other genres are misclassified as genre **K** but **K** seldomly is misclassified as any other genre, this implies that **K** is more separated from other genres in our feature space. Also one can deduce that the other genres have more interlacement in our feature space.

6. Visualising text complexity

Each of the components derived from the factor analysis comprises several individual text complexity features that depict different aspects about the analysed texts, as can be seen in Table 2. The components can not easily be labelled in a meaningful way. Instead we propose to visualise them in a radar diagram, c.f. Branco et al. (2014), Figure 1.

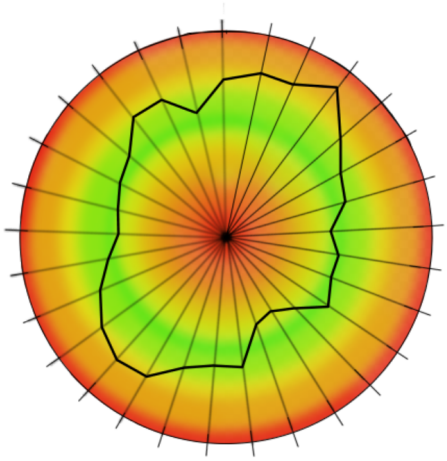


Figure 1: Visualisation of components.

The pattern in the radar chart, resulting from a text analysis, communicates something about a text's complexity, the inner line in the radar chart in Figure 1. Different texts provide different patterns and it may be possible to use such patterns to characterise a text's complexity and also compare its complexity with other texts' complexity.

The components are, thus, visualised in an intuitive way, where the pattern communicates text complexity. However, the components should also have justified names and definitions. A remaining issue is the domain-specific terminology concerning text complexity, as the meaning of the components has to be communicated along with the assessment. A huge endeavour as the components comprise features that reflect different, and sometimes opposing, qualities of a text.

The components should also be sorted in a way in which related components are closer to one another. Making use of the interactivity of a digital tool, the visualisation could be revised even further. By combining the extracted components into overall categories that may be presented at first, revealing the most important components from each category by selecting the corresponding section in a radar chart, the radar chart may become more comprehensible. This final visualisation with the components therein needs to be evaluated to properly see if it is more intuitive and if the components give users an understanding of a text's complexity.

7. Conclusions

We have shown that a component based text complexity analysis can be used to classify texts in genres. Assuming that genres have different text properties the components, thus, also say something about texts' complexity. The results are based on measuring complexity of Swedish, but very few measures are specific for Swedish. Further research should study how to define genres such that the text complexity feature space is more separated, thus leading to "stronger" genres, i.e. more distinguished genres - in a textual complexity sense.

We have also suggested that component based text complexity measures can be visualised in a radar diagram. Further research on visualisation includes conducting studies on users' understanding of text complexity using radar charts and on finding meaningful ways to reorganise the components.

Acknowledgements

This research is financed by Vinnova and RISE SICS East. We are indebted to Jakob Säll and Simon Cavedoni for initial research on PCA for visualisation.

References

Douglas Biber. 1988. *Variation across speech and writing*. Cambridge: Cambridge Univ. Press, 1988.

António Branco, João Rodrigues, Francisco Costa, Joao Silva, and Rui Vaz. 2014. Rolling out text categorization for language learning assessment supported by language technology. In *International Conference on Computational Processing of the Portuguese Language*, pages 256–261. Springer.

Eva Ejerhed, Gunnel Källgren, and Benny Brodda. 2006. Stockholm Umeå Corpus version 2.0.

Johan Falkenjack and Arne Jönsson. 2014. Classifying easy-to-read texts without parsing. In *The 3rd Workshop*

on Predicting and Improving Text Readability for Target Reader Populations (PITR 2014), Göteborg, Sweden.

Johan Falkenjack, Katarina Heimann Mühlenbock, and Arne Jönsson. 2013. Features indicating readability in Swedish text. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NoDaLiDa-2013), Oslo, Norway*, number 085 in NEALT Proceedings Series 16, pages 27–40. Linköping University Electronic Press.

Johan Falkenjack, Marina Santini, and Arne Jönsson. 2016. An Exploratory Study on Genre Classification using Readability Features. In *Proceedings of the The Sixth Swedish Language Technology Conference (SLTC 2016), Umeå, Sweden*.

Johan Falkenjack, Evelina Rennes, Daniel Fahlborg, Vida Johansson, and Arne Jönsson. 2017. Services for text simplification and analysis. In *Proceedings of the 21st Nordic Conference on Computational Linguistics, Gothemburg, Sweden*.

Katarina Heimann Mühlenbock. 2013. *I see what you mean. Assessing readability for specific target groups*. Dissertation, Språkbanken, Dept of Swedish, University of Gothenburg.

G. Henry. 1979. The relation between linguistic factors identified by a principal components analysis of written style and reading comprehension as measured by cloze tests. *Journal of Research in Reading*, 2(2):120–128.

Yi-Shian Lee, Hou-Chiang Tseng, Ju-Ling Chen, Chun-Yi Peng, Tao-Hsing Chang, and Yao-Ting Sung. 2012. Constructing a novel chinese readability classification model using principal component analysis and genetic programming. In *Proceedings of the 12th IEEE International Conference on Advanced Learning Technologies, ICALT 2012*, pages 164–166, 07.

Brian P O'Connor. 2000. SPSS and BAS programs for determining the number of components using parallel analysis and velicer's MAP test. *Behavior Research Methods, Instruments, & Computers*, 32(3):396–402.

Sarah E. Schwarm and Mari Ostendorf. 2005. Reading level assessment using support vector machines and statistical language models. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*.

An Aligned Resource of Swedish Complex-Simple Sentence Pairs

Evelina Rennes

Department of Computer and Information Science
Linköping University
evelina.rennes@liu.se

Abstract

We present a method for aligning comparable corpora of simple-complex articles at the sentence level. Three methods were tested; Average Alignment (AA), Maximum Alignment (MA), and Hungarian Alignment (HA). For evaluating the algorithms, and finding the optimal combination of parameters, a dataset of manually annotated sentences was constructed. The algorithms were evaluated against the manually annotated dataset, and the best-performing algorithm proved to be the MA algorithm, which resulted in corpus comprising 59,513 aligned sentence pairs, of which 17,653 were unique sentences.

1. Introduction

Data-driven methods have gained ground within the NLP community, and the field of Automatic Text Simplification (ATS) is no exception. Such techniques often require parallel corpora, but especially for less-resourced languages, the availability of parallel or comparable corpora is limited. For English, datasets have been constructed and used for ATS purposes, such as the Parallel Wikipedia Simplification (PWKP) corpus (Zhu et al., 2010) and the Newsela Corpus (Xu et al., 2015). Rennes and Jönsson (2016) constructed a corpus of the easy-to-read and standard parts of the websites of Swedish municipalities and public authorities. However, the material was not aligned, not at the document level nor at the sentence level, and in this paper, we aim to find a method for effectively aligning this corpus, in order to achieve a parallel resource that can be used for the automatic induction of ATS transformation patterns, or as training data for machine learning approaches to ATS.

2. Sentence Alignment

We aimed to align the corpus of Swedish public authorities and municipalities websites (Rennes and Jönsson, 2016) at the sentence level. Previous work on alignment often used Wikipedia and Simple English Wikipedia as resource to be aligned (Zhu et al., 2010; Coster and Kauchak, 2011; Hwang et al., 2015; Kajiwara and Komachi, 2016). This is suitable since the articles of Simple English Wikipedia generally have a corresponding standard Wikipedia article, meaning that they can easily be divided into article pairs on the same topic. Since the websites of Swedish public authorities and municipalities are structured in various ways, there was no simple way of collecting documents pairs. A simple title match did not work sufficiently well, due to the fact that a website may have one simple page that corresponds to several pages in standard Swedish, or the other way around. For this reason, we used a simple TF-IDF approach in order to align documents of a certain domain with similar content. After collecting similar documents, these were used for alignment at the sentence level.

The methods proposed by (Song and Roth, 2015); Average Alignment (AA), Maximum Alignment (MA), and Hungarian Alignment (HA) were used, combining word embeddings to create a sentence similarity score. The methods have previously been used to align monolingual material in order to create a resource for text simplification for English (Kajiwara and Komachi, 2016).

The Average Alignment (AA) algorithm averages the pairwise similarities of all words of a pair of sentences.

$$SIM_{avg}(x,y) = \frac{1}{|x||y|}\sum_{i=0}^{|x|}\sum_{j=0}^{|y|}\phi(x_i,y_j) \tag{1}$$

where ϕ represent the cosine similarity and $|x|$ and $|y|$ is the number of words in sentence x and y.

The Maximum Alignment (MA) algorithm considers the most similar word pair in the sentences, resulting in an asymmetric similarity score, that is then made symmetric by computing the pairwise word similarities in reversed order, and averaging the resulting score.

$$\begin{aligned} SIM_{max}(x,y) &= \frac{1}{2}(SIM_a(x,y) + SIM_a(y,x)), \\ SIM_a(x,y) &= \frac{1}{|x|}\sum_{i=0}^{|x|}\max_j \phi(x_i,y_j) \end{aligned} \tag{2}$$

where ϕ represent the cosine similarity and $|x|$ is the number of words in sentence x.

The Hungarian Alignment (HA) algorithm determines the assignment that maximises the sum with respect to all words in sentences x and y,

$$SIM_{hun}(x,y) = \frac{1}{\min(|x|,|y|)}\sum_{i=0}^{|x|}\phi(x_i,h(x,y,i)) \tag{3}$$

where h is a function to find a word y_i that maximises the sum, according to the Hungarian algorithm (Kuhn, 1955). This algorithm is restricted to one-to-one word pair mappings.

For all algorithms, we could alter word similarity thresholds (deciding when a word pair is regarded to be similar enough) and sentence similarity thresholds (judging when a sentence pair should be aligned).

The word embeddings used for the alignment were *Swectors* (Fallgren et al., 2016), a set of Swedish word vectors trained on news texts.

2.1 OOV handling

Working with word embeddings, it is inevitable that words not appearing in the vocabulary are encountered. These unknown words are called Out-Of-Vocabulary (OOV) words, and several solutions to handle these have been proposed. One common solution is to generate a word vector to be used instead of the unseen word, either by using the same vector for all OOV words, or to create a random vector every time such a word is encountered. The previous approach (Kajiwara and Komachi, 2016) using the same methods handled OOV words by simply ignoring the them, only using known words for the similarity calculation. This could be done since the word embeddings were trained on a large scale corpus, which means that the number of OOV words was likely to be small. Since our set of word embeddings was smaller, we handled this issue by generating new vectors whenever an unseen word was encountered. Instead of using a random vector, we used Mimick (Pinter et al., 2017), where a recurrent neural network is trained to mimick word embeddings based on a word's spelling. The intuition here is that the generated word vector might be closer to other words that share certain spelling patterns, than to a completely randomised vector.

The Mimick RNN was trained on Swectors using the default settings: one LSTM layer with 50 hidden units, 60 training epochs (no dropout), character embedding dimension = 20, nonlinearity function $g = tanh^2$.

3. Manually Annotated Dataset

For the evaluation of the algorithms, a manually annotated gold standard data set was constructed; a subset of article pairs (based on a simple title match) from the web corpus constructed in Rennes and Jönsson (2016). All combinations of simple/complex sentences of each article pair were presented to the annotators (two payed undergraduate students and one graduate student), and each sentence pair was rated on a scale from 0 to 3. Each number on the scale represented a category described in Hwang et al. (2015):

(0) Bad	The sentences discuss unrelated concepts
(1) Partial	The sentences discuss unrelated concepts, but share a short related phrase that does not match considerably
(2) Good Partial	A sentence completely covers the other sentence, but contains an additional clause or phrase that has information which is not contained within the other sentence
(3) Good	The semantics of the simple and standard sentence completely match, possibly with small omissions (e.g., pronouns, dates, or numbers)

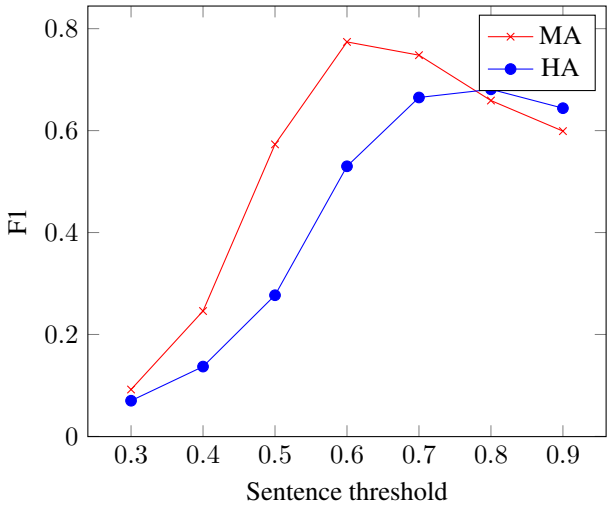


Figure 1: Distribution of F1 scores with word threshold of 0.49

10,915 sentence pairs were annotated, of which 90 sentences were annotated as *Good*, 85 sentences were annotated as *Good Partial*, 95 sentences were annotated as *Partial*, and 10,645 sentences were annotated as *Bad*.

4. Evaluation

The performance of the algorithms for automatic alignment was evaluated by comparing to the manually annotated dataset. All sentences of the manually annotated dataset scored with the label *Good* or *Good Partial* were considered correct alignments. The performance of the algorithms was measured with precision and recall, and a F1 score was calculated. This process was iterated with different values for the word similarity threshold and sentence similarity threshold, and if a sentence pair received a score above the sentence similarity threshold, it was aligned by the algorithms. For all conditions, the AA algorithm produced a low amount of alignments (< 10), and will for this reason be disregarded in this evaluation. The purpose of the evaluation was to find the optimal combination of parameters, before running the best-performing algorithm on the whole web corpus.

The combination of parameters that maximised the F1 score was the MA algorithm, with Mimick OOV handling, word alignment threshold 0.49, and sentence alignment threshold 0.6. This resulted in a F1 score of 0.774 (precision = 0.818, recall = 0.734). Of the top F1 scores, this condition also resulted in the highest amount of aligned sentences (159). The variation of F1 scores for MA and HA over different sentence alignment thresholds at the word alignment threshold of 0.49 is presented in Figure 1.

5. Final Corpus

Running the MA algorithm on the full collection of documents with the parameters that maximised F1, resulted in a total of 59,513 aligned sentence pairs. There were many duplicates among the aligned sentences, and when only considering unique sentence pairs, the number of sentence pairs was 17,653.

6. Concluding remarks

We have presented a way of aligning Swedish complex-simple sentence pairs, by applying different algorithms to calculate sentence similarity scores based on combinations of word embeddings. The best-performing algorithm for Swedish proved to be the Maximum Alignment (MA) algorithm, using Mimick word vectors for OOV handling, reaching a F1 score of 0.774.

Partial matches, or *n*-to-*m* matches, were considered to some extent since the alignment algorithm found partial sentences if they got a sentence similarity measure reaching over the sentence threshold. However, we did not investigate this issue in particular for this first evaluation, and this should be further studied.

Running the MA algorithm on the full collection of documents resulted in a total of 59,513 aligned sentence pairs. It would be valuable to evaluate this dataset further in order to investigate whether or not the sentence pairs are paraphrases. It would also be interesting to vary the parameters of the Mimick word vector generation, in order to see whether we can reach even better results. Another interesting issue concerns the hypothesis on which this method is based: do the different sentences of a sentence pair in fact differ in complexity? This will be further investigated with regards to complexity measures, and by querying human judges.

Acknowledgements

This research is financed by Vinnova and RISE SICS East.

References

William Coster and David Kauchak. 2011. Simple english wikipedia: a new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 665–669. Association for Computational Linguistics.

Per Fallgren, Jesper Segeblad, and Marco Kuhlmann. 2016. Towards a standard dataset of swedish word vectors. In *Proceedings of the Sixth Swedish Language Technology Conference (SLTC), Umeå, Sweden*.

William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. 2015. Aligning sentences from standard wikipedia to simple wikipedia. In *HLT-NAACL*, pages 211–217.

Tomoyuki Kajiwara and Mamoru Komachi. 2016. Building a monolingual parallel corpus for text simplification using sentence similarity based on alignment between word embeddings. In *Proceedings of COLING, Osaka, Japan*, pages 1147–1158.

Harold W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

Yuval Pinter, Robert Guthrie, and Jacob Eisenstein. 2017. Mimicking word embeddings using subword rnns. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 102–112.

Evelina Rennes and Arne Jönsson. 2016. Towards a corpus of easy to read authority web texts. In *Proceedings of the Sixth Swedish Language Technology Conference (SLTC-16), Umeå, Sweden*.

Yangqiu Song and Dan Roth. 2015. Unsupervised sparse vector densification for short text similarity. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1275–1280.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd international conference on computational linguistics*, pages 1353–1361. Association for Computational Linguistics.

Universal Dependency Parsing at Uppsala University

Joakim Nivre **Miryam de Lhoneux** **Aaron Smith** **Sara Stymne**

Uppsala University
Department of Linguistics and Philology

Abstract

We present ongoing work in the Universal Dependency Parsing project at Uppsala University, which investigates parsing models for typologically diverse languages within the framework of Universal Dependencies. After presenting the basic architecture of UUParser, we review two recent studies concerned with the impact of pre-trained word embeddings, character models and part-of-speech tags, on the one hand, and with pooling training data from multiple languages and treebanks, on the other. We conclude with a report from the recently concluded CoNLL shared task on multilingual universal dependency parsing.

1. Introduction

The accuracy of syntactic parsers has increased gradually over the last decades, and it is not uncommon today to see evaluation scores around 95% for standard benchmarks such as the Penn Treebank of English. However, these results are not only limited to certain text types, such as carefully edited newspaper text, but typically only hold for a small set of languages, with special structural characteristics and supported by large-scale resources.

The goal of the Universal Dependency Parsing project at Uppsala University is to study parsing models for typologically diverse languages in order to find out what techniques work well across languages and what aspects require language-specific adaptation. The central hypothesis is that parsing models need a better abstraction over concrete realization patterns, such as morphological inflection, function words and word order, in a way that is informed by linguistic typology. To test this hypothesis, we extend and analyze existing dependency-based parsing models to better cope with typological diversity and adapt them to the representations of Universal Dependencies (UD), a system for cross-linguistically consistent grammatical analysis so far applied to over 70 languages (Nivre et al., 2016).

In this paper, we report on two recent studies within the project. The first is concerned with the impact and interaction of three techniques for word representation: *word embeddings*, *character models*, and *part-of-speech tags* (Smith et al., 2018b). The second is concerned with *treebank embeddings*, a technique for combining heterogeneous data sets for parser training, both within and across languages (Stymne et al., 2018; Smith et al., 2018a). We begin with a short description of UUParser, the parser implementation used in all experiments, and end with a short report on our participation in the recently concluded CoNLL 2018 shared task on multilingual universal dependency parsing (Smith et al., 2018a; Zeman et al., 2018).

2. UUParser

UUParser is a greedy transition-based parser (Nivre, 2008) based on the framework of Kiperwasser and Goldberg (2016), where BiLSTMs are used to learn representations of tokens in context, and are trained together with a multilayer perceptron (MLP) that predicts transitions and arc labels based on a set of BiLSTM vectors. For each input token w_i, the representation x_i fed into the BiLSTM layer minimally includes a word embedding $e(w_i)$ but may be extended with additional information, and the experiments described in the following sections in fact only manipulate the input representations, keeping the rest of the parsing architecture constant.

The transition system used by UUParser is a variant of the arc-hybrid transition system (Kuhlmann et al., 2011), extended with a SWAP transition to allow the construction of non-projective dependency trees (Nivre, 2009) and a static-dynamic oracle to allow the parser to learn from non-optimal configurations at training time in order to recover better from mistakes at test time (de Lhoneux et al., 2017b). The input to the MLP that predicts transitions and arc labels consists of the BiLSTM vectors of the top three tokens on the stack and their rightmost and leftmost dependents, plus the first token in the buffer and its leftmost dependent, which is equivalent to the extended feature set of Kiperwasser and Goldberg (2016). For a more detailed description of UUParser, see de Lhoneux et al. (2017a) and Smith et al. (2018a).

3. Word Embeddings, Character Models, and Part-of-Speech Tags

Representing input tokens by embeddings – dense continuous vectors – instead of sparse discrete representations is standard in neural network approaches to syntactic parsing and other NLP tasks, and a number of different enhancements of these representations have been proposed, including pre-training word embeddings on large unlabeled corpora (Chen and Manning, 2014; Kiperwasser and Goldberg, 2016), adding embeddings of part-of-speech tags (Chen and Manning, 2014), and adding character-based representations (Ballesteros et al., 2015). All of these techniques have been shown to improve parsing accuracy, but there have been few systematic studies of exactly why they improve accuracy and to what extent the benefits of different techniques are complementary or redundant in relation to each other.

In Smith et al. (2018b), we study the interaction of pre-trained word embeddings, character models and (embedded) part-of-speech tags in the context of the UUParser by

systematically varying the input representation x_i of a word w_i. In the simplest model, x_i is equal to a randomly initialized word embedding $e^r(w_i)$:

$$x_i = e^r(w_i) \quad (1)$$

In the most complex model, the randomly initialized embedding is replaced by a pre-trained embedding $e^t(w_i)$, which is concatenated with a character-based vector BiLSTM$(ch_{1:m})$, obtained by running a BiLSTM over the characters $ch_{1:m}$ of w_i, and an embedding $e(p_i)$ of the word's universal part-of-speech tag (Nivre et al., 2016):

$$x_i = e^t(w_i) \circ \text{BiLSTM}(ch_{1:m}) \circ e(p_i) \quad (2)$$

In addition to the simplest and most complex models, we test all combinations of one and two enhancements in experiments on nine treebanks from Universal Dependencies (Nivre et al., 2016) (v2.0): Ancient Greek (PROIEL), Arabic (PADT), Chinese (GSD), English (EWT), Finnish (TDT), Hebrew (HTB), Korean (GSD), Russian (GSD) and Swedish (Talbanken). Our main findings can be summarized as follows:

- For all techniques, improvements are largest for low-frequency and open-class words and for morphologically rich languages.
- These improvements are largely redundant when the techniques are used together.
- Character-based models are the most effective technique for low-frequency words.
- Part-of-speech tags are potentially effective for high-frequency function words, but current state-of-the-art taggers are not accurate enough to fully exploit this.
- Large character embeddings are helpful in morphologically rich languages, regardless of character set size.

4. Multi-Treebank Models

When training parsers, we sometimes want to combine (annotated) data from multiple, heterogeneous sources. In a monolingual setting, we may have access to treebanks containing different text genres or annotated in slightly different styles. In a multilingual setting, we may want to combine training data from multiple languages in order to improve parsing accuracy for low-resource languages. Simply concatenating the training sets, however, is unlikely to give optimal performance and often results in degraded performance. In recent work, we have explored the use of *treebank embeddings* for parser training with heterogeneous treebanks, generalizing the *language embeddings* of Ammar et al. (2016) to apply not only in multilingual but also in monolingual settings.

The basic idea is to add a treebank embedding $e(tb_i)$ to the input vector x_i associated with each token w_i:

$$x_i = e^t(w_i) \circ \text{BiLSTM}(ch_{1:m}) \circ e(p_i) \circ e(tb_i) \quad (3)$$

At training time, the treebank embeddings allow the parser to learn from multiple treebanks while remaining sensitive to the idiosyncracies of each one. At parsing time, we can select the treebank embedding that is most suitable for the input text, whether in a particular language or belonging to a particular text genre.

In Stymne et al. (2018), we show that treebank embeddings provide an effective way to combine multiple heterogeneous treebanks in the monolingual setting. In experiments with 24 treebanks in 9 languages from Universal Dependencies v2.1, we observe an average increase in labeled attachment score (LAS) by 3.5 percentage points compared to a single-treebank model, and by 2.2 percentage points compared to simple concatenation of training sets. The treebank embedding technique performs on par with the fine-tuning method of Che et al. (2017) and Shi et al. (2017) but is both simpler and more efficient, since only one model is used at both training and parsing time. Another advantage of the treebank embedding technique is that it is very reliable and, unlike the simple concatenation, never degrades performance compared to the single-model baseline.

In Smith et al. (2018a), we show that treebank embeddings can be used both monolingually, to combine several treebanks for a single language, and multilingually, mainly for closely related languages, especially where one or more of the languages have limited amounts of training data. Moreover, the monolingual and the multilingual case can be seamlessly integrated, so that we can train multilingual models where one or more languages have multiple treebanks. In the 2018 CoNLL Shared Task, we use only 34 models to parse test sets from 84 treebanks and show that this improves LAS by 1.66 percentage points on average over all test sets, by 3.54 for test sets where the corresponding training sets are characterized as "small" by the shared task organizers, and by as much as 7.61 percentage points for low-resource languages that have practically no training data.

5. The 2018 CoNLL Shared Task

The Uppsala team participated in the 2018 CoNLL Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies (Zeman et al., 2018), using a pipeline system consisting of three components. The first component is a model for joint sentence and word segmentation, which uses the BiRNN-CRF framework of Shao et al. (2018) to predict sentence and word boundaries in the raw input text. The second component is a part-of-speech tagger based on Bohnet et al. (2018), which employs a sentence-based character model and also predicts morphological features. The final component is UUParser, which takes the segmented words and their predicted tags and features as input and produces full dependency trees.

After evaluation on the official test sets, the Uppsala system ranked 7th of 27 systems with respect to LAS, with a macro-average F1 of 72.37, making it the highest ranking transition-based parser in this year's shared task. It also reached the highest average score for word segmentation (98.18), universal part-of-speech tags (90.91), and morphological features (87.59). For more information about the Uppsala system in the CoNLL shared tasks of 2017 and 2018, see de Lhoneux et al. (2017a) and Smith et al. (2018a).

References

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with LSTMs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 349–359.

Bernd Bohnet, Ryan McDonald, Goncalo Simoes, Daniel Andor, Emily Pitler, and Joshua Maynez. 2018. Morphosyntactic Tagging with a Meta-BiLSTM Model over Context Sensitive Token Encodings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Wanxiang Che, Jiang Guo, Yuxuan Wang, Bo Zheng, Huaipeng Zhao, Yang Liu, Dechuan Teng, and Ting Liu. 2017. The hit-scir system for end-to-end parsing of universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 52–62.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750.

Miryam de Lhoneux, Yan Shao, Ali Basirat, Eliyahu Kiperwasser, Sara Stymne, Yoav Goldberg, and Joakim Nivre. 2017a. From raw text to Universal Dependencies – Look, no tags! In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 207–217.

Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2017b. Arc-hybrid non-projective dependency parsing with a static-dynamic oracle. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 99–104.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Marco Kuhlmann, Carlos Gómez-Rodríguez, and Giorgio Satta. 2011. Dynamic programming algorithms for transition-based dependency parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 673–682.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Dan Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*.

Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34:513–553.

Joakim Nivre. 2009. Non-projective dependency parsing in expected linear time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP)*, pages 351–359.

Yan Shao, Christian Hardmeier, and Joakim Nivre. 2018. Universal Word Segmentation: Implementation and Interpretation. *Transactions of the Association for Computational Linguistics*, 6:421–435.

Tianze Shi, Felix G. Wu, Xilun Chen, and Yao Cheng. 2017. Combining global models for parsing universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 31–39.

Aaron Smith, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 2018a. 82 treebanks, 34 models: Universal dependency parsing with multi-treebank models. In *Proceedings of the 2018 CoNLL Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*.

Aaron Smith, Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2018b. An investigation of the interactions between pre-trained word embeddings, character models and pos tags in dependency parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Sara Stymne, Miryam Lhoneux, Aaron Smith, and Joakim Nivre. 2018. Parser training with heterogeneous treebanks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 619–625.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*.

The Koala Part-of-Speech and Morphological Tagset for Swedish

Yvonne Adesam, Gerlof Bouma, Richard Johansson

Språkbanken, Dept. of Swedish
University of Gothenburg
{yvonne.adesam,gerlof.bouma,richard.johansson}@gu.se

1. Introduction

The part-of-speech tagset of the Stockholm-Umeå corpus (SUC, Ejerhed et al., 1992) has been the predominant tagset for automatic part-of-speech tagging of Swedish text in the past two decades. Part of its success has been the one million word Stockholm-Umeå corpus itself. The billion word corpora available through Språkbanken's Korp (http://spraakbanken.gu.se/korp) interface have been annotated with the SUC tagset.

The SUC guidelines predate the Swedish language reference grammar *Svenska Akadamiens grammatik* (Teleman et al., 1999, henceforth: SAG), which, since its appearance, has become the point of reference for researchers and students of descriptive/theoretical Swedish linguistics.

There are considerable differences between these two reference points. These are not just differences in inventory and classification, but also in purpose and nature. SUC's annotation system is supposed to be an applicable set of annotation guidelines, whereas SAG is a comprehensive descriptive grammar, which records all kinds of variations and subtleties of the Swedish language. Operationalization of the part-of-speech inventory is not a priority in SAG. Many ambiguities and delimitation problems are noted, but they are not resolved.

Koala was a four year project with the aim of improving the quality and relevance of the linguistic annotation in Språkbanken's processing pipeline. As part of this project, we created the 100 000 token Eukalyptus treebank, for which a new set of guidelines, targeting several levels of annotation (morphological, syntactic and lexical-semantic), was written. Even though they are not a direct operationalization of SAG, the guidelines have SAG as a primary source of inspiration, and consequently our view on Swedish grammar should appeal to the linguists using Språkbanken's resources.

In the following, we present a comparative overview of the part-of-speech and morphological feature tagset of the Koala guidelines, against the background of SAG and SUC. In addition we consider the pioneering work of Teleman (1974, henceforth: MAMBA) and the Swedish implementation of the Universal Dependency part-of-speech tag guidelines (Nivre, 2014, universaldependencies.org/sv/, henceforth: UDP).

2. Design principles

Our part-of-speech definitions are, as is common, based on inflectional/derivational, distributional and/or denotational arguments (ordered from most to least important). However, since we developed the guidelines for the phrase and dependency annotation (Adesam et al., 2015) simultaneously, distributional evidence needs special attention. A distributional generalization could in principle be captured through assignment of a part-of-speech or of a grammatical function. All else being equal, the Koala guidelines prefer using grammatical function to capture generalizations, keeping the part-of-speech assignment constant. For instance, head-like adjectives in NPs (*de anställda* 'the employed.ADJ'), are considered to be adjectives; verb particles are typically prepositions or adverbs (*ta hit*, 'bring' lit. 'take here.ADV') but could be other parts-of-speech; and predicatively used interjections are just that and not adjectives (*vara bläää* 'to be yuck').

Following Borin et al. (2013), we also assign parts-of-speech to multi-word units. In these cases, the balance between the different types of evidence often shifts. For instance, multi-word nouns may look very different from other nouns with regard to inflection, multi-word adjectives may look just like prepositional phrases, and multi-word verbs need not follow the same distributional facts as single-word verbs.

Part-of-Speech		Features
AB	Adverb	degree, relative
AJ	Adjective	degree, gender, number, definiteness
EN	Proper name	
IJ	Interjection	
KO	Coordinator	
NN	Common noun	gender, number, definiteness
NU	Numeral	
PE	Preposition	
PO	Pronoun	gender, number, definiteness, form, relative
SU	Subordinator	
SY	Symbol	type
UO	Foreign word	
VB	Verb	mood/finiteness, voice, tense

Table 1: Koala part-of-speech and morphological labels.

3. Parts-of-Speech

The parts-of-speech are divided into 13 main categories, which can get a number of different features describing subdivisions or morphological properties. There are three general features that may apply to any part-of-speech: 1) abbreviation (*ex.* 'example', *s.a.s* 'so to speak'), 2) elliptical coordination (*hög- [och lågstadium]* 'upper sec-

ondary [and lower primary school]', *[lägenhetssäljare och] -köpare* '[apartment seller and] buyer'), and 3) genitive. The genitive label is used for both lexical and phrasal genitives, but is always applied at the word level (*husets [tak]* 'house's.NN.DEF.GEN [roof]', *[mannen jag] sågs [fru]* '[the man I] saw's.VB.GEN [wife]'). Our approach contrasts with other approaches in that we do not consider Swedish to have case, and that the label for the genitive marker may be used for anything. SUC, SAG, and UDP have the nominative vs genitive case distinction on nouns, proper names, adjectives, pronouns, participles and numerals. MAMBA holds a middle ground as it has a feature 'genitive suffix', which, however, only applies to certain parts-of-speech.

In addition to these general features, there are part-of-speech specific features. The complete set of parts-of-speech and their features are listed in Table 1.

3.1 Nouns and Proper Names

In the Koala tagset, nouns are categorized for gender (neuter, common), number (singular, plural), and definiteness (definite, indefinite). Proper names have no specific features. MAMBA's noun category includes both common and proper nouns, but does not specify inherent morphological features such as number and gender. The MAMBA guidelines also contain a number of finer categories, such as adjectival and verbal nouns. SUC and UDP treat common and proper nouns as two different categories.

To deal with the fuzzy distinction between common nouns and proper names, we restrict the use of the proper name category to obvious examples of names, such as person names, names of locations, and cases where the denotation does not match the description. Therefore, *Riksdagen* 'Parliament' is labelled as a common noun, regardless of its uppercase initial, while *Hasselbacken* (lit. 'hazel hill') is labelled as a proper name when used to refer to the restaurant of that name.

3.2 Pronouns

Pronouns are categorized for gender (common, neuter, masculine), number (singular, plural), definiteness (definite, non-definite), form (subject, object, possessive), and whether they are interrogative or relative.

The pronoun category in Koala combines a number of different types that are distinguished in the other standards. For instance, SAG, SUC and MAMBA distinguish many different types of pronouns. SUC and UDP have different categories for determiners and (non-dependent) pronouns – we consider this to be a difference in syntactic function. SAGs relational pronouns, such as *samma, höger, egen, sista* 'same, right(most), own, last', are classified as adjectives in Koala.

3.3 Numerals

Numerals are the non-inflected cardinals and can be written both with numbers and letters, e.g. *tre, 42, femtioelva* 'three, 42, many' (latter, lit: 'fifty-eleven'). Apart from lacking inflection, they stand out by their unusual word formation rules. Ordinals, however, are grouped with adjectives on distributional grounds. Words like *miljon, miljard* 'million, billion' inflect like nouns, and are thus tagged as such, as are quantity denoting words like *dussin* 'dozen'.

3.4 One

The numeral, indefinite article, and pronoun *en, ett* 'a/one.COM, a/one.NEU', coincide in form. SAG discusses *en, ett* in the context of numerals and in the context of articles/pronouns, while still treating it as one item. Although there are clear cut cases, there are many genuinely ambiguous occurrences. MAMBA gives it a category of its own, thus recognizing its difficult status. SUC categorizes it as either a special cardinal with gender distinction, a determiner, or a pronoun – which unfortunately leads to inconsistent tagging by the annotators. UDP also distinguishes numeral, determiner and pronoun *en, ett*.

As mentioned, the Koala tagset does not distinguish between pronouns and determiners. Moreover, we consider the fact that *en, ett* inflects, even in its 'numeral use', to be sufficient grounds for its status as pronoun. In our view, the difference between numeral and pronominal use does not correspond to a difference in part-of-speech.

3.5 Adjectives

Adjectives are categorized for degree (positive, comparative, superlative), gender (neuter, common, masculine), number (singular, plural), and definiteness (definite, indefinite). Adjectives can also have adverbial use. In contrast to SUC and SAG, but in similarity to UDP, participles and ordinals are tagged as adjectives.

3.6 Prepositions

Prepositions have no specific features. Compared to SUC, this category also contains many verb particles (see below).

3.7 Adverbs

Adverbs are categorized for degree (positive, comparative, superlative), and whether they are relative/interrogative. SAG states that the boundary between adjectives and adverbs is unclear. In the Koala tagset, adverbs cannot be inflected for number, gender, or definiteness. In terms of distribution, adverbs are not used prenominally or predicatively.

SUC has a separate category for interrogative and relative adverbs, which we mark with a feature. UDP separates negations from adverbs and tags them as particles, while we treat them as any other adverb. MAMBA has a very fine-grained set of categories for different types of adverbs.

3.8 Verbs

Verbs are categorized for mood or finiteness (indicative, conjunctive, imperative, supine, infinitive), voice (active, s-form), and tense (present, past). UDP marks auxiliaries, and views copulas as auxiliaries. We follow SAG's reasoning that the border between auxiliaries and main verbs is fuzzy, and do not distinguish auxiliaries from other verbs. SUC presents a similar reasoning.

In similarity to the SUC tagset, we mark s-forms rather than passives, because of the inherent ambiguity in e.g. *barnen pussades*, with a passive meaning, 'the children were kissed', an active absolute meaning, 'the kids kissed

(someone)', and an active reciprocal meaning, 'the children kissed each other'.

3.9 Verb Particles

Verb particle is not a part-of-speech in Koala. SAG mentions the function of *particle adverbial* for prepositions and adverbs. In line with this view, we consider verb particle to be a syntactic function. Most verb particles are prepositions or adverbs, and are annotated as such, for instance *[hoppa] i* '[jump] in', *[slå] igen* 'close' (lit. '[hit] together'). Others are nouns or adjectives, such as *[göra] skada* 'damage' (lit., '[make] damage'), *sitta fast* 'be stuck' (lit. '[sit] tight'). In some cases, the part-of-speech category is ad hoc, as the word only occurs as a verb particle, for instance *[slå] slint* 'fail, misfire' (lit. '[hit] slip.NN').

Although UDP has a 'particle' category, this is used for the infinitive marker and negations. Verb particles are treated like in Koala. SUC tags verb particles as a separate category. This category was added during the annotation process and is not mentioned in the description of the tag system (Ejerhed et al., 1992).

3.10 Subordinators and Coordinators

Subordinators and coordinators have no specific features. The infinitive marker *att*, which is considered it's own category by SUC, and a particle by UDP, is a subordinator in Koala. The word *som* is tagged as a relative pronoun, conjunction, or relative adverb in SUC. In accordance with SAG, we always consider it to be a subordinator, apart from cases where it is a coordinator, as in *barn som vuxna* 'children and grown ups'.

3.11 Interjections

Interjections have no specific features. The category is fairly similar between different tagsets.

3.12 Symbols

Symbols are any kind of non-alpha-numeric token, and are labelled as either punctuation or other symbol. Punctuation includes all types of sentence internal and external delimiters. The subcategory 'other symbols' applies to different types. They are special in that they are allowed to be the head of all kinds of phrases. For instance, the symbol '<3' kan be used to head a verb phrase (*Jag <3 bebisar men inte när de skriker* 'I <3 babies, but not when they scream'), or a noun phrase (*Fått mycket <3 och många nya kontakter!* 'Got lots of <3 and many new contacts!').

Neither SUC nor Mamba have a category for symbols, although they have a category for punctuation. UDP has two different categories for punctuation and symbols.

3.13 Foreign words

Foreign words have no specific features. Other parts-of-speech often take priority over the foreign word label. For example, foreign names are tagged as names. As with symbols, we allow foreign words to figure as head in any phrase (*ett mycket overdue slut* 'a very overdue end'). The category is also used in SUC, while UDP only has the category 'other'.

4. Conclusions

We have described the Koala part-of-speech tagset in comparison to a number of other standards. The tagset is used in the Eukalyptus treebank, which is made freely available through Språkbanken.

Acknowledgements

The Eukalyptus treebank was developed within the Koala project, funded by Riksbankens Jubilemsfond, grant number In13-0320:1.

References

Yvonne Adesam, Gerlof Bouma, and Richard Johansson. 2015. Defining the Eukalyptus forest – the Koala treebank of Swedish. In *Proceedings of NODALIDA*, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

Lars Borin, Markus Forsberg, and Lennart Lönngren. 2013. SALDO: a touch of yin to WordNet's yang. *Language Resources and Evaluation*, 47(4):1191–1211.

Eva Ejerhed, Gunnel Källgren, Ola Wennstedt, and Magnus Åström. 1992. The linguistic annotation system of the Stockholm-Umeå corpus project - description and guidelines. Technical report, Department of Linguistics, Umeå University.

Joakim Nivre. 2014. Universal Dependencies for Swedish. In *Proceedings of SLTC)*, Uppsala.

Ulf Teleman, Staffan Hellberg, and Erik Andersson. 1999. *Svenska Akademiens Grammatik*. Svenska Akademien, Stockholm.

Ulf Teleman. 1974. *Manual för grammatisk beskrivning av talad och skriven svenska*. Studentlitteratur, Lund.

Is the whole greater than the sum of its parts? A corpus-based pilot study of the lexical complexity in multi-word expressions

David Alfter, Elena Volodina

University of Gothenburg
Göteborgs universitet, institutionen för svenska språket, Box 200, 405 30 Göteborg
firstname.lastname@gu.se

1. Introduction

Multi-word expressions (MWEs) have been found to be important predictors of general proficiency in language learners, which is reflected in the fact that learners of higher proficiency levels tend to use more multi-word expressions (Erman et al., 2016, p. 111), but they are often under-used by learners at earlier levels (Ellis et al., 2015, p. 362). Given these findings, there seems to be a need to focus on MWEs in language learning settings and their role at different levels of proficiency.

When looking at MWEs from a language learner perspective, we are interested in *when* learners of different proficiency levels should be able to deal with certain MWEs. This information is important when generating exercises for language learners of different proficiency levels, so as to ensure that the learner will be able to understand and deal with the language the exercise item contains.

Previous work for Swedish word-based lexical complexity has focused on single words only (Alfter and Volodina, 2018), assessing lexical complexity of single words in terms of different orthographic, morphological, semantic and contextual features and then assigning a label to each word using the Common European Framework of Reference (CEFR) (Council of Europe, 2001) scale. This scale ranges from A1 (absolute beginner) over A2, B1, B2, C1 to C2 (native-like). Each lexical item is assigned one target CEFR level. It should be pointed out that the CEFR levels are not to be understood as CEFR levels of lexical units, but as labels which correspond to the earliest proficiency level at which a learner of that level should be able to understand the lexical item.

We hope to extend the aforementioned work by comparing CEFR levels assigned to MWEs in comparison to the levels assigned to each of their constituents.

2. Experimental setup

The starting point for this study is the Swedish CEFRLex resource SVALex (François et al., 2016). This resource is based on the COCTAILL corpus (Volodina et al., 2014), a corpus of CEFR-graded textbooks. The resource lists how often a lexical unit occurs at different CEFR levels. The resource not only contains single words, but also MWEs. We have mapped each entry to a single level following first-occurrence approach as in Gala et al. (2013), Gala et al. (2014), Alfter and Volodina (2018), meaning that each entry was assigned as target level the level of the text where it first occurred. The division of MWEs into different part-of-speech labels is based on the automatic assignment of part-of-speech tags from Sparv[1], the annotation tool used for tagging the corpus. Besides providing part-of-speech tags for single words, it also automatically recognizes MWEs and assigns a single part-of-speech tag to the whole expression.

In total, there are 1444 entries identified as MWEs, of which 207 are phrasal/particle verbs which are treated separately from verbal MWEs. The reason for treating particle verbs as separate category from the rest of the MWEs is that we want to see whether particle verbs behave differently. The intuition is that most particle verbs are learned at an early stage as chunks and that (some of) the constituents thereof often occur on their own with a different or more abstract meaning at higher levels, for example *dyka upp* "to appear, arrive, show up" vs *dyka* "to dive" or *mata in* "to input" vs *mata* "to feed". We used the upcoming version 3 of Saldo to identify possible particle verbs [2].

The remaining verbal MWEs consist of figurative expressions such as *peka med hela handen* "to forcefully instruct/suggest" (lit. to point with the whole hand) or *spetsa öronen* "to be all ears" (lit. to sharpen the ears). As the identification of phrasal verbs was done based on a word list, and given the non-exhaustive nature of the list, reflexive verbs and particle verbs that were not in our resource are also present in the general verbal MWE category.

Since most of the MWEs in this data set are figurative in nature, and since figurative language is non-compositional with regards to the meaning of its parts, we want to explore how MWEs behave with regards to CEFR levels as a whole versus the separate levels of its parts.

To this aim, we go over all MWE entries from the mapped, single-label version of SVALex and check, for each entry, whether at least one of the components of the MWE can be found with an assigned level in the resource. However, since MWEs can include inflected word forms (e.g. *ha många järn i elden* "to have several irons in the fire") and the SVALex resource only lists lemmas, we use a second resource as a backup. This second resource is calculated from the COCTAILL textbook corpus (Volodina et al., 2014) and the SweLL learner essay pilot corpus (Volodina et al., 2016), similarly to other CEFRLex resources[3], but taking into account all occurring word forms instead of

[1] https://spraakbanken.gu.se/sparv

[2] The authors would like to express their gratitude towards Lars Borin for sharing with them a preliminary version of Saldo 3.

[3] http://cental.uclouvain.be/cefrlex/

lemmas.

For each MWE that we can find at least one constituent in one of the two lookup resources, we compare the level assigned to the MWE against the level(s) of its constituent(s). In example 1 we can see that *peka med hela handen* was assigned level C1 while the highest level of its constituents is *peka* at level B1. Example 2 shows a MWE where only one of its constituents was found in the word lists while the level for *spetsa* is unknown.

We count how often the level of the MWE as a whole is higher or equal to the highest level among its constituents and how often the level of the MWE is lower than the highest level of its constituents. In case the level of the MWE is lower than the level of one of its constituents, we also calculate whether the difference in levels is within one CEFR level or not. In example 3, the level of the whole expression *till höger* 'to the right' is assigned level A1, but the highest level among its constituents is *höger* 'right' with level A2. However in this case, the difference in levels is within one CEFR level. As such, it will not be counted as being "less or equal" in exact matching (left half of table 1) but it will be counted as "less or equal" in "within one CEFR level" (right half of table 1).

(1) peka med hela handen ⇒ C1
   a. peka ⇒ B1
   b. med ⇒ A1
   c. hela ⇒ A1
   d. handen ⇒ A2

(2) spetsa öronen ⇒ C1
   a. spetsa ⇒ UNKOWN
   b. öronen ⇒ A2

(3) till höger ⇒ A1
   a. till ⇒ A1
   b. höger ⇒ A2

3. Results

Table 1 shows the results broken down by part-of-speech. The *Less or equal* column counts how often the highest *constituent* level was *lower* or equal to the MWE level. The *Higher* column shows how often the highest constituent level was higher than the MWE level. The left part of the table shows *exact* counts, i.e. how often an MWE was assigned a level higher or equal to the highest level of its constituents. The right part of the table shows the *within-one-level* counts where, if the level of the MWE was lower than the highest level of its constituents but the deviation was within one CEFR level, it was counted as being equal. As can be seen from the results, most MWEs are made up of words with complexity levels lower or equal to the whole expression.

This table also shows that there is an imbalance with regard to part-of-speech of the MWEs. There are many verbal and adverbial MWEs but not many nominal of adjectival MWEs.

4. Discussion and Conclusion

We have found that most MWEs follow the expected pattern where the target level assigned to the whole expression is higher or equal to the highest level assigned to any of its components. This also holds true for the majority of particle verbs. We found only 17% of MWEs not following this pattern in exact counting and only 5% of MWEs not following this pattern in within-one-level counting. The 17% and 5% of MWEs that did not follow this pattern include for example *spänna fast* 'to clamp, to strap' assigned level A2 as a whole but level B2 for its two constituents as shown in example 4 or *skaka hand* 'to shake hands' which was assigned A2 as overall level but *skaka* 'to shake' was assigned level C1 as shown in example 5. Some of the deviations are probably due to data sparsity. Indeed, in the absence of enough data, level assignment will be wrong; if a word only occurs once in the corpus, it will necessarily be assigned the level at which it occurred. Another cause for deviations from the expected pattern is that some multi-word expressions simply are learned at earlier levels, as chunks, than their constituents. Examples include *andas in/ut* 'to breathe in/out' assigned level A2 vs *andas* 'to breathe' assigned level B1, or, in exact matching mode, *växa upp* 'to grow up', assigned level A1, versus *växa* 'to grow', assigned level A2, shown respectively in examples 6 and 7

(4) spänna fast ⇒ A2
   a. spänna ⇒ B2
   b. fast ⇒ B2

(5) skaka hand ⇒ A2
   a. skaka ⇒ C1
   b. hand ⇒ A2

(6) andas in/ut ⇒ A2
   a. andas ⇒ B1
   b. in/ut ⇒ A1

(7) växa upp ⇒ A1
   a. växa ⇒ A2
   b. upp ⇒ A1

Our findings seem to suggest that most MWEs are understandeable at the earliest at the highest CEFR label among its constituents. This finding can be useful when selecting non-graded MWEs for students of different proficiency levels.We will integrate this knowledge into the automatic exercise generation for language learners in the experimental Intelligent Computer-Assisted Language Learning (ICALL) platform Lärka [4]. When generating exercises that target different proficiency groups, it is important to select adequate stimuli; if the stimulus is too hard and the learner cannot deal with it, they will get frustrated. If the stimulus is too easy, the learner will get bored. In a system that dynamically adapts the level of exercises to the current learner's proficiency, having such insights about when a (non-graded) multi-word expression can be expected to be understood, based solely on the assigned CEFR levels of its constituents, is important.

[4] https://spraakbanken.gu.se/larka

	Exact			Within one CEFR level		
Part-of-Speech	Less or equal	Higher	Percent (Higher/All)	Less or equal	Higher	Percent (Higher/All)
Nouns	28	8	0.22	33	3	0.08
Verbs	324	75	0.19	375	24	0.06
Adjectives	31	7	0.18	36	2	0.05
Adverbs	336	71	0.17	376	31	0.08
Particle verbs	352	42	0.11	391	3	0.01
Prepositions	29	10	0.26	34	5	0.13
Proper names	50	9	0.15	58	1	0.02
Conjunctions	4	2	0.33	5	1	0.17
Interjections	28	10	0.26	34	4	0.11
Pronouns	21	4	0.16	23	2	0.08
Subjunctions	1	1	0.50	2	0	0.0
All	1204	239	0.17	1367	76	0.05

Table 1: Results by part-of-speech

In the future, we would like to evaluate the findings with learners and teachers. It could also be interesting to investigate whether the same findings hold true for learner essays, i.e. do learners produce MWEs at the earliest only after having produced at least one of the MWEs constituents first? Finally, it would also be interesting to test the approach on a bigger corpus.

References

David Alfter and Elena Volodina. 2018. Towards Single Word Lexical Complexity Prediction. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*.

Council of Europe. 2001. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Press Syndicate of the University of Cambridge.

Nick C. Ellis, Rita Simpson-Vlach, Ute Römer, Matthew Brook ODonnell, and Stefanie Wulff. 2015. Learner corpora and formulaic language in second language acquisition research. In Sylviane Granger, Gaëtanelle Gilquin, and Fanny Meunier, editors, *The Cambridge handbook of learner corpus research*, chapter 16, pages 357–378. Cambridge University Press.

Britt Erman, Fanny Forsberg Lundell, and Margareta Lewis. 2016. Formulaic language in advanced second language acquisition and use. In Kenneth Hyltenstam, editor, *Advanced proficiency and exceptional ability in second languages*, chapter 4, pages 111–147. Walter de Gruyter Boston.

Thomas François, Elena Volodina, Ildikó Pilán, and Anaïs Tack. 2016. SVALex: a CEFR-graded Lexical Resource for Swedish Foreign and Second Language Learners. In *LREC*.

Núria Gala, Thomas François, and Cédrick Fairon. 2013. Towards a French lexicon with difficulty measures: NLP helping to bridge the gap between traditional dictionaries and specialized lexicons. *E-lexicography in the 21st century: thinking outside the paper., Tallin, Estonia*.

Núria Gala, Thomas François, Delphine Bernhard, and Cédrick Fairon. 2014. Un modèle pour prédire la complexité lexicale et graduer les mots. In *TALN 2014*, pages 91–102.

Elena Volodina, Ildikó Pilán, Stian Rødven Eide, and Hannes Heidarsson. 2014. You get what you annotate: a pedagogically annotated corpus of coursebooks for Swedish as a Second Language. In *Proceedings of the third workshop on NLP for computer-assisted language learning at SLTC 2014, Uppsala University*, number 107. Linköping University Electronic Press.

Elena Volodina, Ildikó Pilán, Ingegerd Enström, Lorena Llozhi, Peter Lundkvist, Gunlög Sundberg, and Monica Sandell. 2016. SweLL on the rise: Swedish learner language corpus for european reference level studies. *arXiv preprint arXiv:1604.06583*.

