Automatic Keyword Extraction Using Domain Knowledge

Anette Hulth[1], Jussi Karlgren[2], Anna Jonsson[3],
Henrik Boström[1], and Lars Asker[1]

[1] Dept. of Computer and Systems Sciences, Stockholm University,
Electrum 230, SE-164 40 Kista, Sweden
[hulth|henke|asker]@dsv.su.se
[2] Swedish Institute of Computer Science,
Box 1263, SE-164 29 Kista, Sweden
jussi@sics.se
[3] Department of Information Studies, University of Sheffield,
Western Bank, Sheffield, S10 2TN, UK
a.jonsson@sheffield.ac.uk

Abstract. Documents can be assigned keywords by frequency analysis of the terms found in the document text, which arguably is the primary source of knowledge about the document itself. By including a hierarchically organised domain specific thesaurus as a second knowledge source the quality of such keywords was improved considerably, as measured by match to previously manually assigned keywords.

1 Introduction

Information retrieval research has long focused on developing and refining methods for full text indexing, with the aim to improve full text retrieval. The practice of assigning keywords[1] to documents in order to either describe the content or to facilitate future retrieval, which is what human indexers do, has more or less been ignored by researchers in the various fields of computer science. However, we believe that keyword indexing — apart from being useful on its own — may play a complementary role to full text indexing. In addition, it is an interesting task for machine learning experiments due to the complexity of the activity. We extend previous work on automatic keyword assignment (see e.g., [1]) to include knowledge from a thesaurus.

In this article, we present experiments where we for each document in a collection automatically extract a list of potential keywords. This list, we envision, can be given to a human indexer, who in turn can choose the most suitable terms from the list. The experiments were conducted on a set of documents

[1] We will call a small set of terms selected to capture the content of a document *keywords. Index terms* is an alternative term we also use; the choice mostly depending on what the set of words is used for: describing the document or facilitating its retrieval.

from the Swedish Parliament which all previously have been manually indexed by professional indexers. Using machine learning algorithms and morphological pre-processing tools we combined knowledge from both the documents themselves and a hierarchically organised thesaurus developed to suit the domain, and found we were able to generate lists of potential keywords that well covered the manually extracted examples.

2 Keyword Extraction

2.1 Manual Keyword Assignment

The traditional way of organising documents and books is by sorting them physically in shelves after categories that have been predetermined. This generally works well, but finding the right balance between category generality and category specificity is difficult; the library client has to learn the categorisation scheme; quite often it is difficult to determine what category a document belongs to; and quite often a document may rightly belong to several categories.

Some of the drawbacks of categorisation can be remedied by installing an *index* to the document collection. Documents can be given several pointers using several methods and can thus be reached by any of several routes. *Indexing* is the practice of establishing correspondences between a set, possibly large and typically finite, of keywords or index terms and individual documents or sections thereof. Keywords are meant to indicate the topic or the content of the text: the set of terms is chosen to reflect the topical structure of the collection, such as it can be determined. Indexing is typically done by indexers — persons who read documents and assign keywords to them. Manual indexing is often both difficult and dull; it poses great demands on consistency from indexing session to indexing session and between different indexers. It is the sort of job which is a prime candidate for automatisation.

Automating human performance is, however, never trivial, even when the task at hand may seem repetitive and non-creative at first glance. Manual indexing is a quite complex task, and difficult to emulate by computers. Manual indexers and abstractors are not consistent, much to the astonishment of documentation researchers [2]. In fact, establishing a general purpose representation of a text's content is probably an impossible task: anticipating future uses of a document is difficult at best.

2.2 Automatic Indexing

By and large computerised indexing schemes have distanced themselves from their early goal of emulating human indexing performance to concentrating on what computers do well, namely working over large bodies of data. Where initially the main body of work in information retrieval research has been to develop methods to handle the relative poverty of data in reference databases, and title-only or abstract-only document bases, the focus has shifted to developing methods to cope with the abundance of data and dynamic nature of document databases.

Typically manual indexing schemes control the indexing process by careful instructions and an established set of allowed keywords or index terms. This naturally reduces variation, but also limits the flexibility of the resulting searches: the trade-off between predictability and flexibility becomes a key issue. The idea of limiting semantic variation to a discrete and predetermined set of well defined terms — an idea which crops up regularly in fields such as artificial intelligence or machine translation — is of course a dramatic simplification of human linguistic behaviour. This is where the most noticeable methodological shift during the past forty years can be found. Systems today typically do not take the set of index terms to be predefined, but use the material they find in the texts themselves as the starting point [3, 4].