Negation detection in Norwegian medical text: Porting a Swedish NegEx to Norwegian Work in progress

Andrius Budrionis*, Hercules Dalianis*†, Kassaye Yitbarek Yigzaw*, Alexandra Makhlysheva*, Taridzo Chomutare*

*Norwegian Centre for E-health Research, University Hospital of North Norway
Tromsø, Norway
†DSV/Stockholm University
Kista, Sweden
Andrius.Budrionis@ehealthresearch.no,
Hercules.Dalianis@ehealthresearch.no,
Kassaye.Yitbarek.Yigzaw@ehealthresearch.no,
Alexandra.Makhlysheva@ehealthresearch.no,
Taridzo.Chomutare@ehealthresearch.no

Abstract

This paper presents an initial effort in developing a negation detection algorithm for Norwegian clinical text. An evaluated version of NegEx for Swedish was extended to support Norwegian clinical text, by translating the negation triggers and adding more negation rules as well as using a pre-processed Norwegian ICD-10 diagnosis code list to detect symptoms and diagnoses. Due to limited access to the Norwegian clinical text the Norwegian NegEx was tested on Norwegian medical scientific text. NegEx found 70 negated symptoms/diagnoses in the text combined of 170 publications in the medical domain. The results are not completely evaluated due to the lacking gold standard. Some challenging erroneous tokenizations of Norwegian words were found in addition to the need for improved preprocessing and matching techniques for the Norwegian ICD-10 code list. This work pointed out the weaknesses of the current implementation and provided insights for future work.

1. Introduction

Free-text is an integral part of clinical documentation. Regardless of the maturity of the electronic health record (EHR) systems and initiatives to structure free-text information, it is still highly prevalent and widely accepted by clinicians. The nature of the free-text fits well with the level of uncertainty in diagnosis-setting process and allows clinicians to document patient history in a way, which is difficult to put into a rigid structure.

While free-text is a recognised and often preferred way of clinical documentation, it presents major challenges for data reuse. Better use of health data accumulated in EHRs is a important aim for future health care worldwide and Norway is no exception (Direktoratet for e-helse, 2017). Retrieval of knowledge from free-text documentation is a complex task. Individualistic patterns for describing patient status, various dialects and spelling errors are well-known challenges to be addressed.

Negated symptoms and diagnoses are part of diagnosis-setting process and are common in medical text. Such concepts have to be identified and handled accordingly in the knowledge retrieval process ensuring that they are not mixed with the positive symptoms and diagnoses, (Groopman, 2007).

Negation detectors (NegEx) are available in several languages, however, to our knowledge, Norwegian NegEx has not yet been developed. A Swedish version of NegEx has been presented earlier (Skeppstedt, 2011) and yielded satisfactory precision and recall when tested on Swedish medical text. This paper presents the development and initial tests of the Norwegian NegEx using earlier presented Swedish work as a starting point.

2. Related research

The existing NegEx algorithms and dictionaries are language-specific and translation is not always sufficient to adopt them in a new language. The most of effort in the field was put into developing negation detection algorithms for English (Chapman et al., 2001; Mehrabi et al., 2015; Peng et al., 2018; Ou and Patrick, 2015). However, tools for processing other languages, such as Swedish (Skeppstedt, 2011), French, German (Chapman et al., 2013), and Chinese (Kang et al., 2017) exist.

The English version of NegEx obtained a precision of 84.5% and a recall of 82.4% when tested on discharge summaries (Chapman et al., 2001). The Swedish version was ported from the English original version of NegEx and obtained a precision of 87.9% and a recall of 91.7% for negation cues on Swedish clinical text. (Skeppstedt, 2011). Tanushi et al. compared various approaches to negation detection (Swedish NegEx, PyConTextNLP and SynNeg). PyConTextNLP is an extension of the English NegEx and SynNeg is based on a syntactic parser that considers sentence boundaries. All three systems produced similar results; SynNeg performed better on long and complex sentences (Tanushi et al., 2013).

Attempts to improve the precision and recall of the original NegEx (English) are desribed in the literature. For in-

stance, Mehrabi et al. included analysis of dependencies between negation terms and other concepts in the sentence aiming to decrease the number of false positives in the original NegEx. The attempt was only partially successful and presented higher precision and recall scores only in one out of three selected corpora (Mehrabi et al., 2015). The findings highlight interoperability concerns, which may be caused by varying practices in documenting patient condition, spelling errors and potentially incomplete dictionary of negation terms. A recent work by Peng et al. presented significant improvement to the original NegEx performance (on average 9.5% higher precision and 5.1% higher F1-score) when tested on two corpora containing radiology reports (Peng et al., 2018). The improvement was achieved utilising patterns on universal dependencies that help to identify the scope of negation triggers.

3. Methods and Data

The Swedish version of NegEx (Skeppstedt, 2011) was ported to Norwegian and evaluated on Norwegian medical scientific text in the domain of gastrointestinal surgery, while we are waiting for access to clinical text from the EHR. The reason to choose gastrointestinal surgery is our future focus on the analysis of free-text notes documenting this type of surgery in patient records in the University Hospital of North Norway.

The Norwegian medical scientific text was downloaded from the *Tidsskrift Den norske legeforening*[1] and transformed from portable document format (PDF) to pure UTF-8 coded text. Specifically scientific publications in the field of gastrointestinal surgery were chosen, in total 170 articles containing 294,745 words.

3.1 Porting the Norwegian NegEx from Swedish

Swedish and Norwegian are closely related languages almost completely comprehensible for speakers from both language groups. They have similar grammar, however spelling is rather different. Norwegian bokmål is a preferred written standard for about 90% of the population and its spelling is derived from Danish language[2].

Since Norwegian and Swedish grammars are similar, for creating the Norwegian version of NegEx, Swedish version of NegEx was taken as a basis. The grammatical differences between Swedish and English are described by Skeppstedt (Skeppstedt, 2011). The list of Swedish negations triggers was translated to Norwegian. Some expressions were added in form of new negation rules in cases when a phrase could be translated in several ways, spelled differently or have both a direct and a reversed word order, and verbs in passive and active voices. Moreover the pseudo negation *ikke minst* (Eng. "at least") was added in a form of a new pseudo negation rule. Further, the English version of NegEx was also taken into account when exact match between Norwegian and Swedish was not available.

Starting with the Swedish version of NegEx with 40 negation rules, additional 26 negation rules were added to a total of 67 negation rules in the Norwegian version of the NegEx. The distribution was as follows: 15 Norwegian POST-negation rules (for instance triggers such as, *negativt, mangler, benektet* (Eng. "negatively", "missing", "denied"), 34 PREN-negation rules (for instance triggers such as, *aldri, ikke, nekte* (Eng. "never", "not", "deny") and 18 PSEUDO-negation rules (for instance triggers such as, *ikke utelukker, ikke forårsaket, ikke bekreftet* (Eng. "not excluded", "not caused", "not confirmed"). POST-negation rules are used to find negated term after the negation trigger and PREN-negation rules are used to find negated term before the negation trigger.

NegEx requires a list of symptoms and diagnoses to be matched to the medical text for negation. A Norwegian version of *ICD-10*[3] from year 2017 was used for this purpose. The Norwegian ICD-10 list contains 19,597 codes and their descriptions in free-text. The ICD-10 list was pre-processed by removing stop-words using the Norwegian Snowball stop-word list[4], and manually removing long modifiers adhoc, for example *Annen spesifisert...* (Eng: Other specified). Finally, single-word symptoms and diagnoses, enabling matching between matching list and the analysed text, were also extracted manually and gave a final list of 19,628 terms.

In addition to the ICD-10 list, 23 significant words from the gastrointestinal surgery domain (Table V in (Soguero-Ruiz et al., 2016)) were added into the domain list. NegEx pre-processed automatically both lists and compiled them into total 19,651 terms.

An example of the Norwegian NegEx correctly functioning on a medical text is presented in the Figure 1.

> *Han var ved innkomst hemodynamisk upåvirket og hadde ikke tegn til [NEGATED]peritonitt[NEGATED].*
> (Eng. "At arrival he was not affected hemodynamically and had no signs of [NEGATED]peritonitis[NEGATED]".)

Figure 1: An example of machine based proper negation detection.

Results of applying NegEx on a corpus containing text from 170 scientific medical publications are presented in Table 1.

[1] Tidsskrift - Den norske legeforening, https://tidsskriftet.no/spesialitet/gastroenterologisk-kirurgi. Accessed 2018-07-16.

[2] Språkrådet, Norwegian: Bokmål vs. Nynorsk, http://www.sprakradet.no/Vi-og-vart/Om-oss/English-and-other-languages/English/norwegian-bokmal-vs.-nynorsk/. Accessed 2018-08-14.

[3] Kodeverket ICD-10, https://ehelse.no/standarder-kodeverk-og-referansekatalog/helsefaglige-kodeverk/kodeverket-icd-10-og-icd-11. Accessed 2018-07-16.

[4] Norwegian stop-word list, Snowball. http://snowball.tartarus.org/algorithms/norwegian/stop.txt. Accessed 2018-07-21.

	Words	Symptoms/diagnoses	Negations
Results on whole corpus	294,745	1,835	70
Manually labelled sub corpus	75,614	-	29
Automatically labelled sub corpus	75,614	526	15

Table 1: Results of Norwegian NegEx applied on Norwegian medical scientific text in gastrointestinal surgery.

4. Results

When executing NegEx on the 170 articles, the system found 70 negated symptoms/diagnoses. NegEx was also executed on a smaller sub corpus (1/4 of the original corpus) that gave in total 15 automatic negation labels, manually labelling gave 29 negations. No precision and recall measurements has been calculated since the final labelling is not ready yet.

5. Lessons learned

Lack of labelled corpora containing clinical Norwegian text, which could be used for the evaluation of the algorithm to validate the NegEx, was the first challenge to be addressed. Therefore, medical literature published in a scientific medical journal was used as a starting point. Labelling of this corpus is currently in progress, and needs to be carried out during several labelling rounds jointly with clinically trained personnel. Labelled corpus will allow us to evaluate the performance of the NegEx in terms of widely accepted precision and recall measures.

The automatically identified negations were manually analysed by the authors. This process resulted in additions to the dictionary of negation triggers and extra negation rules in the NegEx algorithm. Parts of the corpus were manually inspected, identifying negations, which were not captured by the NegEx. But also erroneous tokenization of the text such as *ikke-operable metastaser* (Eng. "inoperable metastases") became machine labelled as *ikke -operable [NEGATED]metastaser[NEGATED]* (Eng. "in -operable [NEGATED]metastases[NEGATED]") meaning that *ikke-operable* (Eng. "inoperable") must be tokenized in one chunk, such that *ikke* (Eng. "not") does not become a negation trigger to metastaser.

Such error analysis highlighted the flaws of the current algorithm, which are to be addressed in the next version of the Norwegian NegEx. The current algorithm uses exact string-matching strategy, looking for an exact match between the text in the corpus and the ICD-10 code list of symptoms and diagnoses.

Currently we are also experimenting with approximate string-matching techniques delivering a more robust mechanism to capture the negated symptoms/diagnoses. The clinical findings mentioned in the text, in many cases, are incomplete or phrased differently in comparison to the ones in the ICD-10 list. Therefore, many of them were not captured by the NegEx, hence one method is to pre-process the ICD-10 code list using bi- or tri-grams, similar to the approach by (Skeppstedt, 2011), to remove the most common occurrences of phrases and long modifiers such as for example *Annen spesifisert...* (Eng. "Other specified") to improve the matching of the symptoms and diagnoses term to the medical text.

Future work will hence address both the aforementioned weaknesses of the algorithm as well as labelling of the electronic patient records from gastrointestinal surgery.

Acknowledgements

Great thanks to Maria Skeppstedt for good advices regarding the programming of the Norwegian NegEx algorithm. This work was funded by the Helse Nord grant (HNF1395-18) to the Norwegian Centre for E-health Research.

References

Wendy W. Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper, and Bruce G. Buchanan. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics*, 34(5):301–310.

Wendy W. Chapman, Dieter Hilert, Sumithra Velupillai, Maria Kvist, Maria Skeppstedt, Brian E. Chapman, Michael Conway, Melissa Tharp, Danielle L. Mowery, and Louise Deleger. 2013. Extending the NegEx Lexicon for Multiple Languages. *Studies in health technology and informatics*, 192:677–681.

Direktoratet for e-helse. 2017. Nasjonal e-helsestrategi 2017-2022. Technical report, Direktoratet for e-helse.

Jerome E. Groopman. 2007. *How doctors think*. Houghton Mifflin Company, New York.

Tian Kang, Shaodian Zhang, Nanfang Xu, Dong Wen, Xingting Zhang, and Jianbo Lei. 2017. Detecting negation and scope in Chinese clinical notes using character and word embedding. *Computer Methods and Programs in Biomedicine*, 140:53–59, March.

Saeed Mehrabi, Anand Krishnan, Sunghwan Sohn, Alexandra M. Roch, Heidi Schmidt, Joe Kesterson, Chris Beesley, Paul Dexter, C. Max Schmidt, Hongfang Liu, and Mathew Palakal. 2015. DEEPEN: A negation detection system for clinical text incorporating dependency relation into NegEx. *Journal of Biomedical Informatics*, 54:213–219, April.

Ying Ou and Jon Patrick. 2015. Automatic negation detection in narrative pathology reports. *Artificial Intelligence in Medicine*, 64(1):41–50, May.

Yifan Peng, Xiaosong Wang, Le Lu, Mohammadhadi Bagheri, Ronald Summers, and Zhiyong Lu. 2018. NegBio: a high-performance tool for negation and uncertainty detection in radiology reports. *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, 2017:188–196.

Maria Skeppstedt. 2011. Negation detection in Swedish clinical text: An adaption of NegEx to Swedish. *Journal of Biomedical Semantics*, 2(Suppl 3):S3.

Cristina Soguero-Ruiz, Kristian Hindberg, José Luis Rojo-Álvarez, Stein Olav Skrøvseth, Fred Godtliebsen,

Kim Mortensen, Arthur Revhaug, Rolv-Ole Lindsetmo, Knut Magne Augestad, and Robert Jenssen. 2016. Support vector feature selection for early detection of anastomosis leakage from bag-of-words in electronic health records. *IEEE journal of biomedical and health informatics*, 20(5):1404–1415.

Hideyuki Tanushi, Hercules Dalianis, Martin Duneld, Maria Kvist, Maria Skeppstedt, and Sumithra Velupillai. 2013. Negation scope delimitation in clinical text using three approaches: Negex, pycontextnlp and synneg. In *19th Nordic Conference of Computational Linguistics (NODALIDA 2013), May 22-24, 2013, Oslo, Norway*, pages 387–474. Linköping University Electronic Press.

Targeted Data-Driven Dependency Parsing for Japanese and Korean

Andrew Dyer, Sara Stymne

Department of Linguistics and Philology
Uppsala University
andrew.dyer.6854@uu.se, sara.stymne@lingfil.uu.se

Abstract

We describe methods for improving data-driven dependency parsing for Japanese and Korean, making use of syntactic similarities between the two languages to create a unified model. We draw on the phrasal unit-focused Triplet/Quadruplet model (Kanayama et al., 2000) and extend this to a feature model for Japanese that works in a transition parser. Our feature model can be applied cross-lingually to Korean via selective delexicalisation, with no requirement for separate models. We find that our model yields improvements in LAS and UAS when applied both monolingually to Japanese, and cross-lingually to Korean as a low resource language.

1. Introduction

While much recent work on data-driven dependency parsing is applied to a wide range of languages (Zeman et al., 2018), typically the same models are applied to all languages without taking into account features of specific languages. This is in contrast to some targeted work on dependency parsing of specific languages, such as the largely rule-based parser for Japanese by Kanayama et al. (2000). In the first part of this paper, we describe how we can use insights from the Triplet/Quadruplet model of Kanayama et al. (2000) to extend a feature model for data-driven dependency parsing, which can improve parsing for Japanese.

A language with many syntactic similarities to Japanese is Korean. It has been shown that the Triplet/Quadruplet model can be ported to Korean with promising results (Kanayama et al., 2014). Inspired by this work, we first show that the feature model we developed for Japanese also gives gains when parsing Korean, without any modifications. The Korean treebank we use is very small, and we go on to explore different variants of delexicalised and semi-lexicalised models for cross-lingual parsing, where we show that further gains for parsing Korean are possible when adding Japanese training data.

In this paper we explore the hypotheses that we can improve accuracy of data-driven dependency parsing by:

- Applying feature models that are tailored towards the phrasal unit structure of Japanese
- Directly applying these feature models to a syntactically similar neighbor: in this case Korean
- Supplementing sparse data for a language with that of a more resource rich other language - in this case, Japanese data can supplement Korean

We use MaltParser (Nivre et al., 2006) to explore these hypotheses.

2. Japanese and Korean Phrasal Units

Japanese grammar is often analysed in terms of *bunsetsu* (文節): discrete, nested phrasal units (PUs) consisting of a nuclear content word (for example a verb, noun or adjective) and a set of function words/morphemes. In a preterminal PU, the last morpheme is often a postpositional particle, encoding what the PU modifies. Although the order of modifying PUs may be very free, the direction of modification is almost invariably left-to-right, and - since Japanese is a head final language - the root word will usually be in the last PU. A large part of parsing Japanese is the task of determining the modifying relations of these phrasal units.

Like in Japanese, Korean phrasal units *(eojeol)* are often bounded by particles which encode information about which PU it is modifying. Many of these map, at least roughly, to a Japanese equivalent, though there are some significant differences in usage which make them not quite interchangeable.

3. The Triplet/Quadruple Model

The Triplet/Quadruplet (TQ) model (Kanayama et al., 2000) is a hybrid parsing method for dependency parsing in Japanese, which makes use of both statistical modeling and a hand-crafted grammar to compare trees in a forest. Under the TQ model, the parser takes a *modifier* phrasal unit, and compares up to three *modification candidate* PUs; the model is named for its statistical methods for when there are two (*Triplet*) or three (*Quadruplet*) modification candidates. The technique makes use of the features of the phrasal unit as a whole, including the head word and its dependents, such as morphemes, particles and punctuation.

Kanayama et al. (2014) also applied the model cross-lingually, with Korean as a target low resource language drawing from Japanese as a resource rich language. The adaptation required new grammar rules to be written for Korean, and the feature model to be modified. Features from Japanese were transferred to the Korean parser, and the same process of comparing modification of up to three modification candidates was used. It was found that, at low resource settings, a cross-lingual parser with Korean as the target language would benefit by as much as 1% UAS when supplemented with Japanese data, though this effect would diminish and in fact reverse as the Korean corpus grew in size.

Our model differs somewhat from this. Instead of graph comparison, we simply use a linear time transition-based

	Features
1	The POS tag of the modifier
2	The form of the modifier
3	The dependency relation of the rightmost dependent of the modifier
4	The POS tag of the rightmost dependent of the modier
5	The form of the rightmost dependent of the modifier
6	The POS tag of the modification candidate
7	The form of the modification candidate
8	The dependency relation of the rightmost dependent of the modification candidate
9	The POS tag of the rightmost dependent of the modification candidate
10	The form of the rightmost dependent of the modification candidate
11	Conjunction 1 * 6
12	Conjunction 2 * 7
13	Conjunction 3 * 8
14	Conjunction 2 * 6
15	Conjunction 5 * 10
16	A distance metric discretised as 0,1,2,5,10

Table 1: Key features of the new feature model

model. However, we retain the phrasal unit focus by combining features of modifier and modification candidate PUs to inform the decisions of the parser. We also apply a similar mapping of particles and adpositions between Japanese and Korean in order to exploit the two languages' similarity in particles and adpositions.

4. Experiments

In all experiments we used MaltParser (Nivre et al., 2006), a data-driven transition-based dependency parser. We used data from the collection of Universal Dependencies corpora, version 2.1 (Nivre et al., 2017). For Japanese we used the large GSD corpus, with a total of 8232 sentences, using the default partitions for training, development and test. For Korean, there are several corpora available of different sizes. Unfortunately the two largest corpora did not have tokenisation schemes that were consistent with Japanese. Whereas in Japanese, particles and adpositions are counted as separate tokens, in both the GSD and Kaist Korean corpora these are adjoined to the content words. We therefore choose to use the smaller PUD corpus, with a total of 1000 sentences, which has a tokenisation scheme consistent with Japanese. We used 500 sentences for training and 500 sentences for testing. We thus treat Korean as a low-resource language in our experiments. In all our experiments we use the gold part-of-speech tags available in these corpora.

In a set of initial experiments we investigated which of the parsing models in MaltParser performed best on Japanese development data. We chose the stack lazy algorithm, which had the best average LAS and UAS score. This algorithm adds dependency relations between the two top elements of the stack and includes a swap transition, which allows parsing of non-projective structures (Nivre, 2009).

In the following we start by reporting results on Japanese parsing. Then we describe how the models developed can be applied also to Korean.

4.1 Japanese Feature Modeling

The feature modeling in this study draws on the phrasal unit structure explained above, and insights from the TQ model (Kanayama et al., 2000). We do not need to write any rules, but can add features reflecting the phrasal unit structure to the feature model used in MaltParser. The new model is designed to treat PUs, looking at the form and POS tag of the relation candidates ($stack_0$ and $stack_1$), their rightmost dependents (as this is where key function words and particles will appear), and conjunctions of these features. The intuition is that, for example, if the first phrasal unit has a subject marker at the end, and the next an object marker at the end, the parser will avoid drawing an arc between these two. The features used are shown in Table 1, and are added to the default features of MaltParser's StackLazy feature model.

Feature model	LAS	UAS
Original features	72.7	83.0
New features	80.7	86.8

Table 2: LAS and UAS scores of the new feature model on the Japanese dev set

Table 2 shows the results with the new feature model for the Japanese development set. As expected, the new feature model achieved a substantial improvement in both LAS and UAS, compared with the out-of-the-box feature model of the Stack Lazy algorithm. Notably, there is more improvement in LAS than UAS, indicating that the new model is more beneficial for determining labeled relations than for syntactic structure.

4.2 Japanese Delexicalisation

Any cross-lingual application of this model would require some amount of delexicalisation, i.e. not to use the actual word forms, but only use the part-of-speech tags. To gauge the effect of delexicalisation on the parser's performance, we compared three delexicalisation settings:

- The default, fully lexicalised data
- Fully delexicalised data (full delex)
- Delexicalisation except for particles and adpositions (PART/ADP only)

The PART/ADP only setting is based on the intuition that phrasal unit ending particles are particularly important for determining the global syntactic structure of a sen-

Lexical model	LAS		UAS	
	Dev	Test	Dev	Test
Lexicalised	80.7	83.7	86.8	88.8
Full delex	74.0	74.7	85.4	84.8
PART/ADP only	84.0	84.9	88.6	89.0
Xling	81.6	82.6	88.5	88.4

Table 3: Performance of lexical models on Japanese test set

tence, as they encode the information of which proceeding phrasal unit should be modified by the current one. Thus in this setting, they are represented by the actual word forms, whereas all other words are represented by the POS tags. Moreover, as detailed previously, the similarities between some Japanese and Korean particles means that some information encoded by Japanese particles may be transferred cross-lingually to Korean, if such information is particularly useful.

The results of the settings are shown in the Dev columns of the three top rows of Table 3. Full delexicalisation weakens performance, particularly on LAS, which is 6.7 points below the baseline. However, removing all lexical information *except* for particles and adpositions improves performance above the baseline - by 1.8 UAS and 3.3 LAS points. This suggests that, while some delexicalisation can benefit the parser, particles and adpositions encode valuable information about dependency relations and sentence structure.

4.3 Cross-Lingual Delexicalisation

Theoretically, the correspondence between Japanese and Korean particles should make it easy to adapt the PART/ADP only delexicaliser to work on both Japanese and Korean by mapping Japanese particles to a concatenation of themselves and their rough Korean equivalents, and vice versa. For example, mapping を (*'wo'*, the Japanese subject marker) to を|을|를 (*'wo|eun|neun'*; 을 *'eun'* and 를 *'neun'* are variations of the same particle). This would mean that, with some preprocessing, the parser could delexicalise the Japanese and Korean corpora while retaining the syntactic information contained in the particles in each language.

There are a few problems with this. The first is that there are instances of polysemy in some particles in both languages. For example, the Japanese particle が (*ga*) can be either a subordinate subject marker or a contrasting conjunction corresponding to the English 'but'; these would map to different Korean particles.

A second, more intractable problem is that some particles are used very differently between the two languages. For example, the Korean particle 에서 (*eseo*) means 'from', but it can also mean 'in' (e.g. "She is studying *in/from* her room"), so long as the verb is not one of direction. This is not so easily overcome by simple rules; the simplest mitigation is to avoid such instances.

For this reason, only the most essential and frequent particles are treated here: for example the subject, topic and object markers, the possessive particle, and constrasting and coordinating conjunctions. The subject marker - が (*ga*) in Japanese and 가/이 (*ga/i*) in Korean - is concatenated with the contrasting conjunction - が in Japanese and 만 (*man*) in Korean. This is not optimal, since both are important syntactically, but the subject marking particle is syntactically central to the construction of most sentences, so it is too important to leave out. The result of this concatenation is that, for example, when the particle 'は' appears in the Japanese corpus, in the delexicalisation process it will be replaced with 'は|은|는'; likewise where '은' appears in the Korean corpus, it will also be replaced with 'は|은|는'. In this way, both corpora are compatible when delexicalised, with all lexical items converted to POS tags *except for* the cross-lingual particles, which are converted to a concatenation of themselves and all equivalent particles. The concatenation scheme is shown in Figure 4. We refer to the resulting cross-lingual delexicalisation scheme as 'Xling'.

When applied to the Japanese dev set, this cross-lingual delexicalisation model (Xling) performs slightly below PART/ADP on UAS, but still above both the lex and full delex baselines, as shown in Table 3 (Dev). We tested the performance of each of the delexicalisation schemes on the GSD corpus test set. The results can be seen in Table 3 (Test). The three delexicalised models had fairly similar results on the test, suggesting that the performance generalises well to new data.

5. Cross-lingual Application

In this section we explore how well the feature models developed for Japanese works for Korean, and explore cross-lingual learning.

5.1 Application of the New Feature Model to Korean

Table 5 shows that the new feature model also outperformed the out-of-the-box baseline considerably for Korean, supporting the hypothesis that Japanese features can be applied successfully to Korean.

Full delexicalisation in Korean produced the lowest results, with PART/ADP only and Xling getting very similar scores in both UAS and LAS. In contrast to Japanese, none of the delexicalisation settings outperformed the fully lexicalised baseline. This was counter to our intuition that a smaller size would benefit more from delexicalisation than a larger one, and needs to be investigated further in future work.

5.2 Cross-Lingual Supplementation

The final experiment was to see if supplementing the sparse Korean data with Japanese data would improve accuracy. Based on the intuition that adding too much Japanese data would skew the parser towards Japanese, we experimented with adding different amounts of Japanese data to the training set. The results are shown in Table 6. In all cases the Xling model outperformed full delexicalisation, notably by a larger margin than for a similar experiment with the Triplet/Quadruplet model in (Kanayama et al., 2014). When at least a 1:1 ratio of Japanese data was used, the Xling model outperformed the Korean only baseline. While adding Japanese data to the Xling Korean parser did improve performance, this plateaued completely after 1:1 ratio. It would therefore seem that adding data from a more resource rich language can improve performance, but that the amount is not important after 1:1 is reached.

particle/ adposition	は	은	는	が	가	이	만	を	을	를	の	의	と	와	과	まで	까지
concatenation	は|은|는			が|가|이|만				を|을|를			の|의		と|와|과			まで|까지	
syntactic function	topic marker			subject marker OR 'but'				object marker			possessive particle		'and'			'until'	

Table 4: Concatenation scheme for Japanese and Korean particles

Features	**Lexicalisation**	**LAS**	**UAS**
Default	Lexicalized	78.7	86.3
New	Lexicalised	88.3	90.5
New	Full delex	82.7	87.2
New	PART/ADP only	85.9	89.1
New	Xling	85.5	88.8

Table 5: Performance of lexical models on Korean

Ratio ja:kr	***Lexicalisation***	**LAS**	**UAS**
0:1	*Full delex*	82.7	87.2
	Xling	85.5	88.8
0.5:1	*Full delex*	82.7	87.1
	Xling	85.5	88.8
1:1	*Full delex*	86.9	90.3
	Xling	**90.0**	**92.3**
2:1	*Full delex*	86.9	90.4
	Xling	89.9	92.2
10:1	*Full delex*	86.9	90.4
	Xling	89.9	92.2
Baseline	*Lexicalised*	88.3	90.5

Table 6: Performance on Korean with Japanese data added, where ratio 0:1 means no Japanese data, and the baseline is a Korean only fully lexicalised model.

6. Discussion

The experiments with Korean worked with a small data set, which could have influenced the conclusions. In Kanayama et al. (2014) the influence of adding Japanese data to Korean is reduced when more Korean training data is added. Another influencing factor was that we used gold POS tags. We leave for future work the investigation of the effect of imperfect predicted tags. It would also be interesting to see if we can extend state-of-the-art neural parsers with these language specific insights, as we could in MaltParser.

It is interesting that the greatest differences in performance between different parsers were in LAS, rather than UAS. The focus of the work of Kanayama et al. (2014) was on UAS, as this would indicate differences in attachment, rather than labeling. This may suggest that, under this model, the particle information is being used more to determine syntactic relation types, rather than syntactic relations themselves.

An advantage of our strategy is that very little manual work is needed, compared to implementing the Triplet/Quadruplet model for two languages (Kanayama et al., 2000; Kanayama et al., 2014). We only needed to specify one feature model and a mapping of particles, compared to needing manual rules for two languages, besides this.

7. Conclusion

In this study we have shown that we can use insights from the mainly rule-based Triplet/Quadruplet model (Kanayama et al., 2000) to extend a feature model for the data-driven MaltParser, in order to improve parsing both for the targeted language Japanese, and for the structurally similar language Korean. Further, we showed that using a suitable level of delexicalisation, we could get further improvement to Korean parsing by adding Japanese data during training.

In all, the study demonstrates the effectiveness of language specific adaptation for data-driven dependency parsing, both monolingually and cross-lingually. We hope that this principle might be extended to other pairs or sets of syntactically similar languages with beneficial results.

References

Hiroshi Kanayama, Kentaro Torisawa, Yutaka Mitsuishi, and Jun'ichi Tsujii. 2000. A hybrid Japanese parser with hand-crafted grammar and statistics. In *Proceedings of The 18th International Conference on Computational Linguistics; Volume 1*, pages 411–417, Saarbrücken, Germany.

Hiroshi Kanayama, Youngja Park, Yuta Tsuboi, and Dongmook Yi. 2014. Learning from a neighbor: Adapting a Japanese parser for Korean through feature transfer learning. In *Proceedings of the EMNLP'2014 Workshop on Language Technology for Closely Related Languages and Language Variants*, pages 2–12, Doha, Qatar.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 2216–2219, Genoa, Italy.

Joakim Nivre, Željko Agić, Lars Ahrenberg, et al. 2017. Universal dependencies 2.1. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Joakim Nivre. 2009. Non-projective dependency parsing in expected linear time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 351–359, Suntec, Singapore.

Daniel Zeman, Filip Ginter, Jan Hajič, Joakim Nivre, Martin Popel, and Milan Straka. 2018. CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Brussels, Belgium.

Identifying Source Words of Lexical Blends in Swedish

Adam Ek

Department of Linguistics
Stockholm University
SE-106 91 Stockholm
adam.ek@ling.su.se

Abstract

This paper provides a method for identifying source words of lexical blends in Swedish. The method (except for the resources used) is language independent. To predict the source words of lexical blends, all possible word combinations that create the blend are extracted from the SALDO lexicon and associated with a set of features. The method was evaluated through two experiments, rankings, and feature ablation, both using cross-validation. The results show that for the top ranking samples, the correct candidate pair is found in 32.2% of the cases. For the top 10 ranking samples, the correct candidate pair was found in 60.6% of the cases. The feature ablation reveals that embedding similarity and frequency are the most important features.

1. Introduction

A productive word formation process in Swedish is compounding, which is the concatenation of two or more words. The analysis of compounds in Swedish Natural Language Processing (NLP) is essential given the productiveness of the word formation process (Sjöbergh and Kann, 2004). Lexical blending is a word formation process that alike compounding concatenates two or more words but also reduces one or more of the words (Mattiello, 2013). What separates blending from compounding is that one of the source words is reduced in a lexical blend.

An example of a lexical blend is *motell* ('motel') which is the concatenation of the reduced forms of *motor* ('motor') and *hotell* ('hotel').

(1) motell = m**ot**[or] + [h]**ot**ell[1]

Example (1) show a blend where the characters in the source words overlap, e.g. `ot` occur in both source words and in the blend, but blends without overlap also exist. A blend without overlap is *alfanummer* ('alphanumber') that is created from *alfabet* ('alphabet') and *telefonnummer* ('telephone number'):

(2) alfanummer = alfa[bet] + [telefon]nummer

The blend in example (2) is created by removing the first word in a compound (*telefon* in *telefon-nummer*) and replacing it with the reduced form of *alfabet*.

Lexical blends and compounds in Swedish are generally rare, and thus are unlikely to appear in any lexicon. This poses a problem for applications relying on lexical information regarding words. A common approach to this problem for compounds is to identify the words used to create the compound and derive the required information from these words. This paper approaches lexical blends in the same manner. Given a model that is able to predict the source words of lexical blends, depending on the external application's purpose, the required information may be derived from the source words.

[1]Bold indicates overlapping letters and brackets indicates parts of the words which are removed.

2. Previous studies

Lexical blends have been left almost untouched in computational linguistics. Two research papers have targeted the identification of source words: (Cook and Stevenson, 2007) and (Cook and Stevenson, 2010). Corpus studies have been more numerous, where the relationship between the source words and the lexical blends have been investigated in (Gries, 2004a), (Gries, 2004b) and (Gries, 2012).

In (Cook and Stevenson, 2010), the authors further develop the method in (Cook and Stevenson, 2007), as such only the latter will be reviewed. A dataset of 324 lexical blends was gathered from previous research articles and from the website `www.wordspy.com`. To identify possible word pair candidates the lexical blend was partitioned into n splits, each split being at least two letters long. For example, *motell* is split into: *((mo,tell), (mot, ell), (mote,ll))*. For each blend split, each word pair in the lexicon where the first word has the same letters as the first part of the split and the second word has the same letters as the second part. In addition to generating candidates in this manner, a syllable splitting strategy was also implemented.

For each candidate word pair, a set of features were extracted. The features used are constructed in such a way that the values should be higher for the correct word pairs, and lower for incorrect word pairs. The features capture the raw, relative, and n-gram frequency of the source words, character contribution to the lexical blend from the source words and ontological/distributional similarity between the source words.

Two types of algorithms were tested, a statistical model and a perceptron algorithm. The statistical model estimates a score for each candidate pair by calculating a normalized score for each feature i by dividing the mean of the feature by its standard deviation. As a final operation, the arctan of the normalized feature score is calculated. A mathematical description of the model is given in Equation (1).

$$score(sw1, sw2) = \sum_{i=0}^{len(f)} arctan(\frac{mean(f_i)}{sd(f_i)}) \quad (1)$$

The performance was evaluated by ranking each candidate pair according to the sum of its feature vector. Both models had the same performance with an accuracy of 40% for the highest scoring word pair, compared to a random baseline and an informed baseline, achieving an accuracy of 6% and 27% respectively.

3. Data

This section presents the resources used and the dataset of lexical blends.

3.1 SALDO

The SALDO lexicon has been used as a resource to extract candidate source words from. SALDO is a semantic and morphological lexicon containing Swedish lemmas (Borin et al., 2008).

3.2 Corpora

A corpus was compiled from two news corpora, Webbnyheter (Webnews) and GP (Göteborgs Posten), between the years 2002 and 2012. The corpora are available at Språkbanken[2]. The corpus was used to measure the frequency of the words in SALDO and to create word embeddings.

3.3 Models

Both word and character embeddings are used as features. The word embeddings model is created from the corpus described in the previous section, using CBOW (Continuous Bag-of-Words) contexts and a minimum frequency of 1. The character embedding model was constructed in (Bojanowski et al., 2016) from the Common Crawl corpus and the Swedish Wikipedia. The model uses CBOW contexts, n-grams between sizes 2 and 5 and negative sampling set at 10. Both models use a window of 5 words.

3.4 Lexical blends

A total of 223 lexical blends were manually identified by the author from the following sources: (a) Nyordslistan[3], (b) Kiddish[4], (c) Slangopedia[5], (d) Språktidningen[6] and (e) personal correspondence. Two criteria were followed when selecting blends: (1) only blends that have two source words are selected and (2) only blends where the beginning part of a word is combined with the ending part of another word are selected. For example, blends where one word is inserted into another word are ignored (e.g. *Samargbete* = *sam**ar**bete* ('cooperation') + ***ar**g* ('angry')) and source words which are combined by using the beginning part of both words (e.g. *Fakus* = *fa[r]* ('father') + *kus[in]* ('cousin')) are ignored.

From the set of 223 lexical blends, 158 have both source words in the SALDO lexicon. The lexical blends without both source words in the SALDO lexicon were discarded.

[2] `https://spraakbanken.gu.se/`

[3] `https://www.sprakochfolkminnen.se/sprak/nyord/nyordslistor.html`

[4] `http://www.kidish.se/`

[5] `http://www.slangopedia.se/`

[6] `http://spraktidningen.se/`

The remaining dataset was split into two datasets, one containing blends with an overlap between the characters in the source words, and one containing blends where the characters do not overlap. Examples of overlapping and noverlapping blends can be seen in Table 1. In total, there are 63 overlapping blends and 95 noverlapping[7] blends.

Table 1: Examples of overlapping and noverlapping blends. OVL = Overlapping, NVL = Noverlapping.

TYPE	BLEND	SOURCE WORDS
OVL	Blorange	bl**o**[nd] ('blonde') + **o**range ('orange')
OVL	Chattityd	ch**att** ('chat') + **att**ityd ('attitude')
NVL	Mizeria	mis[är] ('misery') + [piz]zeria ('pizzeria')
NVL	Promelur	prome[nad] ('stroll') + [tupp]lur ('nap')

4. Method

This section presents the system architecture, how candidate word pairs were generated and the set of features used.

4.1 System architecture

The system was implemented in Python 3, using the libraries `sklearn` for a logistic regression implementation, `epitran` (Mortensen et al., 2018) to translate orthographic words to IPA-symbols and `pyphonetics` to measure the Levenshtein distance between phonetic representations.

4.2 Candidate selection

Candidate word pairs are generated by first splitting the blend into its two beginning and ending characters. All the words in the SALDO lexicon with the identical beginning characters are put into a prefix set, and all words with the identical ending characters are put into a suffix set. To generate candidate pairs, the product of the two sets is computed.

To select acceptable candidate pairs for the overlapping blends, all word pairs which can be combined in two or more ways are selected. E.g. the candidate pair (*bror*, *vokabulär*) can be combined into the blend *brokabulär* in two ways: `bro` + `kabulär` and `br` + `okabulär`. For noverlapping blends, only candidate pairs which can be combined into the blend in one way are selected, e.g. *fri* and *semester* can form *frimester* only by `fri` + `mester`.

4.3 Features

The features used by the model are described below.

Embedding score (1-3, 7-9): The embedding score features are calculated by taking the sum of the word and character embedding vector for the source words and the lexical blend.

Embedding similarity (4-6, 10-12): The feature captures the cosine similarity from the character and word embedding models between the source words, and between the source words and the lexical blend.

[7] *Noverlap* is a blend coined by the author. It is the blending of 'no' and 'overlap' and denotes lexical blends which do not overlap.

Bi- and trigram similarity (13-16): The bi- and trigram similarity are captured by counting the number of shared bi- and trigrams between the two source words.
Longest common substring (17): The longest common substring between the two source words is calculated as a feature.
Levenshtein distance (18-24): Two types of Levenshtein distances are measured: orthographic and phonetic. The Levenshtein distance is measured between the two source words, and between the source words and the lexical blend. The words were translated from text to IPA with the `epitran` package (Mortensen et al., 2018), and the phonetic Levenshtein distance was calculated using `pyphonetics`.
Phonemes (24-25): The number of phonemes in the first and second source word in relation to the number of phonemes in the lexical blend is calculated.
Syllables (26-27): The number of syllables in the first and second source word in relation to the number of syllables in the lexical blend is calculated. To calculate the number of syllables, the number of vowels in the words were used.
Word length (28-29): The number of characters in the source word is counted relative to the number of characters in the lexical blend.
Contribution (30-31): The contribution of each source word to the lexical blend is calculated by dividing the number of characters contributed by each source word by the total number of characters in the blend.
Removal (32): This feature takes the sum of lengths from the source words divided by the length of the lexical blend. This feature measures how much of the source words combined is removed to create the blend.
Source word splits (33): The number of ways to split the candidate pair to create the lexical blend.
Affix frequency (35, 37): The affix frequency is calculated by finding the correct blend split and calculating the frequency of the source word relative to the total frequency of all words with the same beginning/end string as the split.
Corpus frequency (34, 36): For each source word, its frequency relative to the corpus is captured.

4.4 Experimental setup

The model is evaluated through two experiments: a ranking experiment and a feature ablation experiment. Both experiments are performed using cross-validation. For overlapping blends 6 folds are used, for noverlapping blends 9 folds are used and for the combined dataset 10 folds are used. Development was performed on the first fold of the overlapping blends. The ranking experiment measures if any correct word pair is found in the top n ranking word pairs where $n = \{1, 3, 5, 10\}$. The ranking is determined by the probability that a word pair belong to the true class. The model is compared against two baselines, the first baseline selects n word pairs at random and the second baseline implements the model used by Cook and Stevenson (2010) as described in Equation (1).

A feature ablation experiment is performed where groups of features are removed. To measure the performance change of overlapping blends MAP (Mean Average Precision) is used and MRR (Mean Reciprocal Rank) is used for noverlapping blends[8].

5. Results

This section presents the results for the ranking and feature ablation experiments.

5.1 Ranking

The results from the ranking experiment are shown in Table 2. The experiment is performed on the overlapping, noverlapping blends and for the datasets combined.

Table 2: Model evaluation of all lexical blends and comparison to the baselines. The evaluation is performed by considering the system to be correct if the top n ranking word pairs contain a correct word pair.

SYSTEM	ACC_1	ACC_3	ACC_5	ACC_{10}
OVERLAP				
Random	0.031	0.063	0.126	0.158
Feature ranking	0.190	0.349	0.365	0.428
Logistic Regression	0.444	0.611	0.666	0.740
NOVERLAP				
Random	0.021	0.052	0.063	0.094
Feature ranking	0.021	0.063	0.115	0.168
Logistic Regression	0.234	0.416	0.437	0.541
ALL				
Random	0.031	0.044	0.088	0.107
Feature ranking	0.069	0.145	0.196	0.240
Logistic Regression	0.322	0.492	0.537	0.606

5.2 Feature ablation

The feature ablation is performed by removing groups of features, e.g. all features that measure the similarities between word embeddings are removed together. The results are shown in Table 3.

6. Discussion

This section discusses the results from the ranking and feature ablation experiments.

6.1 Ranking

The performance of the random baseline on all datasets is low for all thresholds, ranging from 2% to 15.8%. The feature ranking performance is only slightly higher than the random baseline for the noverlapping blends and complete dataset with an accuracy range from 2% to 24%. For overlapping blends, the feature ranking baseline performs much better with an accuracy between 19% and 42.8%.

In comparison to the baselines, the logistic regression has better performance. The accuracy ranges from 23.4% to 74%, where the best results are obtained for the overlapping blends. The performance on the noverlapping blends is the lowest, while the performance of the complete dataset is in-between that of overlapping and noverlapping blends.

[8]MAP = MRR if the number of correct word pairs is one, which is the case for the noverlapping blends.

Table 3: Feature ablation experiments with groups of features removed. OVL = Overlapping blends, NVL = Noverlapping blends, ALL = Overlapping and noverlapping blends combined.

	OVL	NVL	ALL
FEATURE GROUP	MAP	MRR	MAP
All features	48.6	34.7	40.5
Character score	+1.5	−1.6	±0.0
Character similarity	−5.4	−5.1	−4.3
Word score	+0.3	−1.3	+0.3
Word similarity	−4.4	−5.4	−3.2
Bigram similarity	−0.7	±0.0	−0.2
Trigram similarity	+0.2	−0.5	+0.3
IPA Levenshtein distance	+1.8	−1.7	+0.1
Levenshtein distance	+2.5	−0.5	+0.6
Phonemes	−0.5	−0.6	−0.2
Syllables	−1.6	+0.1	+0.7
Length	±0.0	−0.5	+0.4
Contribution	±0.0	+0.4	+0.1
Removal	−0.1	−0.1	+0.5
Splits	−1.2	+0.1	−0.7
Corpus frequency	±0.0	±0.0	±0.0
Affix frequency	−2.7	−8.7	−5.4

It can be observed that the noverlapping dataset is most difficult for all methods. In part, this is caused by the large number of possible candidates for each blend[9].

Comparing the results to (Cook and Stevenson, 2010), who evaluated on the top ranking sample, the performance of the current system on the combined dataset is 7.8 percentage points lower. The lower performance is not unexpected as the dataset used in (Cook and Stevenson, 2010) is roughly twice the size of the current dataset.

6.2 Feature ablation

The feature ablation shows that three groups of features produce large performance changes compared to the other groups. These groups are character similarity, word similarity, and affix frequency.

There is a slight difference between the impact of character and word similarity, where the average performance loss is higher for character similarity. This may in part be because character embeddings are able to produce a vector for the lexical blend, which word embeddings usually cannot. This allows the character embeddings model to measure the similarity between the source words and the blend in addition to the similarity between the source words.

The affix frequency feature showed a high performance loss, especially for the noverlapping blends. This shows that the frequency of the source words is important, and primarily when compared with other possible candidates since the corpus frequency feature does not have any impact. The affix frequency feature is the only feature which considers the relationship between the current candidate pair and the other possible candidate pairs. Constructing additional features with the same principle would be beneficial given that the number of candidate pairs may be quite large.

The orthographic and phonetic features that capture the relationship between the source words and the blends show small changes. This indicates that the features as they currently are realized do not seem to capture the orthography or phonology in a meaningful way.

[9] In some cases, there are up to 500 000 incorrect word pairs and one correct pair.

7. Conclusions and future work

The results are promising given the small dataset. It is encouraging that the only language-dependent features are the resources, as such the method should also be applicable to English (where a much larger dataset of lexical blends can be found).

Improvements to the model will focus on re-thinking the orthographic and phonetic features used and incorporating more features that take into account the other possible candidates and finding the more prominent candidates.

The candidate selection will also be refined, with finding a more efficient and accurate method of selecting word pairs, for example by splitting the blend based on syllables and/or morphemes.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Lars Borin, Markus Forsberg, and Lennart Lönngren. 2008. Saldo 1.0 (svenskt associationslexikon version 2). *Språkbanken, University of Gothenburg*.

Paul Cook and Suzanne Stevenson. 2007. Automagically inferring the source words of lexical blends. In *Proceedings of the Tenth Conference of the Pacific Association for Computational Linguistics (PACLING-2007)*, pages 289–297.

Paul Cook and Suzanne Stevenson. 2010. Automatically identifying the source words of lexical blends in english. *Computational Linguistics*, 36(1):129–149.

Stefan Th Gries. 2004a. Isnt that fantabulous? how similarity motivates intentional morphological blends in english. *Language, culture, and mind*, pages 415–428.

Stefan Th Gries. 2004b. Shouldn't it be breakfunch? a quantitative analysis of blend structure in english. *Linguistics*, pages 639–668.

Stefan Th Gries. 2012. Quantitative corpus data on blend formation: Psycho-and cognitive-linguistic perspectives stefan th. gries. *Cross-disciplinary perspectives on lexical blending*, 252:145.

Elisa Mattiello. 2013. *Extra-grammatical morphology in English: abbreviations, blends, reduplicatives, and related phenomena*, volume 82. Walter de Gruyter.

David R. Mortensen, Siddharth Dalmia, and Patrick Littell. 2018. Epitran: Precision G2P for many languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), May.

Jonas Sjöbergh and Viggo Kann. 2004. Finding the correct interpretation of swedish compounds, a statistical approach. In *In Proc. 4th Int. Conf. Language Resources and Evaluation (LREC), pages 899–902.*

Annotation of learner corpora: first SweLL insights

Elena Volodina[1], Lena Granstedt[2], Beáta Megyesi[3], Julia Prentice[1], Dan Rosén[1], Carl-Johan Schenström[1], Gunlög Sundberg[4], Mats Wirén[4]

[1]University of Gothenburg (Sweden), [2]Umeå University (Sweden)
[3]Uppsala University (Sweden), [4]Stockholm University (Sweden)
elena.volodina@gu.se

1. Introduction

SweLL - Swedish Learner Language - is a project aimed at setting up an electronic infrastructure for collecting, annotating, browsing and analyzing Swedish learner language (Volodina et al., 2016). During the first year of the project, a number of the project aims have been addressed, such as

1. legal and ethical aspects of essay collection
2. principles of learner language annotation
3. tools and platforms for securing the previous steps

As the practice shows, annotation of learner texts is a very sensitive process demanding a lot of compromises between ethical and legal demands on the one hand, and research and technical demands, on the other. Below, is a concise description of the current status of the SweLL project with numerous evidence of the above-mentioned compromises[1].

2. Legal issues and their consequences

Spreading an electronic resource through an infrastructure entails responsibility to the data subjects, in our case language learners, who have agreed to provide their texts and personal information. The requirement of collecting and storing informed consents, obligation to remove a learner and their data from the registers if they desire so as well as national and international laws and ethical regulations regarding personal integrity and discrimination create certain difficulties in making the data open for all types of uses. To argue for the data to be accessible to users outside individual projects, handling of data should be 'bulletproof' at each stage and there are several stages to consider, namely, data acquisition, data storage, data aggregation, data analysis, data usage, data sharing and data disposal (Accenture, 2016). Most of the steps deal with organizational and management decisions/precautions or preparatory steps before uploading data to the infrastructure. In the text below, we concentrate on the stages relevant to infrastructure usage where learner specific characteristics in the texts and metadata present risks at the data usage and data sharing stages.

To start with, within European countries, there is a requirement to ensure personal non-identifiability when adding essay information with personal metadata. According to the EU General Data Protection Regulation (GDPR), Article 4[2], "personal data means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person..." (Commission, 2016, art.4). Consider Figure 1, where adding up information from the two sources – a learner text and socio-demographic metadata – can give away a learner. Even though the name as such is not revealed to the data users, indirect clues can be used to identify a person.

SOCIO-DEMOGRAPHIC METADATA

- L1: Luxembourgian, Chinese
- Year of birth: 1986
- Gender: male
- Education / highest degree: MA
- Time in L2 country: 3 years
- Other languages: Russian, Korean, German, French

TASK METADATA:

- Date: April 2018
- CEFR level: B1

TEXT:
I lived in Denmark before, in Svaneke. It was less thenn Berlin. I like there too because I had more friends. But I have better work here. In Svaneke job was on one webpage. In Berlin I work on many webpages. I am web develooper. But Berlin is closer to Louxembourg that Svaneke.

Figure 1. Example of (selected) metadata and an essay text for a fake learner

In view of this, unlike a number of learner corpora projects, the SweLL project adopted a rather restrictive approach to metadata. For instance, it does not provide a student's country of origin or nationality (restricting information to the mother tongue (L1) only), nor the year of birth, but rather a 5-year span (e.g. 1970–1974), to complicate possible identification of a learner through aggregated personal information. For the same reason, no information is provided on the educational establishment where the essays

[1]Parts of Sections 2 and 3 have originally been written by the abstract co-authors for the article by Stemle et al. (2019) and are re-used with the permission of the LCR volume editors

[2]https://gdpr-info.eu/art-4-gdpr/

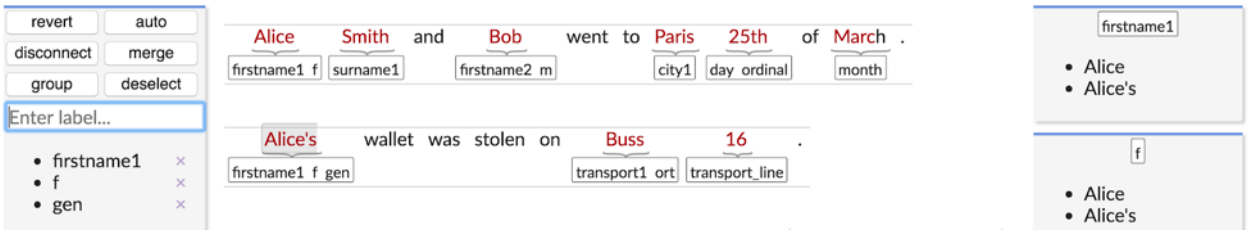


Figure 2: Anonymization compact view in SweLL anonymization tool (Rosén et al., 2018)

have been collected. This comes as a natural consequence of, on the one hand, the national Swedish legislation on open access to public data (Riksdagen, 1949, ch.2), and on the other, the stricter current European legislation on personal data integrity (Commission, 2016).

Ethical Review Boards set further requirements on the so called sensitive data, i.e., data that can reveal a (potentially identifiable) person's sexual orientation, religion, political views or ethnicity, which may lead to discrimination. Unless it can be ensured that the person behind the (meta)data will not be revealed, Ethical Review Boards are entitled to require an application which should list all potential scenarios for data usage, moreover restricting data usage to internal use only, within the project. This in itself is counterproductive since a research infrastructure is aimed at providing electronically available data to researchers outside the project for any potential research questions that cannot be foreseen in advance.