This shift is accompanied by the shift from a set-theoretical view of document bases to a probabilistic view of retrieval: modern retrieval systems typically do not view retrieval as operations on a set of documents, with user requests as constraints on set membership, but instead rank documents for likelihood of relevance to the words or terms the reader has offered to the system, based on some probabilistic calculation.

The indexes typically generated by present-day systems are geared towards fully automatic retrieval of full texts rather than a traditional print index which will be used for access to bibliographical data or card catalogues. A traditional print index naturally must be small enough to be useful for human users. Under the assumption that no human user ever will actually read the index terms, the number of index terms can be allowed to grow practically with no limit. This makes the task of indexing texts different from the task that earlier efforts worked on.

2.3 Integrating the approaches

While the past decades have seen rapid development of full-text systems, in general, manual indexing has not been supplanted by automatic full-text retrieval systems. Manual indexing has been performed continuously all along, and recently renewed attention to the value of manual indexing has been brought to the field, by Yahoo, e.g., with its manually produced catalogue index made up of few, well-edited terms. (Experiments on automatically assigning Yahoo categories have been performed by Mladenić [5].) Manual indexing with its high quality and excellent precision will continue to have a role to fulfil in information access applications and services — but there is ample room to develop semi-automatic tools to ensure consistency and raise productivity of human indexers.

Combinations of automatic and manual approaches seem most promising. A digital library can capitalise on the qualitative work done by manual indexing to improve topical clustering of documents. If a simple topical clustering tool is available, clustering hand-categorised documents in a suitable number of topical clusters affords the possibility of using the manually assigned keywords as reasonably lucid representation of the topical clusters. Thereafter uncategorised documents can be directed automatically to the cluster nearest to them, with the clusters of higher quality, and better described — thanks to the keywords.

3 Document Representation with Thesaurus Knowledge

A thesaurus or a term database which is hierarchically organised will have valuable information for indexing purposes. The richness of the semantical relations between the included terms, which to some extent resembles the knowledge of the world of a human, is difficult to derive solely from word occurrence frequencies in documents. We will here report on experiments on bills from the 16 committees at the Swedish parliament and the thesaurus used for manual indexing of these documents.

3.1 Standard Methods: tf.idf

Arguably, the most important knowledge source for finding important descriptors for a document is the document itself. Picking the most central terms in a document can be done using *term frequency* or the tf measure: frequent terms — allowing for document length normalisation — can be assumed to be important.

A second important knowledge source about the comparative utility of descriptors is their *linguistic context*: a frequent term is only important if it frequently is infrequent. This insight can be estimated using the standard collection frequency or idf measure: calculating the proportion of documents a term participates in.

3.2 Thesaurus

The public record of parliamentary activities has a central place in the public perception of Swedish political life, and it is important that the material is accessible. The Swedish Parliament manually indexes a large number of documents in order to give access both to information specialists and to the general public. This indexing effort has been ongoing for a long period of time, during which an extensive hierarchically organised domain specific thesaurus has been developed, assuring a consistent vocabulary.

The thesaurus from the parliament, which follows the ISO 2788 standard, consists of 2 500 terms organised hierarchically by *broader term* (BT)/*narrower term* (NT) relations. Figure 1 shows an excerpt from the thesaurus: the word *arbetshandikapp* (employment disability), its broader term, some narrower terms, some related terms (RT) and a brief description of the concept the term refers to (SN – scope notes).

Arbetshandikapp	(employment disability)
BT	Arbetsliv (working life)
NT	Arbetsbiträde (working assistant)
NT	Näringshjälp (grant for resettlement in an independent business)
NT	Skyddat arbete (sheltered employment)
RT	Anställningsfrämjande åtgärder (measures to stimulate employment opportunities)
RT	Handikapp (handicap)
RT	Lönebidrag (salary contribution)
SN	Nedsatt arbetsförmåga pga fysiska, psykiska, förståndsmässiga eller socialmedicinska handikapp --- däri inbegripet missbruk av alkohol eller annat berusningsmedel. (Reduced ability to work due to physical, psychological, rational or social medical disability — including abuse of alcohol or other intoxicant.)