To make learner data less "sensitive" (according to the Ethical Review Boards' definition) as well as to minimize personal identifiability from a text, learner essays need to be anonymized, so that information in the actual text that may give away the author, is either substituted/pseudonymized (e.g. Poland →Greece); made noisy (e.g. Poland →Europe); or completely removed, see text in Figure 1 where a lot of personal information is provided. Whereas suggestions for anonymization of "structured" or "listed" types of personal information (e.g., personal names, city names, telephone numbers, etc.) can be supported through use of automatic methods as adopted from the medical domain (El Emam and Arbuckle, 2013), "unstructured" types of potentially sensitive information (e.g. *We were happy to participate in a demonstration against Erdogan*) will still need to be marked up manually.

In the SweLL project, data is anonymized in two steps – first manually marking up (1) information that directly or indirectly can reveal the author as well as (2) sensitive information about the author, and then rendering the 'placeholders', e.g. 'firstname1' in Figure 2, according to an associated algorithm. Thus, for 'firstname1 f' a female name will be randomly selected from a list of names registered in Sweden. This two-step process potentially opens a possibility to set an essay into different cultural contexts, for example by selecting names and cities from a certain country or part of the world. However, the question of the influence of anonymization on readability, reader attitudes and assessment is still an open one, as well as how it is best to render personal or potentially sensitive information.

To secure a safe environment for anonymization, a special solution has been developed in the SweLL project, called SweLL-kiosk. A SweLL-kiosk is an encrypted environment that protects unauthorized users to get access to the non-anonymized versions of the essays. Kiosks are equipped with a project management system, a database for storing all versions of the files, and a simplified version of SVALA, SweLL annotation tool, containing anonymization functionalities. Essays that have been anonymized, are exported from the kiosk database to Språkbanken's databases.

3. Normalization and error annotation

Annotation of a standard corpus follows a number of steps including tokenization, morphosyntactic tagging, lemmatization and parsing, all of them assuming a standard language. However, a learner corpus includes texts exhibiting deviations from the standard version of the target language for which the tools have been designed. While standard language can be relatively accurately annotated with existing automatic methods, annotating learner language with the same tools is more error-prone due to various (and often overlapping) types of errors, as in e.g. **I has was* (morphology and agreement) or **We wrote down it* (word order).

Automatic tools aimed at standard language can sometimes be applied with more or less satisfactory results even to learner language. Where available, spelling or grammar checking tools providing suggestions can be used to approximate a corrected version of the text. Alternatively (and more commonly), an additional manual step is added, namely *normalization* which means rewriting the original learner text to a grammatically correct target hypothesis (Lüdeling et al., 2005), before applying a standard annotation pipeline. Most projects, further, combine normalization with *error-annotation*, i.e. labelling the type of change that has been applied to the original text. In SweLL, the two steps - normalization and error-annotation - are separated as conceptually independent ones.

3.1 Normalization

Normalization entails interpretation of intentions of the author, which on many occasions is difficult to make. Consider the following example: **jag trivs mycket bor med dem* (Eng. I enjoy live with them) (see Figure 3). Applying the main principle of normalization that *any change to a grammatically correct version should be as minimal as possible*, i.e. THE PRINCIPLE OF MINIMAL CHANGE, the seemingly best way would be to change the original sequence to *Jag trivs mycket bra med dem*, that is, *bor →bra*. However, this

change does not reflect objectively the knowledge of the learner, namely usage of the verb *att bo* versus the adjective *bra*, with *bra* being used correctly by the learner in the other parts of the text. The referenced minimal change does not seem to reflect the semantics that the learner is trying to convey, either. The Second Language Acquisition (SLA) researchers involved in the SweLL project were unanimous about changing this sentence to *Jag trivs mycket med att bo med dem*.

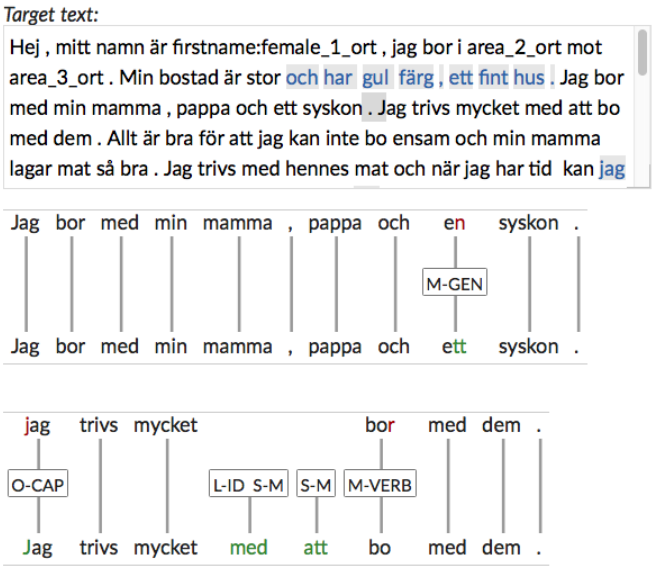


Figure 3: Original and normalized versions of a learner text, with error tags added on the edges. Gloss of the original layer (with some imitation of the errors): *I live with my mother, father and e syster . i enjoy live with them.* The question is, should *live* be changed to *living* or *(my) life*?

Error annotation that is applied to the corrected version is in fact NOT about labeling errors that a learner has made. It rather reflects the difference between the original and normalized versions, and depends upon which normalization variant is accepted. It makes the normalization step extremely important. In the example with the two correction versions of the sentence **jag trivs mycket bor med dem*, error labels could describe either a spelling correction (*bor →bra*) or, as we see in Figure 3, a wrong form of a verb (*bor →bo*) plus idiomaticity problem in using the verb *att trivas* (*trivs →trivs med att*). As such, we cannot claim that we are error-labeling the learner language. We are labeling the type of correction we have introduced.

Several experiments with normalization and error-annotation within the SweLL project have proven that normalization as a separate step is a conceptually right way to go for several reasons:

- It helps to build a better understanding of a learner's linguistic competence (e.g. that (s)he is able to spell the adjective *bra* correctly) so that the changes in the normalized version would take that into account.
- It can be outsourced to SLA researchers for doing it, since (1) normalization takes much less time compared to error-annotation and thus can be done quickly, and (2) SLA researcher reasoning rests on a basis of competence in the SLA field and experience with second language learners, whereas project assistants, who are often L1 students within linguistics, do not have this type of insights into learner language.
- Error annotation depends on the change applied to the original text, and thus should rather start from comparison of the two versions (in contrast to adding error labels at the same time as normalizing a text segment).
- Inter-annotator agreement with respect to error codes can be objectively measured only given that the annotators are working on the same normalized version.

3.2 Error annotation

We start this section with an anonymous quotation: "Taxonomies are like underwear; everyone needs them, but no one wants someone else's." With respect to error annotation projects, this is both true and false. Even though so far very few learner corpus projects have managed to reuse each other's error taxonomies, several projects have tried to build on previous work. Let us demonstrate the problems of re-using someone else's taxonomy with an example from the SweLL project.

Since the SweLL project is in an early stage, there is a direct incentive to learn from the experience of other projects to ensure a certain degree of comparability. In this respect, the SweLL project has looked into some error annotation taxonomies, namely of ASK (Tenfjord et al., 2006) and MERLIN (Boyd et al., 2014).

The initial SweLL tagset was a result of testing the ASK taxonomy (23 tags) and the MERLIN taxonomy (64 tags) on a set of Swedish essays. It turned out that annotating with the highly intricate MERLIN taxonomy took twice as much time as with the ASK taxonomy, leaving a lot of inter-annotator disagreements. As a result of this experiment, the ASK taxonomy has been adopted with several modifications and was tested in a pilot study with the involved researchers. Once again, practical usage of the taxonomy led the SweLL researchers to important insights with reference to tag names and their coverage. See for example Figure 4, where three annotators agreed on both the segment in need of correction (top row) and on the target hypothesis (second row), but not on the error label (O, INV, OINV describing various types of word order errors). Consequently, both the tag names and the number of tags have been reviewed to avoid ambiguity – leaving very little of the original ASK taxonomy as a result.

The strongest argument for reviewing the ASK taxonomy was the possible drop in annotation quality unless the tagset is reduced or changed, an idea also supported in previous annotation projects (Fort, 2016).

To support normalization and error-annotation in a parallel fashion, a tool SVALA has been developed (Rosén et al., 2018) which is now undergoing an extensive testing in its beta version.

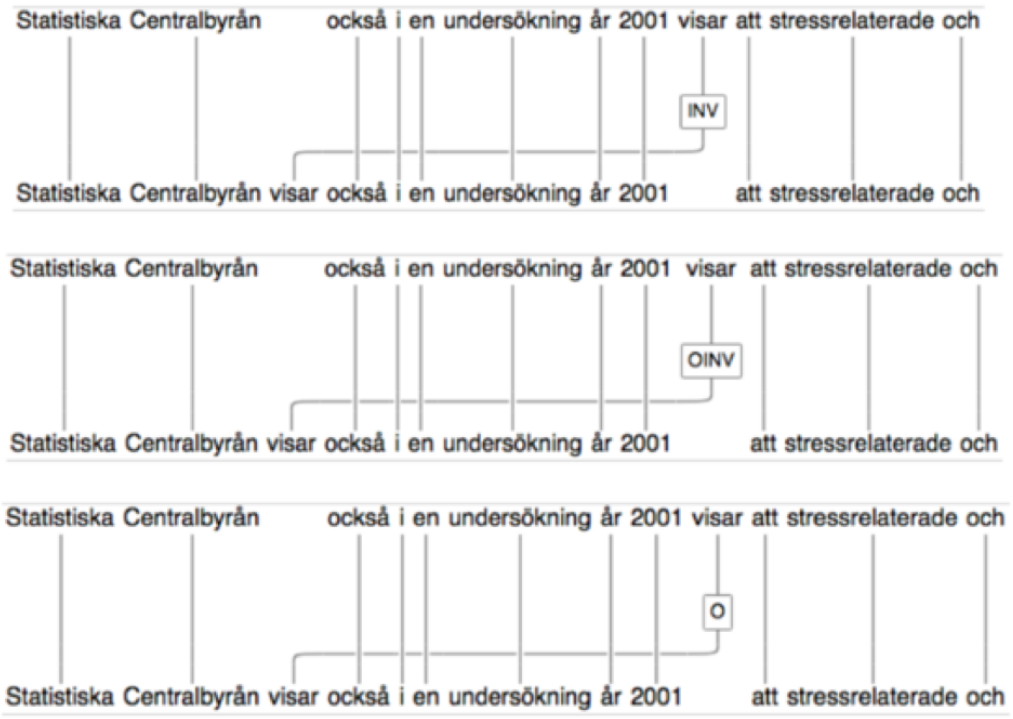


Figure 4: Inspecting error annotation done by three annotators, SweLL error annotation pilot
Gloss: Central Statistical Agency [...] also in a report from 2001 [shows] that stress-related and...
Error code explanations: *INV* Non-application of subject/verb inversion, *OINV* Application of subject/verb inversion in inappropriate contexts, *O* word (or phrase) order error

4. Future prospects

To summarize, the SweLL infrastructure has been extensively developing towards opening a possibility for continuous collection and annotation of learner essays. So far three pilot studies have been carried within the project group, with the aim to produce high quality guidelines, non-ambiguous tag sets and top performing tools. The work is still ongoing. A full scale annotation of essays is planned for 2019.

Next, SweLL will look into the necessary functionalities for visualizing, browsing and statistically analyzing learner corpora - to make learner texts as accessible for SLA research as possible.

References

Accenture. 2016. *Building digital trust: The role of data ethics in the digital age.*

Adriane Boyd, Jirka Hana, Lionel Nicolas, Detmar Meurers, Katrin Wisniewski, Andrea Abel, Karin Schöne, Barbora Stindlová, and Chiara Vettori. 2014. The merlin corpus: Learner language and the cefr. In *LREC*, pages 1281–1288.

European Commission. 2016. *General data protection regulation.* Official Journal of the European Union, 59, 1-88.

Khaled El Emam and Luk Arbuckle. 2013. *Anonymizing health data: case studies and methods to get you started.* " O'Reilly Media, Inc.".

Karën Fort. 2016. *Collaborative Annotation for Reliable Natural Language Processing: Technical and Sociological Aspects*. John Wiley & Sons.

Anke Lüdeling, Maik Walter, Emil Kroymann, and Peter Adolphs. 2005. Multi-level error annotation in learner corpora. *Proceedings of corpus linguistics 2005*, 1:14–17.

Riksdagen. 1949. *Tryckfrihetsförordningen (1949:105).*

Dan Rosén, Mats Wirén, and Elena Volodina. 2018. Error Coding of Second-Language Learner Texts Based on Mostly Automatic Alignment of Parallel Corpora. In *CLARIN Annual conference 2018.*

Egon W. Stemle, Adriane Boyd, Maarten Janssen, Therese Lindström Tiedemann, Nives Mikelić Preradović, Alexandr Rosen, Dan Rosén, and Elena Volodina. 2019. Working together towards an ideal infrastructure for language learner corpora. *Learner Corpus Research 2017, post-conference volume.*

Kari Tenfjord, Paul Meurer, and Knut Hofland. 2006. The ask corpus: A language learner corpus of norwegian as a second language. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 1821–1824.

Elena Volodina, Beata Megyesi, Mats Wirén, Lena Granstedt, Julia Prentice, Monica Reichenberg, and Gunlög Sundberg. 2016. A Friend in Need? Research agenda for electronic Second Language infrastructure. In *Proceedings of SLTC 2016, Umeå, Sweden.*

Finite-State Methods in the Time of Neural Networks

Martin Berglund[⋆], Henrik Björklund[†], Johanna Björklund[†]

[⋆†]Language Processing Center North
[⋆]Universität der Bundeswehr München, [†]Umeå University
martin.berglund@unibw.de, henrikb@cs.umu.se, johanna@cs.umu.se,

Abstract

In this paper we survey the intersection of neural networks and automata theory in contexts where the combination may be useful for natural language processing tasks. This area remains in its infancy despite some of the problems having been considered for several decades, as key techniques have only recently become practical in the wake of the deep learning revolution. Much work has been focused on improving the level to which neural networks can be trained to classify formal languages. Here, however, our primary concern is with isolating some finite state aspects of the process, such as directing the application of neural networks using state machines, and the extraction of state machines from neural networks. Beyond a short survey of some of the literature we offer our commentary on the more promising future directions of research within this area.

1. Introduction

Finite-state methods have long been of fundamental importance for natural language processing (NLP), providing a mathematical backbone of algorithms and representational forms (Roche and Schabes, 1997). In recent years, however, neural networks (NNs) and, in particular, deep neural networks (DNNs) have become increasingly popular, and their performance have surpassed that of automata-based systems for many NLP tasks. Although the history of using NNs to learn languages is long (see for example (Cleeremans et al., 1989)), the approach necessitated the type of high-performance computers that we have today to gain momentum. It is encouraging to see such rapid progress, and to consider the positive impact it may have on our daily lives.

On the negative side, the computational strength of DNNs in general also means that many of the problems that can be stated about them are undecidable. In particular they act as opaque classifiers, where many classes of problems require large structured outputs (e.g. annotating the input) to be satisfactorily answered. Since DNNs are in many aspects resilient to formal analysis, we are frequently forced to resort to empirical methods. The results thus obtained can never be taken as certain, and as the solution space grows the probability of any one result diminishes. For some applications, a high probability is as good as a fact, but when it comes to, for example, aeronautics and medicine, we generally prefer formal proofs. It is therefore undesirable that NLP drifts too far from its linguistic and mathematical origins, and evolves into a purely empirical science.

In this short work, we survey points where automata theory, neural networks, and NLP come into touch. The aim is to lay the groundwork for a discussion of the future of finite-state methods in NLP. We focus in particular on hybrid approaches and finite-state extraction from neural networks, allowing the application of standard techniques and lending structural motivation to the way in which the input is classified.

2. Extracting Automata from Networks

When modelling language with NNs, it is natural to consider recurrent neural networks (RNNs, the special one-dimensional case of recursive neural networks where the recursion forms a unary tree), since they can be used to read variable length sequences. When dealing with regular languages, the second-order recurrent networks we mentioned above are of particular interest (Zeng et al., 1993; Giles et al., 1992). The reason for this is that we can see the hidden neurons in such a network as modelling states. The weight of neuron q given input a and neuron p can be seen as representing the strength of a transition from state p to state q while reading an a. Such networks have been used both to implement DFAs directly and for learning regular languages. For long strings, however, the performance of these networks is often poor, since the representation of automaton states may deteriorate over time (Zeng et al., 1993). For this reason, Giles et al. provide a learning algorithm together with a procedure for extracting an actual DFA from the trained network. They only perform experiments on simple languages that can be represented by automata with a handful of states, but report that the extracted DFA often outperforms the network from which it was derived (Giles et al., 1992).

There is also more recent work in this vein. (Grachev et al., 2017) train recurrent neural network to recognize regular languages. The RNNs considered are extended with an adder function that decides whether the output vector corresponds to an accepting or rejecting state. After training, a finite state automaton can simply be read from the internal tensor representation of the RNN. The approach works well for simple languages, but also suffers from the vanishing gradient problem for more complex languages.

(Weiss et al., 2018) also consider the problem of extracting finite-state automata from RNNs, but they propose the use of the L-star learning algorithm by Angluin (Angluin, 1987), that infers the target language of the network through a series of so-called membership and equivalence queries. The advantage of this approach is that it is largely agnostic about the type of RNN used. Also, as the L-star al-

gorithm has since its introduction been extended to a wide range of domains and settings, the proposed technique is likely have broad applicability. The downside is that when the input RNN does not capture a regular language, the state-space of the derived automaton must be bounded by a threshold parameter, or it will grow infinitely large as the inference process proceeds.

3. Hybrid Approaches

3.1 Recursive NNs and parse trees

A natural way of combining discrete structure and neural networks is to use so-called recursive neural networks that take parse trees of natural language sentences as input. For convenience of representation, these parse trees are typically binarized. The general idea is then as follows: The leaves in the tree are labeled by fixed length vectors, most commonly word embeddings obtained separately or trained together with the network. The vector representation of an interior node is computed as a combination of the representations of its two children. For this purpose, a function (network) is trained. Simultaneously, another function is trained that takes a node representation and returns a value. In this way, when the recursive network is run on a tree, each node is assigned value, and the value given to the root can be taken to represent the entire tree. An early training algorithm for recursive networks was developed by (Goller and Küchler, 1996). More recently, several modified algorithms have appeared. One example is the algorithm from (Socher et al., 2013), which works with multidimensional weight matrices and underlies the Stanford NLP sentiment analysis.[1]

It should be mentioned here that the discrete parse trees mentioned above are often obtained using methods that involve neural networks. As a matter of fact, parsing is an area where hybrid approaches are used. For example, transition based dependency parsers use transition systems with stacks, but some of them employ neural networks to decide what transitions to use; see, e.g., (Chen and Manning, 2014).

3.2 Weighted automata

Another example of hybridization is the combination of NNs and weighted finite state automata (WFA). (Li et al., 2017) propose a type of WFA in which the internal linear weight function has been replaced by a non-linear one. The authors show that their model can be efficiently trained by a spectral algorithm that uses an auto-encoder network to adjust weights.

The same author team continue their work in (Rabusseau et al., 2018), wher they study the relationship between WFAs and second order recurrent neural networks (2-RNN). The latter model is a recurrent network where there is, for each hidden neuron, a weight corresponding to every pair consisting of an input and a hidden layer neuron. This means that for a network with k inputs and n hidden neurons, the weights are indexed by $W_{i,j,k}$, where $i, j \in \{1, \ldots, n\}$ and $k \in \{1, \ldots, k\}$. At time t, given input vector $I^{(t)}$ and with current values $S^{(t)}$ for the hidden neurons, the activation and next value for the neurons is computed by the equations

$$a_i = \Sigma_{j,k} W_{ijk} S_j^{(t)} I_k^{(t)} \quad \text{and} \quad S_i^{(t+1)} = g(a_i),$$

where g is some suitable activation function such as a sigmoid. The authors show that for input sequences of discrete symbols, WFAs and second order RNNs with *linear activation functions* are equivalent. This leads to the conclusion that linear 2-RNNs are an extension of WFAs, since they can also handle non-discrete input. To conclude, Li et al. give an extension of the spectral learning algorithms for WFAs to a proven learning algorithm for linear 2-RNNs.

3.3 Soft patterns

Another neural version of WFA is named SoPa, which is short for *soft patterns* (Schwartz et al., 2018). Intuitively, this device is a cross between a simple RNN and a convolutional neural network, and designed to efficiently match a text against a set of patterns at the level of word vectors. SoPA can be trained from data, but have to be parameterized with the number and length of the patters to learn. On the upside, they perform as well as, or better than, the baseline NNs, one of which is a bidirectional long short-term recurrent network, and one is a convolutional network.

3.4 Stacks

A third hybrid model is proposed by (Sun et al., 2017), who equip a neural network with a stack. The NN is then trained to recognise a context-free language L, whereupon a push-down automaton for L is extracted from the neural network. In the experiments, context-free languages such as the balanced parenthesis language and 1^n0^n are successfully inferred.

4. Future Directions

When considering practical use cases, the leap from neural networks to finite-state models is not very radical. Most immediately the representation of weights and values typically chosen is constant-sized floating point values (making for a finite-sized structure, and limiting the total information a recursive network can propagate), but beyond this the network design and iterative optimization techniques applied for training are predicated on some insensitivity to small perturbations (i.e. some local smoothness to the objective function). This is evidenced by an ongoing shift towards lower precision representations of weights and values (e.g. small integers or 16-bit floating point), favoring instead higher performance which enables the use of larger datasets (Gupta et al., 2015; Micikevicius et al., 2018).

While it is only a small step from considering such perturbation to considering coarser discretization of the weights, this is not necessarily the most promising approach, rather we propose two partially overlapping directions to consider:

First, given a highly structured neural network, we take as an example here a 2D convolutional network in image processing (the recurrent case in 1D makes for a more tractable but less illustrative case). From such a network,

[1] `http://nlp.stanford.edu/sentiment/`

we can consider extracting a finite-state device which approximates some aspect of the network, without convolution. That is, rather than discretizing the convolution, attempt to construct a device such as a 6-way finite automaton (able to walk in the four cardinal directions as well as up and down in zoom level). Likely, this is best done by letting the automaton at each position query a fragment of the original network, successively shrinking the size of the fragment and limiting the number of steps the automaton may take to force approximation and, hopefully, generalization. It will clearly be hard to capture many properties in such a relatively limited fashion, but finding cases where such automata can capture interesting properties in few steps would produce important information on what aspects of the picture the original network considered important, and may in some cases indeed produce a compact and useful representation of a recognizer.

Second, given work along the lines described in the first point above, one would then wish to score the training procedure not on the quality of the neural network produced, but the quality of the approximating automaton produced, which given sufficient smoothness in the construction would allow the training of the automata model constructed.

5. Conclusion

There is reason to believe that our understanding of neural networks and what functions they can efficiently approximate will improve greatly over the next few years. The research on topics such as understandable AI is intensive and there has already been substantial progress. For example, Lin et al. showed that properties such as symmetry, locality, compositionality and polynomial log-probability are realizable by simple networks. In particular, using deep networks for such properties requires exponentially fewer parameters than flat networks (Lin et al., 2017). A better understanding of what deep neural nets are actually good at will help us determine when and how to use them and for which tasks other methods are better suited.

We mentioned above the discovery by Rabusseau et al. of the equivalence between weighted automata and a certain class of recurrent NNs (Rabusseau et al., 2018). More results of this kind, relating neural network to finite state models may lead back to more use of the easier to understand and analyze finite state models. It may also, as in the case mentioned here, lead to more transferal of knowledge generated by finite state research into NN research.

Finally, going back to (Lin et al., 2017) gives a great indicator of some of the most promising avenues for the future. As research deciphers exactly why the current deep neural network training techniques are so successful for certain sets of problems (elucidating both their power and their limitations) the results should be leveraged to improve learning algorithms and inference techniques for automata in almost a lockstep fashion. This may involve not only the re-imagining of the appropriate state machine models as considered above, but also completely novel ways of short-circuiting the overall training of a neural network into a finite state machine chain.

References

Angluin, D. (1987). Learning regular sets from queries and counterexamples. *Information and Computation*, 75:87–106.

Chen, D. and Manning, C. D. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750.

Cleeremans, A., Servan-Schreiber, D., and McClelland, J. (1989). Finite state automata and simple recurrent networks. *Neural Computation*, 1:372–381.

Giles, C. L., Miller, C. B., Chen, D., Chen, H.-H., Sun, G.-Z., and Lee, Y.-C. (1992). Learning and extracting finite state automata with second-order recurrent neural networks. *Neural Computation*, 4(3):393–405.

Goller, C. and Küchler, A. (1996). Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of the International Conference on Neural Networks (ICNN'96)*.

Grachev, P., Lobanov, I., Smetannikov, I., and Filchenkov, A. (2017). Neural network for synthesizing deterministic finite automata. *Procedia Computer Science*, 119(C):73–82.

Gupta, S., Agrawal, A., Gopalakrishnan, K., and Narayanan, P. (2015). Deep learning with limited numerical precision. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML'15)*, pages 1737–1746.

Li, T., Rabusseau, G., and Precup, D. (2017). Neural network based nonlinear weighted finite automata. *CoRR*, abs/1709.04380.

Lin, H., Tegmark, M., and Rolnick, D. (2017). Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168(6):1223–1247.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., and Wu, H. (2018). Mixed precision training. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*.

Rabusseau, G., Li, T., and Precup, D. (2018). Connecting weighted automata and recurrent neural networks through spectral learning. *CoRR*, abs:1807.01406.

Roche, E. and Schabes, Y. (1997). *Finite-state language processing*. MIT press.

Schwartz, R., Thomson, S., and Smith, N. A. (2018). Bridging cnns, rnns, and weighted finite-state machines. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 295–305.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Sun, G., Giles, C. L., Chen, H., and Lee, Y. (2017). The neural network pushdown automaton: Model, stack and learning simulations. *CoRR*, abs/1711.05738.

Weiss, G., Goldberg, Y., and Yahav, E. (2018). Extracting automata from recurrent neural networks using queries and counterexamples. In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning (ICML'18)*, volume 80 of *JMLR Workshop and Conference Proceedings*, pages 5244–5253. JMLR.org.

Zeng, Z., Goodman, R., and Smyth, P. (1993). Learning finite state machines with self-clustering recurrent networks. *Neural Computation*, 5(4):976–990.

Interactive correction of speech recognition errors: implementation and evaluation for English and Swedish

Peter Ljunglöf, J. Magnus Kjellberg

Department of Computer Science and Engineering
University of Gothenburg and Chalmers University of Technology
peter.ljunglof@cse.gu.se, magnus.kjellberg@chalmers.se

1. Introduction

In the MUSTE project we explore how to make quick fixes to simple texts using as few interactions as possible (Ljunglöf, 2011). There are several situations where this could be useful, such as when you are driving (and don't have access to a keyboard), if your device is too small for a proper keyboard (such as a mobile phone), or if you have a communicative disability (e.g., cerebral palsy, visual impairment, or something else).

Assume that the user dictated a text message in their phone, and the speech recogniser got most of the message correct, but there were a few words that turned out slightly wrong. In the system that we envision, the user would point at the incorrect words, and the phone would then suggest possible substitutions based on phonological, syntactic and semantic properties. The suggestions for substitutions are presented in a menu from which the user can select the correct choice, or ask for a new menu of suggestions. The sentence can be further modified in small steps to finally reach the intended text.

At SLTC in 2016, we presented a very limited study to see if it would be interesting to investigate the approach further (Ljunglöf, 2016), and now we report on a larger-scale study that we conducted during spring 2018. The main goal of our study is to see if this kind of editing interface can be useful: how probable is it that the system suggests the intended correction, and what parameters are important for the system when calculating good suggestions? We present how the experiment system works, how we have evalutated its performance, and the evaluation results, for both English and Swedish speech recognition error correction.

2. Related work

Suhm et al. (2001) give an overview of strategies for speech error correction. One of the main strategies for correcting a misinterpreted word is to select from a list of alternatives, which is what we use in this project. The approach we are using is based on ideas from multimodal text editing (Ljunglöf, 2011), but we are using statistical models instead of grammars to suggest replacements. Liang et al. (2014; 2015) use a similar approach to ours, but they have a slightly more complicated interface with different editing operations, and they only evaluate Japanese. The Parakeet system (Vertanen and Kristensson, 2010) uses even more complex editing operations, making it possible to correct several errors at once, but on the other hand increases the cognitive burden on the user.

Previous evaluations of interactive speech input correction systems have mainly been performed on human subjects (Cuřín et al., 2011; Kumar et al., 2012; Suhm et al., 2001; Vertanen, 2006). In contrast, our evaluation is purely corpus- and lexicon-based and does not involve human subjects, which can be a promising complement to expensive evaluations on human subjects.

3. Implementation

When the selects the incorrect word(s), the system must come up with a reasonable list of substitution words. There are several possible approaches, more or less advanced. In this study we have chosen an approach in the middle when it comes to complexity.

3.1 Datasets

We use the following datasets in our system, for training the algorithms and for evalutation (see table 1):

Language model corpus: A large monolingual corpus for calculating n-gram frequences and language models. We used the English and Swedish Wikipedia,[1] containing approx. 1900m tokens (for English) and 370m tokens (for Swedish), respectively.

Parallel error correction corpus: A parallel corpus with speech recognition errors and their corrected counterparts. To create this corpus, we used a corpus for speech recognition training which consists of recorded utterances paired with gold-standard transcriptions. We automatically transcribed each recorded utterance with speech recogniser, and if the transcription differed from the gold-standard we added this transcription pair to our parallel corpus.

We created the English corpus from two open-source datasets, the VoxForge speech corpus[2] and Mozilla Common Voice,[3] totalling 270k recorded and transcribed utterances. We used the CMU Sphinx speech recognition toolkit[4] to transcribe the recordings. 32% of the utterances were recognised incorrectly, so our English parallel corpus contains 87k utterances.

The Swedish corpus is created from the dataset collected by Nordisk Språkteknologi (NST), freely available from the Norwegian Språkbanken,[5] containing 477k recorded and

[1] Wikipedia downloads, https://dumps.wikimedia.org
[2] VoxForge project, http://voxforge.org
[3] Mozilla Common Voice, https://voice.mozilla.org
[4] CMU Sphinx, http://cmusphinx.sourceforge.net
[5] NST database, https://www.nb.no/sprakbanken/repositorium

Dataset	English	Size	Swedish	Size
Language model corpus	English Wikipedia	1900m tokens	Swedish Wikipedia	390m tokens
Parallel error corpus (transcribed with)	VoxForge + Mozilla (CMU Sphinx)	171k errors	NST database (Google speech)	39k errors
Phonetic dictionary	CMU pronouncing dict.	123k entries	KTH phonetic dict.	938k entries

Table 1: Datasets used for training and evaluation.

	Utterances with errors	Substitutions involving at most 2 words on either side								
		total	1-0	2-0	0-1	0-2	1-1	2-1	1-2	2-2
English	87,307	76,009	6%	2%	8%	2%	40%	14%	12%	15%
Swedish	38,526	41,941	3%	<1%	6%	<1%	58%	17%	8%	8%

Table 2: Statistics for the parallel error corpora.

transcribed utterances. We transcribed 51k of the recordings using Google cloud speech recognition.[6] Google speech returns an n-best list, so we picked a transcription randomly from the 5 best candidates, to increase the number of incorrect transcriptions. Our final Swedish parallel corpus contains 39k utterances.

Table 2 shows the distribution of the different kinds of errors in the parallel corpora. In total there are 76k errors involving at most 2 words on either side for English, and 42k errors for Swedish. Most notable is that the by far most common error is a 1-1 word substitution.

Phonetic dictionary: A dictionary for converting between written text and their phonological representations. For English we used the CMU pronouncing dictionary,[7] containing 123k entries; and for Swedish we used the phonetic dictionary from the KTH Royal Institute of Technology,[8] containing 938k entries.

3.2 Workflow

After the system has recognised an utterance, it presents the sentence to the user. The user can then select a word, which is interpreted by the system as a request to replace the word with another word. The system does this by reordering a large internal dictionary, according to how probable it is that the new word is what the user originally intended when dictating the utterance. After reordering, the n topmost suggestions will be presented to the user, where n depends on the available space for presenting suggestions but in our evaluation we assume $n = 10$.

The system first uses an intitial filtering method to select the 10,000 most promising candidates from the starting dictionary. It is important that the initial filter is both efficient and selects good candidates, so we have tested three different methods for performing the first filter.

The candidates are reordered in a second phase. We use five different methods for calculating the probability that a dictionary word is a good substitution for the selected word. Logistic regression is used to combine the methods, and the candidate words are sorted by their final probability. For evaluation of logistic regression we used 10 times cross validation. The workflow is shown is figure 1.

[6]Google speech, https://cloud.google.com/speech-to-text
[7]CMU-dict., http://www.speech.cs.cmu.edu/cgi-bin/cmudict
[8]KTH Swedish ASR models, https://www.speech.kth.se/asr

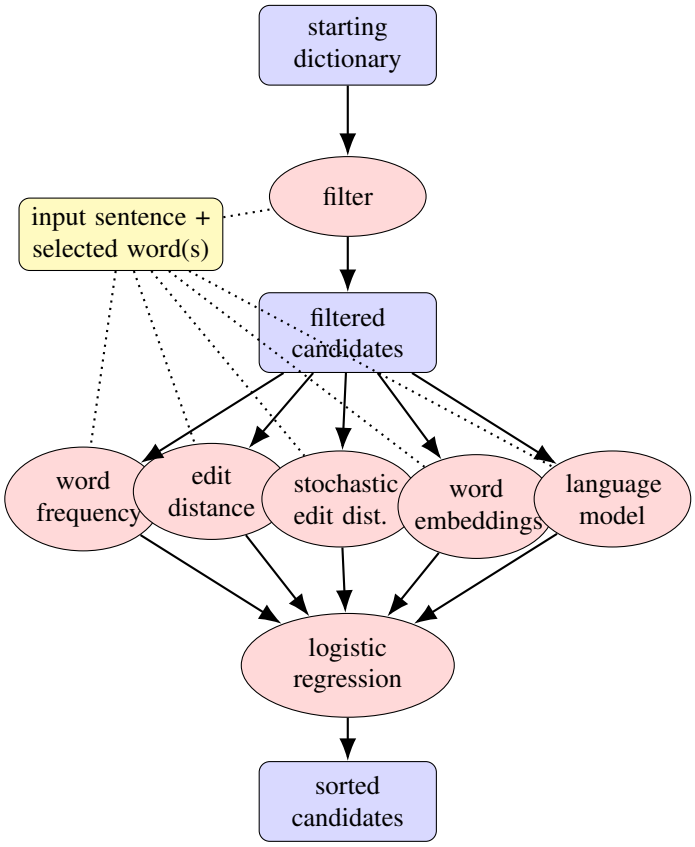


Figure 1: Workflow of the system

3.3 Models for error correction

To estimate the probability that a given dictionary word is the intended word, the system takes into account (1) how common the substitution is according to some corpus, (2) how similar the substitution is to the original word, and (3) how probable it is that the substitution blends in with the rest of the utterance. In our investigation we have implemented and tested five different methods for (1–3).

One of the intentions with our work is to investigate which methods work best for suggesting substitutions for misinterpreted words and phrases. We have implemented and evaluated the following five methods.

Word probability: All the substitution suggestions are taken from a large dictionary which is calculated from the

language model corpus. As an initial ranking of the words and phrases, we calculated unigram and bigram frequencies. To reduce the size of the database, we filtered out all bigrams with frequency less than 5. Finally we transformed all words into their phonological representations, using the dictionary. This resulted in an English frequency distribution for 123k unigrams and 3200k bigrams. For Swedish the corresponding figures are 185k unigrams and 1430k bigrams.

Word similarity: To measure the similarity between the selected word and the substitution, we use a phonological similarity score. This is measured by calculating the *Levenshtein edit distance* (Levenshtein, 1966) between the phonological transcriptions of the erroneous word and the correct word. Since the initial dictionary is so large, we need to be able to quickly filter the words that are close to the erroneous word. For this we pre-calculate a similarity index using SymSpell[9] (a similar algorithm is described by Bocek et al. (2007)). When building the similarity index, we have used a maximum edit distance of 5.

A slightly more advanced similarity measure is the *stochastic edit distance* which uses different weights for different phoneme pairs (Ristad and Yianilos, 1998). To train the weights we have used 10% of the parallel corpus. The implementation is much slower than SymSpell, so we can only perform this in the later reordering phase.

Utterance probability: We train a *KenLM language model* (Heafield, 2011) from the language model corpus. This language model is used for querying the syntactic probability of an utterance when replacing the selected word with an alternative.

Finally, we use *word2vec word embeddings* (Mikolov et al., 2013) trained from the language model corpus, as a semantic probability measure for the substituted utterance.

3.4 Replacing several words

It is possible that a selected word should be split in two or more shorter words (e.g., “awake” vs “a week”). It is also possible that two consecutive words should be merged into one (e.g., “camp fang” vs “campaign”), or even replaced with two other words (e.g., “her die” vs “heard I”). Our system is able to handle both 1- and 2-word substitutions, but the complexity increases when we want find a pair of words to suggest. E.g., the size of the English initial dictionary increases from 123k to 3200k, so the initial filtering method has to process more candidates, and the risk of suggesting bad substitutions increases. Nevertheless, we did conduct an initial study on some two-word substitutions.

4. Evaluation

We performed two evaluations: different methods for the first filtering phase, and different methods (and combinations) for the second reordering phase. These correspond to the pink ellipses in figure 1.

Our main evaluation has been on the most common corrections, where one word is replaced by one word. This is the 1-1 error type in table 2. An initial estimate tells that the accuracy of the other error types are worse, and our methods and workflow will probably need more thinking to improve correction of 1-2, 2-1 and 2-2 errors.

[9] SymSpell, https://github.com/wolfgarbe/SymSpell

Correct substitution...	English	Swedish
...is in intitial dictionary	99%	96%
...remains after first filter		
– SymSpell	84%	56%
– KenLM	67%	29%
– word2vec	82%	—

Table 3: Utterances where the correct suggestion remains after the first filtering phase.

4.1 First filter

We tried three different methods for filtering out the first 10k candidates: SymSpell, KenLM and word2vec. The evaluation was made on 1000 random utterances from the parallel corpus, and we measured for how many utterances, the correct suggestion still remained after the first filtering phase. As seen in table 3, SymSpell and word2vec both performed quite well for English, whereas KenLM fared worse. For more than 80% of the utterances, the correct candidate remained until the second phase. This suggests that the speech recognition errors are normally quite similar to the intended utterance, both with respect to phonology (SymSpell) and semantics (word2vec). We did not try to combine the three methods into a unified first filter, but that is of course a natural next step.

For Swedish the results are worse, and we have not done any investigation as to why this is. But one factor could be that the training corpora are smaller than their English counterpart. We did not have time to evaluate word2vec as first filter, for Swedish.

In addition we performed a limited evaluation of the error types 1-2, 2-1 and 2-2 for English. We only tested SymSpell, and the accuracy drops to 20–30% for these error types. We did not evaluate the second reordering phase for these error types.

4.2 Reordering the candidates

After filtering out the 10k most promising candidates, we reorder them. The n topmost candidates in this ordered list can then be presented to the user, where n depends on the available space for presentation. In this evaluation we assume that $n = 10$.

We tried all possible combinations of our five ranking methods. Table 4 shows the most important results: ALL means that we combine all five methods, $\neg m$ means that all methods except m are combined, and m means that we only used method m. The methods are abbreviated in the table: wf (word frequency), ss (SymSpell), sed (stochastic edit distance), klm (KenLM), and w2v (word2vec). The evaluation was made on 1000 random utterances from the parallel corpus, and we measured for how many utterances, the correct suggestion was among the top-10 suggestions after the second reordering phase.

Not surprisingly, the more methods we combine the better the accuracy. KenLM is the method which contributes the most, which is shown by the drop of accuracy when

	First filter (SymSpell)	The correct substitution is among the top-10 suggestions										
		ALL	¬wf	¬ss	¬sed	¬klm	¬w2v	wf	ss	sed	klm	w2v
English	84%	44%	45%	41%	44%	31%	42%	15%	23%	17%	36%	11%
Swedish	56%	38%	37%	32%	37%	35%	37%	8%	29%	9%	16%	2%

Table 4: Utterances where the correct suggestion is among the top-10 after the second sorting phase. The abbreviations are: wf (word frequency), ss (SymSpell), sed (stochastic edit distance), klm (KenLM), w2v (word2vec).

we leave it out, and the high accuracy when we only use KenLM. Stochastic edit distance seems to not be better than SymSpell, perhaps the weights are trained on too little data. Apart from that, it is difficult to draw conclusive conclusions. 44% of the English errors got the correct substitution among the top-10 candidates. For Swedish the results are in general 5–10 points lower, which probably partly has to do with smaller training data.

5. Discussion and future work

One conclusion to draw from this evaluation is that almost half of all 1-word speech recognition errors can be corrected using this touch-friendly method. With better ranking methods, better combination of the methods, and more training data, we are convinced that the accuracy can increase substantially.

Our next goal is to also increase the accuracy for 1-2, 2-1 and 2-2 substitutions, and perform a serious evaluation of those too. After that there are several possible paths:

- To improve the first filter by combining all methods, and perhaps add more methods – the important issue here is that the methods we use for first filtering must be very efficient.
- Investigate more ranking methods for the second phase, such as morphology or syntax. If available, context could be used for increasing the retrieval rate, e.g., topics and words from previous utterances and conversations. Bidirectional LSTM or other neural network architectures are also possible.
- Improve the datasets – if the main application is to correct text messages, we want to make use of a corpus of text messages.
- The pronunciation dictionaries can be improved, e.g., by using the recent CMU Sphinx G2P toolkit.[10]
- It would probably be very useful to use the internal information from the speech recogniser. Either the n-best list of results, or the internal states.

6. Acknowledgements

This research is funded by Chalmers ICT Area of Advance, and the Swedish Research Council (Vetenskapsrådet).

References

Thomas Bocek, Ela Hunt, and Burkhard Stiller. 2007. Fast similarity search in large dictionaries. Technical report, Department of Informatics, University of Zurich, April. http://fastss.csg.uzh.ch/.

Jan Cuřín, Martin Labský, Tomáš Macek, Jan Kleindienst, Holger Quast, Hoi Young, Ann Thyme-Gobbel, and Lars König. 2011. Dictating and editing short texts while driving: Distraction and task completion. In *AutomotiveUI 2011, 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, Salzburg, Austria.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of SMT 2011, the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, UK.

Anuj Kumar, Tim Paek, and Bongshin Lee. 2012. Voice typing: A new speech interaction model for dictation on touchscreen devices. In *Proceedings of CHI 2012, SIGCHI Conference on Human Factors in Computing Systems*, Austin, Texas, USA.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Yuan Liang, Koji Iwano, and Koichi Shinoda. 2014. Simple gesture-based error correction interface for smartphone speech recognition. In *Proceedings of Interspeech 2014*, Singapore.

Yuan Liang, Koji Iwano, and Koichi Shinoda. 2015. Error correction using long context match for smartphone speech recognition. *IEICE Transactions on Information and Systems*, E98–D(11):1932–1942.

Peter Ljunglöf. 2011. Editing syntax trees on the surface. In *Nodalida'11: 18th Nordic Conference of Computational Linguistics*, Rīga, Latvia.

Peter Ljunglöf. 2016. Towards interactive correction of speech recognition errors. In *SLTC'16, 6th Swedish Language Technology Conference*, Umeå, Sweden.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS'13, 26th International Conference on Neural Information Processing Systems*, pages 3111–3119.

Eric Ristad and Peter N. Yianilos. 1998. Learning string edit distance. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 20, May.

Bernard Suhm, Brad Myers, and Alex Waibel. 2001. Multimodal error correction for speech user interfaces. *ACM Transactions on Computer-Human Interaction*, 8(1):60–98.

Keith Vertanen and Per Ola Kristensson. 2010. Intelligently aiding human-guided correction of speech recognition. In *Proceedings of AAAI'10, the Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Keith Vertanen. 2006. Speech and speech recognition during dictation corrections. In *Proceedings of Interspeech 2006*, Pittsburgh, Pennsylvania, USA.

[10] G2P toolkit, https://github.com/cmusphinx/g2p-seq2seq

Towards an Annotation of Narrative Structure in Literary Fiction

Mats Wirén, Adam Ek and Robert Östling

Department of Linguistics
Stockholm University
SE-106 91 Stockholm, Sweden
{mats.wiren, adam.ek, robert}@ling.su.se

Abstract

This is a progress report from a project with two interrelated aims: to develop an annotation scheme for narrative structure in literary fiction, and to develop computational methods to perform aspects of such annotation automatically. We have begun the latter by designing one of the first methods for identification of speakers and addressees in literary fiction, and describe this along with the underlying annotation scheme and its motivations.

1. Background and Introduction

If the analysis of linguistic structure attempts to answer the question "Who does what to whom?", narrative structure can be said to deal with "Who tells what, and how?" (Jahn, 2017, Section N2). The first question in narrative structure thus concerns aspects such as who is speaking, whether it is a character in the story, and if it is a first-person or third-person narrator. The second question is related to the basic elements of the story: characters and events, and how the sequence of events forms a plot. The third question concerns how the narrative is constructed: ordering of the events, the perspective from which the narrative is seen, how much information the narrator has access to, etc.

Analysing narrative structure in fiction is obviously useful in literary science, but what are the wider implications? A brief answer is that narrative structure is manifested in many other domains than fiction, such as journalism, political discourse and religious text. Arguably, stories and narration are everywhere, and uncovering their structure provides a level of analysis that naturally builds on, but goes beyond, linguistic structure.

We have developed an annotation scheme for narrative structure which covers aspects of all three questions above (Wirén et al., 2018). Furthermore, we have developed a method for computational analysis related to the second question above, namely, identification of speakers and addressees in the dialogue between characters in a story (Ek et al., 2018).

Many aspects of narrative structure are currently lacking in our annotation scheme. To guide further development of this, general works in narratology (Rimmon-Kenan, 2002; Genette, 1983) as well as computationally oriented frameworks will be valuable. With respect to the latter, we expect that TimeML (Pustejovsky et al., 2005) will be useful as a basis for the representation of the temporal ordering of events that is currently missing. Furthermore, the MPQA Opinion Corpus (Wiebe et al., 2005) may be a valuable source if we decide to represent the opinions, beliefs or sentiments of speakers. Discourse-annotated corpora, most notably the Penn Discourse Treebank (Prasad et al., 2008) and the RST Treebank (Carlson et al., 2001), will be similarly valuable.

This purpose of this paper, however, is to report the current state of our interrelated annotation scheme and computational analysis.

2. Annotation Scheme

Literary fiction typically consists of passages alternating between the two following levels (Jahn, 2017, Section N2.3):

1. Transmission from a narrator to a narratee. We refer to this as *narrator's discourse*.
2. Transmission between characters in the story, such as spoken dialogue, interior monologue or thoughts. We refer to this as *characters' discourse* or simply *dialogue*.

To annotate these kinds of discourse, we use opening and closing variants of the tags <NARRATOR> and <CHARACTERS>. The latter thus corresponds to dialogue between the characters, consisting of one or more *turns* (annotated as <TURN>), each of which we assume is associated with one speaker and one or more (possibly differing) addressees. A turn consists of one or more *lines*, each of which we assume has the same addressee(s).

The following passage (from Sally Rooney (2017), *Conversations with Friends*, London: Faber & Faber, page 112) illustrates the main aspects of our annotation scheme that are relevant here.

<NARRATOR>
Nick started laughing then. Melissa just looked away as if she wasn't paying attention to the conversation. I pulled my shoulders back fractionally to feel Nick's arm against my skin.
</NARRATOR>
<CHARACTERS>
<TURN>
We're all on the same side here, Derek said.
<Derek--ALL>
Nick, you're an oppressive white male, you back me up. <Derek-- Nick>
</TURN>
<TURN>
I actually quite agree with Bobbi, said Nick. Oppressive though I certainly am.

```
  <Nick--Derek>
</TURN>
</CHARACTERS>
```

The first turn is divided into two lines since there is a change in addressee (indicated by a vocative). The second turn consists of one line. Each line is annotated with its speaker and addressee, respectively (for example, `<Nick--Derek>`).

Not all the words in what we here annotate as a line are necessarily being spoken by a character. In the first and third line above, the narrator attributes the speech to a character by using a speech-verb construction (”Derek said” and ”said Nick”, respectively). Since these constructions are relatively predictible, we have chosen not to annotate them for the version of the annotation here, but there are clearly more general forms of narration inside lines that will require this.

3. Identifying Speakers and Addressees

Identifying speakers in dialogue, also known as quote attribution, has been explored in literary fiction by, among others, Elson et al. (2010), O'Keefe et al. (2012), He et al. (2013) and Muzny et al. (2017). As far as we know, however, the problem of identifying addressees in literary fiction has only been dealt with by Yeung and Lee (2017).

Basically, authors can indicate the identity of speakers and addressees explicitly, often with a speech verb for the speaker (”Derek said”) or a vocative for the addressee (”Nick, you're…”), anaphorically (using a pronoun or definite description), or implicitly (as in the line beginning with ”Nick, you're…”, where the speaker has to be inferred from the previous context). Many other sources of information, such as the default order of turn-taking, are also available to the reader.

We have developed a method that performs identification of speakers and addressees using an averaged perceptron model[1]. The model is trained and tested on data pooled from parts of four Swedish novels: August Strindberg, *The Red Room* (1879), Hjalmar Söderberg, *The Serious Game* (1912), Birger Sjöberg, *The Quartet That Split Up*, part I (1924) and Karin Boye, *Kallocain* (1940). The data used is in Swedish, and contain in total 822 lines of dialogue.

As a basis for the method, the set of speakers and addressees was extracted from the annotation, along with a list of speech verbs. A dialogue consists of one or more turns, and each turn consists of one or more lines. The task is to assign a speaker and an addressee label to each line in the dialogue. This was viewed as a sequence labelling task in the sense of predicting a sequence of speaker and addressee characters. To select the best sequence of characters, a beam search of size 10 was used.

For each line, features are extracted for each character. The features used are all binary, and capture the following information:

- Frequency and mentions in the immediately preceding narrator's discourse and in all preceding narrator discourses.
- Character mentions and character mentions with speech verb in the current and two preceding lines of dialogue.
- The recency of the latest mention (for example, x is the nth most recently mentioned character).
- Hypothesised sequence, for example, which characters have been assigned as speakers and addressees in the sequence currently.
- To resolve pronouns the character mentioned most recently is used.

The evaluation was performed using four-fold cross-validation, treating each author as a fold. The results were compared to three baselines for speakers and addressees, respectively: a random baseline (two characters are selected randomly and are alternately distributed throughout the lines); a latest-mention baseline (the two latest mentioned characters are alternately distributed throughout the lines), and a modified latest-mention baseline (the two latest mentioned characters occurring with a speech verb are alternately distributed throughout the lines). The results of the perceptron model compared to the baselines are shown in Table 1.

Table 1: Accuracy of the averaged perceptron model with respect to the speaker and addressee identification task compared to three baselines.

SYSTEM	SPEAKER	ADDRESSEE
Random	27.0	23.6
Latest mention	44.6	39.9
Latest mention + speech verb	29.2	28.1
Perceptron	**63.7**	**46.0**

In this paper, we present some additional results using English texts. None of the features used in identifying speakers and addressees rely specifically on Swedish. Thus, adapting the model to English texts only required adding English speech verbs and personal pronouns to the existing list of such words.

In the new experiment, we trained the model using the previous dataset containing excerpts from Swedish novels, and use an English corpus of short stories and excerpts from novels as the test set.[2]. In total, the new test data contains 306 lines of dialogue. The performance of the model using the English dataset as test set is presented in Table 2

4. Discussion

This section discusses our results and ideas for further work.

4.1 Results

The performance of our method for speaker identification is well above the baselines, but it is lower than in previous

[1]For a more in-depth description see (Ek et al., 2018)

[2]The corpus was provided by the SANTA workshop: `https://sharedtasksinthedh.github.io/2018/01/20/corpus/`

Table 2: Accuracy of the averaged perceptron model with respect to the speaker and addressee identification task compared to three baselines (English test set).

SYSTEM	SPEAKER	ADDRESSEE
Random	43.3	36.0
Latest mention	39.6	36.1
Latest mention + speech verb	43.2	35.7
Perceptron	**69.6**	**48.6**

approaches. We think that this is largely due to the fact that our annotated dataset is smaller than in previous studies, and that the variation is larger since we have more authors.

The random baseline for the English test data is higher than the baselines for the Swedish data, indicating that the texts should be easier to analyse. The performance of the model for the English data is better than the one for the Swedish data, but compared to the baseline improvements we would expect the model's performance to be higher. The new English test data is quite different from the Swedish data. The texts are shorter self-contained stories, and generally not chapters within a novel. As such, there tends to be less information in the running text about character mentions and hence mention order, which proved to be important features in Ek et al. (2018).