Fig. 1. Excerpt from the thesaurus used at the Swedish Parliament (with English equivalents).

4 Empirical Evaluation

The goal of our experiments was to automatically find all the keywords assigned by the human indexers as well as to suggest or generate further potential keywords. The decision to identify a term as a potential keyword was made on the basis of a set of features calculated for each content class word in the text. We will refer to words from the texts that actually were chosen as keywords by the human indexer as *positive* examples and the other terms as *negative* examples. By an *example* we mean one term with its feature values. Our purpose was to build a keyword identifier that would emphasise high recall with rather less regard for precision — or in other words to get false positives rather than to miss keywords assigned by the human indexers.

4.1 Experimental Set-up

For our experiments we used 128 electronic documents in Swedish: bills from the 16 different committees from the year 98/99. The style of the texts is quite homogeneous: rather formal and dense in information. The subjects, however, differ widely, being for example social welfare, foreign affairs and housing policy. The length of the bills is also quite varying: in this set ranging from 117 to 11 269 words per document, although only 26 documents have more than 1 000 words.

For all documents, the corresponding keywords, assigned by the professional indexers were available. The number of keywords per document varies between 1 and 12 in the used set, and a longer document tend to have a larger number of keywords.

In order to know in what way to best make use of the hierarchy of the thesaurus, we first inspected how the training texts were indexed manually. We found that several texts that contained a number of sibling terms — i.e. terms that shared a common broader term — were indexed with either the broader mother term, or even the yet broader grandmother term. We found — unsurprisingly — that the number of sibling terms seemed to influence the presence of the mother or grandmother term. This seemed to be a useful factor to take into consideration to find potential keywords along with the frequency data. In conclusion, the features we used for our experiment are displayed in figure 2.

Term features	**Thesaurus features**
Term frequency (tf)	Mother term; present or not (ma)
Normalised frequency (nf)	Grandmother term; present or not (gran)
Inverse document frequency (idf)	Number of siblings in document; including term itself (sib(d))
	Number of siblings in thesaurus (sib(t))
	Number of children in document; including term itself (kid(d))
	Number of children in thesaurus (kid(t))

Fig. 2. The nine chosen features from the two knowledge sources.

The words in the documents were annotated for part of speech and morphologically normalised to base form using a two-level morphological analysis tool developed by Lingsoft Oy of Helsinki. After this process, all words were in lower case, and they were all single-word terms. Form word classes were removed, as well as verbs and adjectives, leaving the nouns for further experiments. To limit mismatches in comparisons towards the thesaurus, whose items were not always in base form, but occasionally determinate or plural, both the surface form as well as the lemmatised form of the documents' words were kept and matched. For each word a set of feature values was calculated. An example of the output of this process is shown in figure 3 for two terms.

Term bistånd (aid)								
tf	**nf**	**idf**	**ma**	**gran**	**sib(d)**	**sib(t)**	**kid(d)**	**kid(t)**
6	0.0050	1/5	1	0	2	5	2	7
Term handel (trade)								
tf	**nf**	**idf**	**ma**	**gran**	**sib(d)**	**sib(t)**	**kid(d)**	**kid(t)**
3	0.0025	1/11	1	0	3	6	1	5

Fig. 3. Example of data for two terms (with English equivalents) (d = in document; t = in thesaurus).

The whole set of documents was divided into two: one set consisting of 99 texts, used for finding the hypothesis; and the rest (29 documents) for testing. Thus, the material used for testing did in no way influence the training. The division into training and test set was made arbitrarily, only taking the diversity of subjects into consideration. Because of this arbitrariness, the proportions of positive and negative examples in the two sets differ slightly. In table 1, we present some details for the training set, the test set and the whole set. As can be noted, the proportion of negative examples is very large, which is often the case in information access tasks.

Table 1. The data set in detail.

	Training set	**Test set**	**Total**
Positive ex. (no)	185	57	242
Positive ex. (%)	1.38	1.20	1.34
Negative ex. (no)	13 175	4 708	17 883
Negative ex. (%)	98.62	98.80	98.66
Total (no)	13 360	4 765	18 125

4.2 Method

Virtual Predict is a system for induction of rules from pre-classified examples [6]. It is based on recent developments within the field of machine learning, in particular inductive logic programming [7]. The system can be viewed as an upgrade of standard decision tree and rule induction systems in that it allows for more expressive hypotheses to be generated and more expressive background knowledge (i.e., logic programs) to be incorporated in the induction process. The major design goal has been to achieve this upgrade in a way so that it should still be possible to emulate the standard techniques with lower expressiveness (but also lower computational cost) within the system if desired. As a side effect, this has allowed the incorporation of several recent methods that have been developed for standard machine learning techniques into the more powerful framework of Virtual Predict.