For addressees, comparisons are more difficult to make. Our evaluations are made on a different authors, whereas Yeung and Lee (2017) use in-domain training data. Consequently, our results should be generalisable to a higher extent. Furthermore, the actual problem studied by Yeung and Lee (2017) is identification of listeners, which is not necessarily the same thing as addressees.

The performance for speakers and addressees differ more than the respective baselines. This suggests that addressees are harder to predict than speakers based on the surface indicators used by our model. Speakers often have reliable indicators in the form of speech verbs, whereas a missing speech verb may signal either an addressee or a person who is not a participant in the dialogue.

4.2 Improved Speaker and Addressee Identification

The features used for speaker and addressee identification are based on surface indicators as described in Section 3. In other words, the features are non-linguistic in the sense that they are not based on a prior linguistic annotation. Although we achieved results well above the baselines in spite of this, it is something that we would like to change in the future. By introducing features based on linguistic annotation such as part-of-speech tagging, syntactic structure and co-reference, we would expect to improve the performance of the model.

A related aspect is that this is necessary to free us from preprocessing of the text and achieve the goal of performing annotation of narrative structure automatically. For example, named-entity recognition could be used to collect the names and aliases of the set of characters, and semantic-role labelling could be used to directly identify the speakers and addressees of lines in dialogue.

In addition to benefitting digital humanities, it should be mentioned that there is at least one possible practical application of a system like this: It could facilitate the production of audiobooks, given that voice actors need to keep track of which character is speaking when (and to who).

4.3 Extended Modelling of Narrative Structure

Characters' discourses can be seen as the lowest or most indirect level of narrative transmission in the sense that the events of a story come across solely through the spoken lines of persons engaging in dialogue. Why have we started to build our computational model from this low level? One reason is that it is very concrete and therefore more amenable to formalisation than higher levels. Authors use various clever indicators to signal to the reader who is speaking and who is being addressed, and by learning how to recognise these indicators, we can build a representation of this level. This can then be used as a building block in the modelling of higher (more abstract) levels.

Our guideline (Wirén et al., 2018) includes several aspects of narrative structure that we have not begun to address computationally. One such aspect is that discourse levels may be embedded into each other. For example, when a character is quoting or recounting a dialogue with someone else, this is represented by embedding that characters' discourse into the current one. This is annotated as an additional opening of `<CHARACTERS>` inside the present one, as in the following example from *Conversations with Friends* (page 145):

```
<CHARACTERS>
<TURN>
I think your wife is a little on edge today, said
Bobbi. <Bobbi--Nick>
She was not impressed with my linen-folding
technique earlier. Also, <Bobbi--Nick>
<CHARACTERS>
<TURN>
she told me she didn't want me 'making any
snide remarks about rich people' <Melissa--
Bobbi>
</TURN>
</CHARACTERS>
when Valerie gets here. Quote.
</TURN>
</CHARACTERS>
```

Our guidelines also include notions related to the perspective of the narrator (Genette, 1983, page 188): *Voice* concerns the narrator's relationship to the story, and more specifically whether the narrator is present in the story or not. *Focalisation* corresponds to the perspective from which the narrative is seen, and specifically how much information the narrator has access to.

Our plan is to gradually extend the computational modelling to experiment with aspects like those mentioned above, motivated by our annotation scheme.

Acknowledgements

We thank the reviewers for very helpful comments. This work has been supported by an infrastructure grant from the

Swedish Research Council (SWE-CLARIN, project 821-2013-2003).

References

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2001. Building a Discourse-tagged Corpus in the Framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue – Volume 16*, SIGDIAL, pages 1–10, Stroudsburg, PA, USA. Association for Computational Linguistics.

Adam Ek, Mats Wirén, Robert Östling, Kristina Nilsson Björkenstam, Gintarė Grigonytė, and Sofia Gustafson Capková. 2018. Identifying speakers and addressees in dialogues extracted from literary fiction. In *Language Resources and Evaluation Conference, Miyazaki, Japan, 7–12 May 2018*. European Language Resources Association.

David K. Elson, Nicholas Dames, and Kathleen R. McKeown. 2010. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 138–147, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gérard Genette. 1983. *Narrative Discourse: An Essay in Method*. Cornell paperbacks. Cornell University Press.

Hua He, Denilson Barbosa, and Grzegorz Kondrak. 2013. Identification of speakers in novels. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1312–1320, Sofia, Bulgaria, August. Association for Computational Linguistics.

Manfred Jahn. 2017. Narratology: A guide to the theory of narrative. English Department, Universität zu Köln, Köln, Germany. `http://www.uni-koeln.de/~ame02/pppn`.

Grace Muzny, Michael Fang, Angel Chang, and Dan Jurafsky. 2017. A two-stage sieve approach for quote attribution. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 460–470, Valencia, Spain, April. Association for Computational Linguistics.

Tim O'Keefe, Silvia Pareti, James R Curran, Irena Koprinska, and Matthew Honnibal. 2012. A sequence labelling approach to quote attribution. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 790–799. Association for Computational Linguistics.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of LREC*.

James Pustejovsky, Robert Knippen, Jessica Littman, and Roser Sauri. 2005. Temporal and Event Information in Natural Language Text. *Language Resources and Evalutation*, 39(2):123–164.

Shlomith Rimmon-Kenan. 2002. *Narrative fiction: Contemporary poetics*. Routledge, 2nd edition.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating Expressions of Opinions and Emotions in Language. *Language Resources and Evalutation*, 39(2–3):165–210.

Mats Wirén, Adam Ek, and Anna Kasaty, 2018. Guidelines for annotation of narrative structure. Department of Linguistics, Stockholm University, Stockholm, Sweden. To be published in *Cultural Analytics*, `http://culturalanalytics.org/`.

Chak Yan Yeung and John Lee. 2017. Identifying speakers and listeners of quoted speech in literary works. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 325–329. Asian Federation of Natural Language Processing.

Language Model Perplexities as Multi-Word Distributional Vectors of Spatial Relations

Mehdi Ghanimifard and Simon Dobnik

Centre for Linguistic Theory and Studies in Probability (CLASP)
University of Gothenburg, firstname.lastname@gu.se

Abstract

Understanding and generating spatial descriptions requires knowledge about the objects that are related and their geometric location. The wide usage of neural language models in different areas including generation of scene description motivates the study of what kind of knowledge is encoded in neural language models about spatial relations. In order to examine this we build distributional representations of multi-word spatial relations based on the perplexity measure of a neural language model. We compare these representations with standard word embeddings in two simple intrinsic tests involving lexical semantic reasoning with spatial relations.

1. Introduction

Spatial descriptions such as "the chair is to the left of the table" contain spatial relations "to the left of" which need to be grounded in visual and perceptual representations in terms of their geometry which is known as symbol grounding (Harnad, 1990).

Experimental studies involving human judgements imply an interplay between geometry and object-specific function in the comprehension of spatial relations (Coventry et al., 2001). Therefore, spatial descriptions must be grounded in two kinds of knowledge. One kind of knowledge is referential meaning of spatial relations, expressed in the geometry of scenes (geometric knowledge (Coventry and Garrod, 2004) or *where* objects are (Landau and Jackendoff, 1993; Landau, 2016)). The other kind of knowledge is higher-level conceptual world knowledge about interactions between objects and their affordances, which is not directly grounded in perceptible situations but is learned through our experience of situations in the world (functional knowledge (Coventry and Garrod, 2004) or *what* objects are related (Landau and Jackendoff, 1993; Landau, 2016)). The success of distributional semantics (Turney and Pantel, 2010) shows that such knowledge can be can extracted from natural language corpora (Dobnik and Kelleher, 2013; Dobnik and Kelleher, 2014).

(Logan and Sadler, 1996) build a vector of human acceptability scores over possible locations which is then used as geometric perceptual representation of a spatial relation. When describing a pair of objects, these vectors can be used to determine the goodness of fit of each spatial template given these objects. They compare geometric vector space representations (of 10 spatial relations from spatial templates) with vector representations from human judgements on how similar these expressions are in the absence of spatial scenes in general. They observe both measures of similarity capture clusters of similar pairs *above/over*, *below/under*, *near to/next to*, and *away from/far from*, but human similarity judgements suggest *left of* and *right of* are similar while geometrically they have the highest distance. An open and interesting question is how both kinds of knowledge interact in choosing a spatial description in natural language generation or disambiguating a visual scene in natural language understanding and how such knowledge can be represented in computational applications. (Ramisa et al., 2015) study the contributions of each feature representations (visual, geometric and textual) in prediction of prepositions. (Schwering, 2007) examine semantic similarity of spatial relations metrics for geographical data retrieval. (Dobnik and Kelleher, 2013; Dobnik and Kelleher, 2014; Dobnik et al., 2018) demonstrate that distributional knowledge can distinguish between functional and geometric bias of spatial relations.

In this paper we examine what knowledge about spatial relations can be learned from text. In particular, we examine distributional representations of spatial relations in spatial descriptions, including word embeddings and distributional representations captured by a simple neural language model. We apply this knowledge in two simple analogical reasoning tests with spatial relations.[1]

2. Representations of spatial relations

Distributional semantic models produce vector representations which capture latent meanings hidden in association of words and documents (Church and Hanks, 1990; Turney and Pantel, 2010). The neural word embeddings which have gained popularity in variety of NLP tasks were initially introduced as a component in neural language models (Bengio et al., 2003). Subsequently, neural language models such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) have become used to specifically learn word embeddings from larger corpora. The word embeddings trained by these models capture world-knowledge regularities in language which can be used for analogical reasoning. For example, "a is to a^* as b is to b^*" can be queried with simple vector arithmetic $king - man + woman \approx queen$[2]. (Levy et al., 2015) show that these properties of word vector representations are not limited to

[1] Our code used in evaluation is available at https://github.com/GU-CLASP/spatial_relations_vectors_sltc2018.

[2] More specifically, with a search over vocabulary with cosine similarity:

neural word embeddings and that other distributional word representations can also handle these tasks to some degree when used on a large dataset and when enhanced with similar strategies or hyper-parameters.

3. Recurrent neural language models for distributional representations

Compared to window-based neural language models for word embedding learning such as Word2Vec and GloVe, recurrent neural language models can be used as generative language models and an estimator of probability of any word sequence. Generative language models often use the chain rule of probability for step-by-step prediction of the next word in a sequence or they can be used in beam-search for finding the best word sequence. In these models, the probability of a phrase or a sentence is defined as the multiplication of conditional probabilities of each word given previous context in a sentence or a phrase.

$$P(w_{1:T}) = \prod_{t=1}^{T-1} P(w_{t+1}|w_{1:t}) \quad (1)$$

where T is the maximum length of the word sequence. Essentially, after optimising parameters over enough batches of data the neural network can estimate the probability of a sequence in Equation 1. The probability of a word sequence is a measure of its commonness which is the opposite of perplexity. The perplexity is often used for expressing the fit of a model to a given test set , but here we will use it as a measure of appropriateness of a test set for a given pre-trained model.

First, we train a neural language model on a large corpus of scene descriptions where we expect a high proportion of prepositions being used in their spatial senses. We implement a recurrent language model with LSTM (Hochreiter and Schmidhuber, 1997) with a word embeddings layer similar to (Gal and Ghahramani, 2016) in Keras (Chollet and others, 2015) with TensorFlow (Abadi et al., 2015) as back-end. The Adam optimiser (Kingma and Ba, 2014) is used for fitting the parameters.

Then, we create a set of patterns for extracting spatial relations including their compound variants based on lists of spatial relations in (Landau, 1996) and (Herskovits, 1986). For each spatial relation, we extract a collection of sentences from a smaller holdout set not used in the training of the language model. The measure of perplexity calculated on a collection of sentences containing the same spatial relation is related to a joint probability of seeing that spatial relation in the extracted context (that target and landmark are part of), which can be roughly expressed as follows:

$$PP(S_{rel}) = P(rel, c_{rel})^{\frac{1}{-N}} \quad (2)$$

where c_{rel} is the set of contexts in a collection of the spatial relation rel, and N is the total number of instances in this collection S_{rel}. For our list of spatial relations $\{r_1, r_2, ..., r_k\}$, we can also use neural network to estimate the joint probability of using each relation in the context of other relations c_j if we artificially create test collections by swapping the relation multi-words in given sentences $S_{i \to j}$ (e.g. replace *to the right of* with *in front of* in its collection of sentences):

$$PP(S_{i \to j}) = PP_{i,j} = P(r_i, c_j)^{\frac{1}{-N'}} \quad (3)$$

where $PP_{i,j}$ is a shorthand notation for the perplexity measure of the neural language model on a sentence collection where relation i is artificially used in the contexts of relation j. If r_i and r_j are associated with two very different contexts, then we expect high perplexity for $S_{i \to j}$.

4. Perplexity vector representations

4.1 Hypothesis 1

The perplexities calculated using Equation 3 on all possible collections of contexts of spatial relations create a confusion matrix. In each cell, high perplexity of a relation swapped into a particular context means less swapability between the two spatial relations, while low perplexity means high swapability and therefore semantic similarity between the two spatial relations. Here we use the perplexity matrix in a way that is similar to a word-context distributional matrix where each vector of the matrix represents a semantic fingerprint for a spatial relation, namely how swappable it is in different contexts. The hypothesis is that such a normalised vector space can be used to detect semantic similarity of spatial relations. We normalise perplexity as follows:

$$m_{i,j} = \frac{PP_{i,j}}{\sum_{j'=1}^{k} PP_{i,j'}}. \quad (4)$$

$$v_i = [m_{i,1}, ..., m_{i,k}] \quad (5)$$

where v_i is the vector representation of the relation r_i.

4.2 Method 1

We use Visual Genome (Krishna et al., 2017) image description corpus. We split the dataset into 90%-10% portions, 90% for training the language model, and 10% for extracting sentences with spatial relations and estimating their perplexity vectors. In order to reduce the noise, the spatial relations with less than 100 instances are removed from our test-set. This leaves us with 29 spatial relations.[3] After swapping all spatial relations in each artificial collection, the perplexity of each collection is computed and normalised and represented as unit vectors as in Equations 4 and 5. *k-means* clustering with $k = 10$ is used to create centroids of clusters which gives us a quantitative representation on how spatial relations are related to each other which can be later examined qualitatively.

4.3 Results 1

The summary of the perplexity vector representation for all 29 spatial relations is shown in Figure 1. The figure shows that clusters of spatial relations can be identified. After applying k-mean clustering on the generated vectors with $k = 10$ we observe meaningful clusters as in Table 1.

$$\underset{b^* \in V/\{a^*,b,a\}}{arg\,max} \quad cos(b^*, a^*a + b)$$

[3] We also examined a case where sentences with low frequency relations were not removed which gave us 97 single- and multi-word relations in total.

above over	on	in to at
beneath below under underneath	to the left of to the right of in back of in the back of in front of	with without
on the front of on back of on front of on the back of	behind in between between by	through outside out
back of		

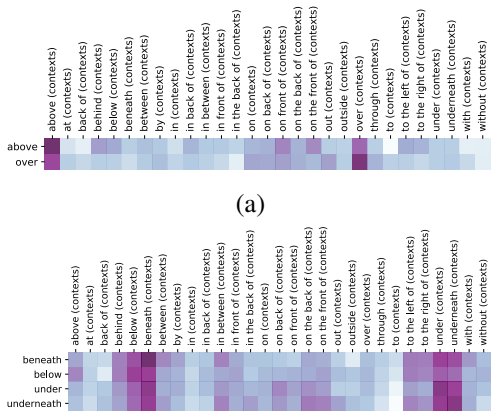
Table 1: The clusters found with nearest neighbours algorithm with $k = 10$

Figure 1: The normalised perplexity matrix of using a spatial relation (rows) in contexts of every other spatial relations (columns). The colour of each cell represents the perplexity of the language model on the collection of sentences representing that context.

These results show association of semantically similar spatial relations such as Figure 2 *above* and *over* as one cluster and *beneath*, *below*, *under*, *underneath* as another cluster. However, what is also interesting is that multi-word expressions containing *left* and *right* are also clustered together as well as multi-words containing *front* and *back*.

5. Analogical reasoning tasks

We can also evaluate the intrinsic properties of vector representations with analogical reasoning tasks. Here we compare the performance of the perplexity vector representations with traditional word embeddings models and therefore we can only use keywords from relations rather than full phrases as explained in Section 2. We hypothesise that distributional representations (including perplexity vector representations) give us intuitive analogical reasoning results.



(a)

(b)

Figure 2: The vectors of two clusters (a) *above*, *over*, (b) *beneath*, *below*, *under*, *underneath*

5.1 Method 2.1

The inference experiment used in this task is similar to the Google analogy test where relations like "a is to a^* as b is to b^*" are used (Mikolov et al., 2013). We manually associate words that are opposite in one geometric dimension as follows:

Group 1	Group 2	Group 3
left, right	above, below	front, back
Group 4	Group 5	Group 6
with, without	in, out	up, down
Group 7		
away, near		

Table 2: Hand-picked and clustered geometrically opposite words

The reason for introducing a test for these pairs is that according to our intuition these will be particularly hard to distinguish by textual word distribution representations. We generate all possible permutations of the analogy test (168 permutations) as follows: (*above* :: *below*, *left* :: ?) where the expected answer is *right*. We evaluate the resulting instances with three different word embeddings: (i) GloVe trained on Common Crawl dataset[4], (ii) GloVe trained on image descriptions from Visual Genome, (iii) word embeddings trained with our recurrent language model on Visual Genome image descriptions.

We also created another variant of this dataset which contains all possible permutations of multi-word spatial relations. This gives us 90,580 possible combinations which were used to evaluate the perplexity vector representations: e.g. (*above* :: *below*, *to the left of* :: ?). Here, any variation containing *right* was considered an acceptable answer.[5]

[4] `http://nlp.stanford.edu/data/glove.42B.300d.zip`

[5] In the previous dataset these are collapsed to a single label.

5.2 Results 2.1

The accuracies in predicting the answer to analogical test are presented in the table below:

GloVe (CC)	GloVe (VG)	RLM embs	Perplexities
0.464	0.363	0.720	0.819

The results show that word embeddings in a recurrent language model give us best performance in analogical reasoning. RNNs capture better the contexts of spatial relations. It is surprising that GloVe trained on Visual Genome (where spatial relations are more dominant) performs worse than GloVe trained on general text (Common Crawl) where one would expect more variation but this could be because Common Crawl is much larger dataset than Visual Genome and therefore they are not comparable in this respect.

Note that the perplexity vectors are evaluated on 216 multi-word analogy questions. Multi-word expressions are made based on the same geometrical groups in Table 2 but the results are not directly comparable with keyword-based test for GloVe and RLM embeddings. However, they nonetheless show that perplexity vectors can distinguish very fine semantic distinctions in spatial relations. The strong result for perplexity vectors might be a consequence of learning context representations as embeddings in a recurrent language model.

5.3 Method 2.2

We also design an "odd-one-out" task. This is based around a presentation of three words where one word has to be identified as the odd one. In this analogical test we take each axis and proximity as meaningful dimensions according to which words can vary:

X-axis	Y-axis	Z-axis
left, right	above, over, under, below	front, back
Insideness	**Proximity**	
in, out	away, near	

One of the most known weaknesses of word embedding representations is antonym/synonym distinction. Representations built on contexts and the measure of similarity may put both matching and opposing words close to each other. Our testing instances are permutations of two words from one dimension and a third word from a different dimension. The odd relation must be selected automatically based on the learned vector representations. For example, for the triple (*above*, *under*, *front*) the expected answer is *front*. To identify the most dissimilar word, cosine distance is used in the semantic vector space.

We performed the experiment on the perplexity vectors of multi-word expressions as in the previous task. We also examine this task on geometrically grounded representations collected as spatial templates in (Logan and Sadler, 1996). Geometric information from spatial templates should result in large distance between geometrically opposite words (e.g *left* and *right*) but similarity between geometrically more closely associated words (e.g *above* and *over*).

5.4 Results 2.2

The following table shows the accuracy in predicting the odd relation out of the three relations by different distributional representations:

GloVe (CC)	GloVe (VG)	RLM embs
0.333	0.337	0.333
Spl templates	**Perplexities**	
0.273 [6]	0.406 [7]	

As it was expected, due to the task design the spatial templates give us the lowest results. All three word embeddings also fail the test as 0.333 is equal to a random choice between the three words. However, perplexity vectors give us better results beyond chance: 0.406 (537 out of 1320). One explanation for the success of perplexity vectors is that they are based on multi-word expressions. The similarity of expressions in synonym and antonym pairs may have created a bias that improves the prediction of the odd ones out, e.g. *above* versus *to the left of* and *to the right of*.

6. Conclusion and future work

We tested whether perplexity of a language model trained on descriptions with spatial relations can be used a measure of semantic association for spatial relations. The idea is based on earlier work (Dobnik and Kelleher, 2013; Dobnik and Kelleher, 2014; Dobnik et al., 2018) where it has been shown that different spatial relations occur in different contexts of target and landmark objects. In particular,

(i) we examined and compared the distributional representations of different spatial relations in terms of the target and landmark contexts;
(ii) we introduced a simple distributional model of multi-word spatial relations based on a pre-trained neural language model and the measure of perplexity as a measure of semantic association;
(iii) we provided support to the claim (Kelleher and Dobnik, 2017) that a significant part of semantic information of spatial relations is reflected in their distributional properties and that a neural language model plays a crucial role in generating spatial descriptions, for example in image captioning systems.

The work could be extended in several ways. First of all, the distributional vector representation based on the text could be compared with (i) human judgements of semantic similarity of spatial relations and (ii) geometric representations of spatial relations. For (i) we expect that these would be similar but for (ii) we expect that they are in a complementary distribution, assuming that individual spatial relations show a different bias to each knowledge. The same method of building vector representations based on perplexity could be applied to other kinds of descriptions where fine-grained semantic distinctions are required.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado,

Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

François Chollet et al. 2015. Keras. `https://github.com/keras-team/keras`.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Kenny R Coventry and Simon C Garrod. 2004. *Saying, seeing, and acting: the psychological semantics of spatial prepositions*. Psychology Press, Hove, East Sussex.

Kenny R Coventry, Mercè Prat-Sala, and Lynn Richards. 2001. The interplay between geometry and function in the comprehension of over, under, above, and below. *Journal of memory and language*, 44(3):376–398.

Simon Dobnik and John D. Kelleher. 2013. Towards an automatic identification of functional and geometric spatial prepositions. In *Proceedings of PRE-CogSsci 2013: Production of referring expressions – bridging the gap between cognitive and computational approaches to reference*, pages 1–6, Berlin, Germany, 31 July.

Simon Dobnik and John D. Kelleher. 2014. Exploration of functional semantics of prepositions from corpora of descriptions of visual scenes. In *Proceedings of the Third V&L Net Workshop on Vision and Language*, pages 33–37, Dublin, Ireland, August. Dublin City University and the Association for Computational Linguistics.

Simon Dobnik, Mehdi Ghanimifard, and John D. Kelleher. 2018. Exploring the functional and geometric bias of spatial relations using neural language models. In *Proceedings of the First International Workshop on Spatial Language Understanding (SpLU 2018) at NAACL-HLT 2018*, pages 1–11, New Orleans, Louisiana, USA, June 6. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027.

Stevan Harnad. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346.

Annette Herskovits. 1986. *Language and spatial cognition: an interdisciplinary study of the prepositions in English*. Cambridge University Press, Cambridge.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

John D. Kelleher and Simon Dobnik. 2017. What is not where: the challenge of integrating spatial representations into deep learning architectures. In Simon Dobnik and Shalom Lappin, editors, *Proceedings of the Conference on Logic and Machine Learning in Natural Language (LaML 2017), Gothenburg, 12 –13 June*, volume 1 of *CLASP Papers in Computational Linguistics*, pages 41–52, Gothenburg, Sweden, November. University of Gothenburg, CLASP, Centre for Language and Studies in Probability.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73.

Barbara Landau and Ray Jackendoff. 1993. "what" and "where" in spatial language and spatial cognition. *Behavioral and Brain Sciences*, 16(2):217–238, 255–265.

Barbara Landau. 1996. Multiple geometric representations of objects in languages and language learners. *Language and space*, pages 317–363.

Barbara Landau. 2016. Update on "what" and "where" in spatial language: A new division of labor for spatial terms. *Cognitive Science*, 41(2):321–350.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

G.D. Logan and D.D. Sadler. 1996. A computational analysis of the apprehension of spatial relations. In M. Bloom, P.and Peterson, L. Nadell, and M. Garrett, editors, *Language and Space*, pages 493–529. MIT Press.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Arnau Ramisa, Josiah Wang, Ying Lu, Emmanuel Dellandrea, Francesc Moreno-Noguer, and Robert Gaizauskas. 2015. Combining geometric, textual and visual features for predicting prepositions in image descriptions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 214–220.

Angela Schwering. 2007. Evaluation of a semantic similarity measure for natural language spatial relations. In *International Conference on Spatial Information Theory*, pages 116–132. Springer.

Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.

Word embeddings for 1250 languages through multi-source projection

Murathan Kurfalı*, **Robert Östling***

Department of Linguistics
Stockholm University
SE-106 91 Stockholm, Sweden
{murathan.kurfali, robert}@ling.su.se

Abstract

We present a method for producing word embeddings in 1250 languages, by projecting multilingual embeddings from high-resource languages through a parallel text. Our evaluation shows that these approach the quality of embeddings obtained from large monolingual corpora and bilingual lexicon resources, but with a smaller vocabulary size. The quality increases if multi-source projection is used, even compared to a cherry-picked closely related single source language.

1. Introduction

There exists a large body of work on monolingual (Mikolov et al., 2013a; Bojanowski et al., 2016), supervised multilingual (Klementiev et al., 2012) and, more recently, unsupervised multilingual (Conneau et al., 2017; Artetxe et al., 2017; Søgaard et al., 2018) word embeddings. Large monolingual corpora are required for learning high-quality monolingual embeddings, and indirectly for unsupervised multilingual embeddings that are created by aligning monolingual embedding spaces across languages. Supervised multilingual embeddings require large amounts of parallel text. For the vast majority of languages, a large amount of *any* kind of text is difficult to obtain. Projection-based approaches (Mikolov et al., 2013b; Guo et al., 2015) require large monolingual corpora in *some* language(s), but only a limited amount of parallel text (or lexicon) in the language of interest.

Given the lack of large monolingual corpora for most languages, we see no other choice than a projection-based approach. However, previous work generally assumes that projection is done from *one* high-resource language, typically English. Since the vast majority of languages differ strongly from English (or any one language) in both grammar and lexicon[1], this looks like a sub-optimal choice on average.

2. Method and Data

We use multi-source embedding projection. This requires multilingual word embeddings for the source (high-resource) languages, and parallel texts between the source languages and the target (low-resource) languages.

As source embeddings, we use the vectors produced by Smith et al. (2017)[2], who aligned monolingual embeddings in 78 languages obtained from Bojanowski et al. (2016) into a common space. Apart from the original monolingual corpora (Wikipedia articles), for each language they used a list of 5000 words translated from English as supervision.

Table 1: Source languages used in the experiments.

Bulgarian (bul)	Greek (ell)	Portuguese (por)
Catalan (cat)	German (deu)	Romanian (ron)
Croatian (hrv)	Hungarian (hun)	Russian (rus)
Czech (ces)	Indonesian (ind)	Slovak (slk)
Danish (dan)	Italian (ita)	Slovenian (slv)
Dutch (nld)	Lithuanian (lit)	Spanish (spa)
Estonian (est)	Macedonian (mkd)	Swedish (swe)
Finnish (fin)	Norwegian (nob)	Turkish (tur)
French (fra)	Polish (pol)	Ukrainian (ukr)

For parallel text, we use the Bible corpus of Mayer and Cysouw (2014). In the version used by us, this corpus contains 1698 translations in 1277 different languages. The 27 languages with highest-quality embeddings (word translation precision above 50%) from Smith et al. (2017) where we had alignments available were used as source languages (Table 1), so we project to a total of 1250 languages.[3]

First we obtain pairwise word alignments between all modern translations[4] in the source languages and the target languages. This amounts to 252 840 bitext alignments, so we use the efficient implementetion of the HMM model by Östling and Tiedemann (2016).[5]

To project embeddings for the target type t—typically a word in some low-resource language—we compute the number of times $c(s,t)$ that t is aligned to each source type s in the word alignments. Since we do multi-source projection, different s may come from different languages. To compensate for noise in the word alignments, we use adjusted counts $c'(s,t)$ where any source type with fewer than a proportion k than the most frequent type in the same

* Authors contributed equally.

[1] That is, the *structure* of the lexicon differs, e.g. in patterns of synonymy and polysemy.

[2] Embeddings downloaded from github.com/Babylonpartners/fastText_multilingual

[3] Available for download at http://mumin.ling.su.se/fotran2018.

[4] When a more recent translation is available, we exclude Bible translations older than 100 years since our focus is on the modern language.

[5] We use the *eflomal* version as recommended: github.com/robertostling/eflomal

source language is given an adjusted count of zero:

$$c'(s,t) = \begin{cases} c(s,t) & c(s,t) \geq k \max_{s' \in L_s} c(s',t) \\ 0 & c(s,t) < k \max_{s' \in L_s} c(s',t) \end{cases}$$

where the $\max$ operation is carried out over types s' in the same language L_s as s. This guarantees that types from all languages are used for the projection. For most experiments we use $k = 0.1$, but we also tried $k = 1$, equivalent to choosing only the single most commonly aligned type per language. The projected vector v_t for target type t is computed as

$$v_t = \frac{1}{\sum_s c'(s,t)} \sum_s c'(s,t) v_s$$

that is, it is aligned to the weighted average of the vectors of the word types s, where the (adjusted) alignment counts are used as weights.

3. Experiments and Results

We use two different methods for evaluation. First, we investigate how well the projected embedding space matches the high-resource space from Smith et al. (2017) (see Section 3.1). Second, we use a bilingual lexicon to estimate word-level translation accuracy (see Section 3.2).

3.1 Embedding Reconstruction

Since we have access to a high-quality multilingual embeddings for the 27 languages in Table 1, we begin by trying to reconstruct the embeddings in one language (here, Swedish) from other languages. Then we can simply compute the mean cosine similarity over the whole vocabulary between the original and the reconstructed embeddings, which serves as a rough reconstruction score.

Our first question is: Does projecting from multiple sources work better than projecting from a single source? If so, what does the optimal subset of source languages look like? To this end, we performed a greedy search by first finding the single language with the best reconstruction score, then the best language to add to this, and so on until the gain is less than a fixed threshold (we use 0.001). We wish to emphasize that this method is generally unrealistic in a low-resource scenario, since we do not have the luxury of multiple closely related high-resource languages to choose from. It is used here to illustrate the effect of language (un)relatedness.

Table 3 shows that single-source projection is always suboptimal, even when we are able to choose the best single language (Norwegian, closely related to Swedish) there is a large gap to the best multi-source combination. If English is used, the result is even worse, and if we happen to pick an unfortunate source language (Estonian) the figures drop even further. Significantly, even unrelated or distantly related languages contribute to reconstruction performance.

Qualitatively, reconstruction performance correlates strongly with frequency as expected (see Table 2). Particularly good results (cosine distance below 0.15) are obtained for pronouns, numerals, and a number of common nouns and verbs.

Table 2: Cosine similarity between the original Swedish vectors and projected vectors using only the unrelated languages (ind, fin, hun, tur, est)

Word	**Gloss**	**Cosine distance**
syster	sister	0.11
femte	fifth	0.12
tre	three	0.12
fjärde	fourth	0.12
fyra	four	0.12
eftersom	since/because	0.13
byggnader	buildings	0.13
han	he	0.13
försökte	tried	0.13
fem	five	0.13

Table 3: Reconstruction performance for different sets of source language(s). Swedish is the target.

Source language(s)	Mean cosine distance
est	0.57
eng	0.53
nob	0.44
nob+nld	0.39
nob+nld+dan	0.37
nob+nld+dan+fin	0.36
nob+nld+dan+fin+pol	0.35
nob+nld+dan+fin+pol+bul	0.35
nob+nld+dan+fin+pol+bul+ron	0.35
nob+nld+dan+fin+pol+bul+ron+slv	0.35

Table 4 shows that the projected embeddings tend to become denser than the original spaces, with lower mean distance between word pairs (which are, on average, unrelated). This is likely a result of noisy word embeddings, since the effect is reduced when a more conservative threshold ($k = 1$) is used.

eftersom 0.13 byggnader 0.13 han 0.13 försökte 0.13 fem 0.13 femte 0.12 tre 0.12 fjärde 0.12 fyra 0.12 syster 0.11

3.2 Word-level Translation

In order to assess the quality of the projected word embeddings, we performed word-by-word translation between

Table 4: Mean of the all pairwise cosine distances within an embedding space. Unless otherwise specified, all embedding spaces are limited to the Bible's vocabulary.

Embedding	Cosine distance
Original English (full vocab.)	0.84
Original English	0.81
Original Swedish (full vocab.)	0.81
Original Swedish	0.73
Projected Swedish ($k = 1$)	0.50
Projected Swedish ($k = 0.1$)	0.45

English and Swedish using the bilingual lexicon from Conneau et al. (2017) as gold standard.[6] This lexicon handles polysemy by having separate translations for each sense of a given word, (e.g. som = 'like'; som = 'as'; som = 'which') and in our experiments translations are accepted as correct if they correspond to *any* of the possible senses. We limited our evaluation to the vocabulary of the parallel text, resulting in a vocabulary size of 7767 Swedish and 7647 English words.

Following the standard practice, we report precision @1,@5,@10. In order to see the effect of the multilingual projection, we compare our results against the projected embeddings using only one language (Norwegian in our case, cf. Table 3) as well as against the original embeddings from Smith et al. (2017). For this we use the same restricted vocabulary as above to ensure a fair comparison. Table 5 and Table 6 show that translation performance is comparable to the original high-resource embeddings. Comparing the two, we see that low-to-high resource translation somewhat benefits from a more conservative (higher) value of the threshold k, but the opposite is true for the high-to-low direction.

For the vast majority of languages we do not have the luxury of projecting from several closely related high-resource languages. We therefore repeated the previous experiment but excluded Indo-European languages. Table 7 shows that the resulting embeddings are of acceptable quality, although somewhat lower than when closely related languages are used.

3.3 Modern-to-Ancient Text Translation

A major inherent weakness with the data available—Bible translations—is that the vocabulary of the projected embeddings is limited to concepts present two millennia ago. However, one interesting property of projecting a well-structured embedding space through the Bible is that semantic connections between modern and ancient concepts are captured. We show the effect of this in Table 8. Modern phenomena such as "police" and "truck driver" become "guards" and "wagon driver". The examples are chosen for illustrative purposes, this is not a sensible model for general-purpose machine translation.

4. Conclusions and Future Work

We have shown that word embeddings of reasonable quality can be obtained by multi-source projection from a small number of high-resource languages. To some extent, the problems with limited target-side vocabulary (Section 3.3) can be alleviated by morphological modeling so that the meanings of inflections, derivations and compounds can be estimated. Guo et al. (2015) present a simple solution that could serve as a starting point, but more work in the style of Bojanowski et al. (2016) and beyond is needed in the context of embedding projection.

The projection method itself also needs further development, as we have only explored naive projection based directly on (low-quality) word alignments in this work.

Finally, our evaluation is limited to Swedish as a target language due to available resources and competence. Future work should investigate whether the encouraging results from our study hold for a wider range of low-resource languages across the world.

[6] https://github.com/facebookresearch/MUSE

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1234–1244. Association for Computational Linguistics.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*, pages 1459–1474. The COLING 2012 Organizing Committee.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel bible corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA).

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *Prague Bulletin of Mathematical Linguistics*, 106:125–146, October.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *CoRR*, abs/1702.03859.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 778–788. Association for Computational Linguistics.

Table 5: English–Swedish word translation performance, $k = 0.1$.

	Eng to Swe			Swe to Eng		
Embedding	p@1	p@5	p@10	p@1	p@5	p@10
Swedish Fasttext Embedding	0.501	0.686	0.745	0.525	0.722	0.780
nob	0.275	0.493	0.567	0.344	0.521	0.578
nob+nld	0.344	0.582	0.657	0.381	0.562	0.621
nob+nld+dan	0.368	0.605	0.673	**0.386**	**0.569**	**0.632**
nob+nld+dan+fin	0.389	0.615	0.682	0.373	0.552	0.618
nob+nld+dan+fin+pol	**0.400**	0.623	0.691	0.372	0.556	0.621
nob+nld+dan+fin+pol+bul	0.392	**0.626**	**0.690**	0.363	0.546	0.608
nob+nld+dan+fin+pol+bul+ron	0.392	0.622	0.688	0.355	0.543	0.605
nob+nld+dan+fin+pol+bul+ron+slv	0.394	0.620	0.686	0.355	0.542	0.606

Table 6: English–Swedish word translation performance, $k = 1$.

	Eng to Swe			Swe to Eng		
Embedding	p@1	p@5	p@10	p@1	p@5	p@10
Swedish Fasttext Embedding	0.501	0.686	0.745	0.525	0.722	0.780
nob	0.231	0.432	0.495	0.346	0.493	0.538
nob+nld	0.304	0.513	0.578	0.391	0.555	0.608
nob+nld+dan	0.342	0.556	0.622	0.402	0.571	0.628
nob+nld+dan+pol	0.355	0.577	0.644	**0.410**	**0.585**	**0.642**
nob+nld+dan+pol+fin	0.369	0.585	0.654	0.403	0.580	0.638
nob+nld+dan+pol+fin+bul	0.374	0.596	0.656	0.397	0.577	0.632
nob+nld+dan+pol+fin+bul+ron	0.378	0.597	0.660	0.395	0.575	0.634
nob+nld+dan+pol+fin+bul+ron+slv	**0.382**	**0.598**	**0.665**	0.395	0.573	0.635

Table 7: English–Swedish word translation performance when the Swedish embeddings are projected from non Indo-European languages.

	Eng to Swe			Swe to Eng		
Embedding	p@1	p@5	p@10	p@1	p@5	p@10
Swedish Fasttext Embedding	0.501	0.686	0.745	0.525	0.722	0.780
ind	0.137	0.344	0.431	0.173	0.335	0.401
ind+fin	0.231	0.462	0.546	0.223	0.394	0.458
ind+fin+hun	0.255	0.493	0.574	0.234	0.399	**0.464**
ind+fin+hun+tur	**0.269**	0.501	**0.583**	0.235	**0.400**	**0.464**
ind+fin+hun+tur+est	0.267	**0.504**	**0.583**	**0.236**	0.395	0.459

Table 8: Sentence-level translation examples, Swedish–English in both directions. Named entities and punctuation are copied verbatim, all other tokens translated word-by-word as in Section 3.2. Only non Indo-European languages are used for projection, to simulate a low-resource scenario. Interesting parts in bold.

Source	**police** say that the **truck** driver was not drunk at the time .
Translation	**vakterna** påstå att den **vagnen** förare hade inte drucken vid den tiden .
Glossing	the-**guards** claim that the **wagon** driver had not drunken by that time .
Source	one city has no **electricity** for months .
Translation	enda stadens har inget **belysningen** för månader .
Glossing	only city's has no **ligthing** for months .
Source	**flygplatsen** är nära Osaka som är en av Japans största städer .
Translation	**fly** is near Osaka which is a of Japan greatest cities .
Reference	the **airport** is near Osaka which is one of Japan's largest cities .

On Visual Coreference Chains Resolution

Simon Dobnik and Sharid Loáiciga

CLASP
University of Gothenburg
name.lastname@gu.se

1. Introduction

"Situated" dialogue involves language and vision. An important aspect of processing situated dialogue is to resolve the reference of linguistic expressions. The challenging aspect is that descriptions are local to the current dialogue and visual context of the conversation (Clark and Wilkes-Gibbs, 1986) and that not all information is expressed linguistically as a lot of meaning can be recovered from the joint visual and dialogue attention. Co-reference resolution has been studied and modelled extensively in the textual domain where the scope of the processing co-reference is within a document. Robust co-reference resolution for dialogue systems is a very much needed task. In this paper we explore to what degree an existing textual co-reference resolution tool can be applied to visual dialogue data. The analysis of error of the co-reference system (i) demonstrates the extent to which such data differs from the written document texts where these tools apply; (ii) provides about the relation between information expressed in language and vision; and (iii) suggests further directions in which co-reference tools should be adapted for visual dialogue.

2. Related Work

Textual coreference resolution is a hard task in its own. Before current end-to-end neural systems raised the state of the art to up to 0.72 F-score in 2017, co-reference resolution success was around 0.63 F-score on the CoNLL2012 dataset. The best performing system to this day for English is that of Lee et al. (2018), who reports an F-score of up to 0.73 in the same dataset. If we compare these scores with other NLP tasks such as named entity recognition or parsing (both with more than 90% accuracy), they appear low.

Given its popularity in contexts with scarce amounts of training data, such as dialogue systems, here we use the Lee et al. (2011)' sieve-based system. For comparison, we also use Clark and Manning (2015)'s mention-pair system. Both are freely available through the Stanford CoreNLP distribution. Building on the output of a parser, they both first identify mentions and then decide if these mentions belong to the same co-referential chain, i.e, they all refer to the same entity. The first achieves this decision making through a series of filters for matching different patterns and the second with two classifiers and a scoring function to combine their outputs.

Unlike the neatly structured written text which is organised in documents, dialogue data is messy. The text is structured in turns that are pronounced by different speakers, and sentence boundaries are not clear (cf. Byron (2003) for an overview). Work on referring expressions generation (Krahmer and van Deemter, 2011; Mitchell et al., 2012; Xu et al., 2015; Lu et al., 2017), on its part, does not typically involve dialogue or the notion of co-reference chain –a central construct for co-reference resolution systems. Furthermore, co-reference resolution tools for dialogue are often custom built to the specific needs of companies or datasets (Rolih, 2018; Smith et al., 2011).

Our aim is to treat vision and language in a uniform manner. For example, (Kelleher, 2006) describes a model of attention in visual dialogue where the attention score is calculated for objects as the weighted integration of linguistic and visual attention scores which are then used in a ranked resolution of reference. (Stoia et al., 2006) proposes a similar model for the domain of route instructions. In all these models, the notion of co-reference chain is not taken into account as in the textual co-reference resolution domain.

The aim of this paper is to provide a preliminary investigationi of to what degree an existing off-the-shelf textual co-reference resolution tool can be used in the domain of the visual dialogue.

3. Data Processing

3.1 Method

The dataset We take the English subsection of the Cups corpus (Dobnik et al., 2015) which consists of two dialogues, each involving two participants, resulting in 598 turns in total. The goal of this corpus is to sample how participants would refer to things in a conversation over a visual scene. A virtual scene involving a table and cups has been designed in a 3-d modelling software and two avatars have been placed at the opposite side of this table representing the conversation participants. A third avatar who is a passive observer of the scene is standing at the side. A screenshot of the scene from each participants view is taken and furthermore some cups have been removed from each participants view but which the other participant can see (Figure 1). The participants are instructed to discuss over a computer terminal their view of the virtual world with each other in order to find the cups that each does not see. An example of the ellicited dialogues is given in example (1).

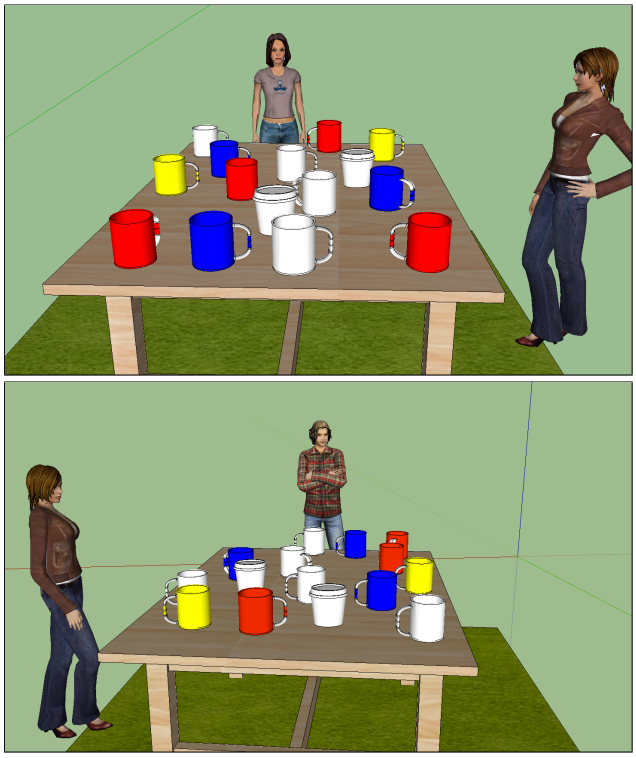


Figure 1: The table scene as seen by Participants 1 and 2 respectively.

(1) A hej
B hej
A först och frömst...
A first of all
A I see lots of cups and containers on the table
B me too
A some white, some red, some yellow, some blue
B I see six white ones
B me too
A i see seven
A but maybe we should move in one direction...
B ok, lets do that

Annotation In this pilot study two annotators annotated the first 100 turns of the GU-EN-P1 dialogue for co-reference chains as described in Pradhan et al. (2011). The annotation follows the CoNLL format with the last column containing the co-reference chains. Each chain is assigned a number id, where the first and the last tokens of a mention within the chain are identified with opening and closing brackets, as illustrated in example (2). In this example, the mentions 'lots of cups and containers' , 'some white, 'some red', 'some yellow', and 'some blue', all belong to the same chain.

This is the standard scheme used on textual data consisting of documents, but presented two challenges for our annotation: (i) in the dialogue data descriptions are made by two conversational participants from their own point of view hence pronouns 'I' and 'you' as well as spatial descriptions such as 'from my view' will have a different referent depending on the context; and (ii) a description 'the red cup' does not have a unique referent through the dialogue but this changes depending on the previous state of the dialogue and the focus on the scene. Both facts are related to our earlier observation that in visual dialogue information is not only communicated in words but also relying on joint attention.

(2)	A	1	i	(2)
	A	2	see	
	A	3	lots	(5
	A	4	of	
	A	5	cups	
	A	6	and	
	A	7	containers	5)
	A	8	on	
	A	9	the	
	A	10	table	(4)
	B	1	me	(1)
	B	2	too	
	A	1	some	(5
	A	2	white	5)
	A	3	,	
	A	4	some	(5
	A	5	red	5)
	A	6	,	
	A	7	some	(5
	A	8	yellow	5)
	A	9	,	
	A	10	some	(5
	A	11	blue	5)

Hence, the annotators also used a visual representation of the scene and descriptions were identified as belonging to the same co-reference chain only if they were referring to the same physical object. We assigned fixed ids to all existing objects in the scene (the cups and the table), as well as person A and B, 'Katie' and the table as frequently used parts of the scene such as B's-left, Katie's-right. However, dialogue participants also dynamically create 'objects' throughout the conversation that they are later referred to as normal objects, e.g. 'the empty space in front of you', 'my white ones (cups)'. For these, annotators introduced additional ids and their approximate location was marked in the representation of the scene. We expect that the challenge of this data and annotation for a textual co-reference system will be the fact the co-reference chains may be very long, e.g. 'I' and 'you' for the entire length of the dialogue. Also, the co-reference chains may be threaded as the same objects may be discussed again in another section of the dialogue. As the dialogue participants do not see exactly the same scene and they see it from a different perspective they may not be referring to the same object although they might believe so.

3.2 Results

We run the annotated data through both the sieve-based and statistical systems from the CoreNLP distribution. Both yielded the exact same output, so our analysis does not distinguish between them.

The official co-reference scorer provided with the CoNLL12 data computes the standard measures MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), CEAF

(Luo, 2005), and BLANC (Recasens and Hovy, 2010)). However, this scorer searches for complete co-reference links, and since the system was unable to find any of the gold links in our data, this oficial scorer produced appalling negative results.

A major cause behind this inability to identify the co-reference chains accurately lies on the dynamic nature of this particular type of dialogue text. For instance, the pronouns 'I' and 'me' refer to either Participant A or B, changing their reference actively as the participants use them, but the systems grouped all pronouns 'I' and 'me' into the same chain (and therefore the same entity) because they have identical forms which is one strong feature for determining co-reference in these systems. This problems affects basically all mentions that refer back to some description in a changing context such as 'my left' and 'your left'.

Concerning the parser, a central element to these systems, we observed that the sentences boundaries were identified often correctly (162 versus 157 in the gold), meaning that almost every turn in the dialogue was identified as a sentence. Some multi-word mentions such as 'a white funny top' or 'the third row from you' were also correctly analysed, suggesting further that the quality of the parser and the mention identification component was acceptable.

Looking at the mentions, however, from 293 manually annotated mentions distributed over 43 entities, the systems were not able to identify any of them correctly. On the contrary, the systems proposed 88 mentions and 28 entities.

Further investigation at the mention level reveals that a major problem was the correct identification of the mention span. For instance, in one sentence, the gold the mentions 'left' and 'red mug' were annotated, but the system identified 'her left' and 'a read mug' instead, producing a complete mistmatch. We counted only 12 mention matches due to this problem, yielding a precision of 12 / 88 = 0.14 and a recall of 12 / 293 = 0.04.

4. Conclusions

The results of our pilot study show that at least the two co-reference resolution systems tested cannot handle visual dialogue data. We expect that the created annotations will help us create a system able to simultaneously model both the language and visual components of this dataset. Current approaches to combining vision and language, e.g. (Xu et al., 2015; Lu et al., 2017) demonstrate that successful deep learning models involving vision and language can be built in the domain of static image captioning. Co-reference resolution (or generation) is a further step where such systems would be applied in a dynamic context. One difficulty that we expect for unsupervised approaches is that co-reference in visual dialogue is not directly observable in features; humans use complex mechanisms of attention to reach joint understanding. This means that a large amount of quality annotated data will be required and effectively the system will have to a learn a model of attention (cf. (Dobnik and Kelleher, 2016) for a top-down mechanistic model of attention).

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566, Granada, Spain.

Donna K Byron. 2003. Understanding referring expressions in situated language some challenges for real-world agents. In *Proceedings of the First International Workshop on Language Understanding and Agents for Real World Interaction*, pages 39–47.

Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1405–1415. Association for Computational Linguistics.

Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22(1):1–39.

Simon Dobnik and John D. Kelleher. 2016. A model for attention-driven judgements in type theory with records. In Julie Hunter, Mandy Simons, and Matthew Stone, editors, *JerSem: The 20th Workshop on the Semantics and Pragmatics of Dialogue*, volume 20, pages 25–34, New Brunswick, NJ USA, July 16–18.

Simon Dobnik, Christine Howes, and John D. Kelleher. 2015. Changing perspective: Local alignment of reference frames in dialogue. In Christine Howes and Staffan Larsson, editors, *Proceedings of goDIAL – Semdial 2015: The 19th Workshop on the Semantics and Pragmatics of Dialogue*, pages 24–32, Gothenburg, Sweden, 24–26th August.

John D Kelleher. 2006. Attention driven reference resolution in multimodal contexts. *Artificial Intelligence Review*, 25(1-2):21–35.

Emiel Krahmer and Kees van Deemter. 2011. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34. Association for Computational Linguistics.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana, June. Association for Computational Linguistics.

Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. arXiv:1612.01887 [cs.CV], 6 June.

Xiaoqiang Luo. 2005. On coreference resolution perfor-

mance metrics. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT/EMNLP 2005, pages 25–32, Vancouver, British Columbia. Association for Computational Linguistics.

Margaret Mitchell, Xufeng Han, Jesse Dodge, Alyssa Mensch, Amit Goyal, Alex Berg, Kota Yamaguchi, Tamara Berg, Karl Stratos, and Hal Daumé III. 2012. Midge: Generating image descriptions from computer vision detections. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 747–756. Association for Computational Linguistics.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27. Association for Computational Linguistics.

Marta Recasens and Eduard Hovy. 2010. Coreference resolution across corpora: languages, coding schemes, and preprocessing information. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL 2010, pages 1423–1432, Uppsala, Sweden. Association for Computational Linguistics.

Gabi Rolih. 2018. Applying coreference resolution for usage in dialog systems. Master's thesis, Department of Linguistics and Philology, Uppsala University, Uppsala, Sweden.

Cameron Smith, Nigel Crook, Simon Dobnik, Daniel Charlton, Johan Boye, Stephen Pulman, Raul Santos de la Camara, Markku Turunen, David Benyon, Jay Bradley, Björn Gambäck, Preben Hansen, Oli Mival, Nick Webb, and Marc Cavazza. 2011. Interaction strategies for an affective conversational agent. *Presence: Teleoperators and Virtual Environments*, 20(5):395–411.

Laura Stoia, Darla Magdalene Shockley, Donna K. Byron, and Eric Fosler-Lussier. 2006. Noun phrase generation for situated dialogs. In *Proceedings of the Fourth International Natural Language Generation Conference*, pages 81–88, Sydney, Australia, July. Association for Computational Linguistics.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on message understanding*, MUC-6, pages 45–52, Columbia, Maryland. Association for Computational Linguistics.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *arXiv*, 1502.03044v3 [cs.LG]:1–22, February 11.

Author Index

Keyword Index