Boosting is one of the techniques that have been incorporated. Boosting is an ensemble learning method which uses a probability distribution over the training examples. This distribution is re-adjusted on each iteration so that the learning algorithm focuses on those examples that have been incorrectly classified on previous iterations. New examples are classified according to a weighted vote of the classifiers produced on each iteration. The boosting method used in Virtual Predict is called *AdaBoost* with an optional setting (stumps only) to allow for faster induction and more compact hypotheses (see [8] for details). Another feature of Virtual Predict is that it allows instances belonging to different classes to be weighted differently. This turned out to be a very useful feature in the current study, as the data set is very unbalanced.

For the training phase we were only interested in the values of the features associated with each word, as described in section 3; contextual data such as collocation of words in a common document or in similar contexts were not taken into account.

4.3 Experimental Results

The parameter setting with the best performance was that with 200 iterations and where the positive examples were given 100 times higher weights than the negative ones. This result can be seen in figure 4 (for the 29 documents in the test set). In the result calculations, we considered only those manually assigned keywords that were single-word terms actually present in the documents.

No. positive correctly classified	54
No. positive incorrectly classified	3
No. negative correctly classified	4291
No. negative incorrectly classified (false positives)	417
Recall positive	0.9474

Fig. 4. Test result.

In figure 4 we can see that the 29 documents have 417 candidate keywords in addition to the ones that were correctly classified. Looking at each document separately, the number of new potential keywords varies between 2 and 47, with a median value of 10. In other words most documents had a reasonably low number of additional suggestions: for 24 documents in the test set this number is below 18.

The number of suggestions (including the correct ones) in percent of the total number of words in the documents ranges from 0.453% to 3.91%, the average being 1.90%. Of all the suggested words, just one meaningless word slipped through (*n1*, which is the name of a funding).

Of the three positive that we were unable to find, one was due to a shortcoming of the pre-processing program that treated the letter *p* from an abbreviation as a word, in addition to a bug in the program assigning the class that, because of the &-character in *Pharmacia & Upjohn*, also treated this word as the word *p*. (The term *Pharmacia & Upjohn* should not have been there at all, as it is a multi-word term.)

An example of the output for one document is shown in figure 5.

4.4 Evaluation

The results from these initial experiments have not yet been evaluated by indexers from the Swedish Parliament. To get an accurate judgement as to the true

Name of document	**a12**
No. of words in input to Virtual Predict	**192**
No. of keywords present in input	**5**
No. of correct keywords found	**5**
sjukgymnastik (physiotherapy)	
läkarvårdsersättning (medical treatment compensation)	
etableringsfrihet (freedom of establishment)	
företagshälsovård (occupational health care)	
arbetsmiljö (work environment)	
No. of candidate keywords (false positives)	**15**
arv (inheritance)	
arbetstagare (employee)	
arbetsgivare (employer)	
konkurrens (competition)	
sjukvård (medical care)	
sjukskrivning (reporting sick)	
primärvård (primary care)	
läkare (doctor)	
etableringsrätt (right of establishment)	
patienter (patients)	
landsting (county council)	
rehabilitering (rehabilitation)	
arbetsmiljölag (occupational safety and health act)	
läkarvårdstaxa (rates for medical treatment)	
finansiering (financing)	
No. of missed keywords present in input	**0**

Fig. 5. Example of output for one document (with English equivalents).

quality of the results this would be highly desirable, since only persons working in the field, with thorough knowledge of the domain, can tell whether specific keywords are likely to be useful. We have, however, ventured an evaluation by reading through 15 of the 29 documents in the test set to be able to compare their actual content to the corresponding derived keywords. The conclusion drawn from this is that 60 up to 80% of the suggested keywords (including the correct ones) are in fact relevant to the content of the documents. We estimated the extracted keywords to be appropriate in describing the documents, and, in some cases, even to better reflect the content than the manually indexed model keywords. This would seem to indicate the potential utility of the tool for an indexing editor.

A convenient feature of inductive logic programming algorithms for the purpose of result evaluation is the ease whereby you can study rules in the generated hypotheses. The rule applied first in the process is the one based on the most discriminating feature, which gives us the possibility of assessing which of the features is the most important in the data set. According to Virtual Predict this feature is the mother term (i.e., a hierarchical feature from the thesaurus).

4.5 A Second Run

In order to establish that a thesaurus is indeed a useful knowledge source, we made a new run with the same parameter setting and the same training set — only removing the six thesaurus features. The result of this run is presented in figure 6. As can be noted, this result supports our view that a hierarchically composed domain knowledge is crucial to a tool with the current aim.

No. positive correctly classified	37
No. positive incorrectly classified	20
No. negative correctly classified	3935
No. negative incorrectly classified (false positives)	773
Recall positive	0.6491

Fig. 6. Test result without the thesaurus features.

5 Concluding remarks

The initial results are, as mentioned earlier, very encouraging. In this stage of algorithm development, we consider recall to be the most important evaluation criterion as it reflects the potential of the current approach: it is crucial to know that none of the terms chosen by an indexer have been missed by the system. There are, however, additional issues that need to be looked into in more depth. The first, and most trivial, thing would be to remove meaningless words, e.g., single characters, before the data is processed by Virtual Predict. In addition we need to take into account abbreviations, as some words are only represented by the short form in the text, e.g., *IT*, not by the full form *informationsteknik* (information technology). For applications to real-life indexing tasks, some form of utility likelihood measure should be used in result (as in e.g., [1]).

As stated earlier, we have so far only looked at single-word terms. This means that a certain amount of both potential index terms as well as terms selected by the human indexers have been ignored. However, as Swedish is rich in compounding, this is much less of a problem than had it been for English (all words in figure 5, e.g., are single-word terms in Swedish). One could possibly start by matching terms in the thesaurus with phrases in the documents.

We also need to further investigate how to take into account the cases where the thesaurus form of a word is not its base form. Alternatives include allowing more forms of the word in the system or normalising the thesaurus. Another improvement to the extraction of potential keywords would be to recognise proper nouns, as they sometimes play an important role in this type of documents. Adding a module for name recognition seems like a straightforward way to approach this issue.

Sometimes an indexer will choose a word that is not present in a document itself, and suggesting keywords absent in the text is a challenging matter. A thesaurus will, however, most likely provide some clues to this such as when broader terms tend to be preferred to some specific terms used in the text.

The potential keywords in our experiments are not necessarily terms chosen from the thesaurus. Very rarely do human indexers go beyond the thesaurus limits — this mainly happens in the case of proper names. We did not feel the need to limit the suggestions to the thesaurus. We want to keep track of new potential words and propose their timely inclusion to the thesaurus, as well as point out terms that do not seem to be used any longer in the thesaurus. Word usage reflect the constantly changing nature of our society, and as phenomena in society vary over time so does the use of words. Keeping a thesaurus up to date is a difficult task, and is in itself a complex research issue. However, we believe that this sort of tool set can lend itself to thesaurus management tools as well as document labelling.

References

1. Turney, P.D. (2000). Learning Algorithms for Keyphrase Extraction. *Information Retrieval*, **2**(4):303–336. Kluwer Academic Publishers.
2. Earl, L.L. (1970). *Information Storage & Retrieval*, volume 6, pp. 313–334. Pergamon Press.
3. Luhn, H.P. (1957). A Statistical Approach to Mechanical Encoding and searching of Literary Information. *IBM Journal of Research and Development*, **1**:309–317.
4. Luhn, H.P. (1959). Auto-Encoding of Documents for Information Retrieval Systems. In: Boaz. M. (ed.) *Modern Trends in Documentation*, pp. 45–58. Pergamon Press, London.

5. Mladenić, D. (1998). Turning Yahoo into an Automatic Web-Page Classifier. In: Prade, H. (ed.) *13th European Conference on Artificial Intelligence ECAI 98*, pp. 473–474.
6. Boström, H. (2000). Manual for Virtual Predict 0.8, Virtual Genetics Inc.
7. Nienhuys-Cheng, S.-H., and de Wolf, R. (1997). *Foundations of Inductive Logic Programming*. LNAI 1228. Springer.
8. Freund Y., and Schapire R.E. (1996). Experiments with a new boosting algorithm. In: *Machine Learning: Proceedings of the Thirteenth International Conference*, pp. 148–156.