Comparing Manual Text Patterns and Machine Learning for Classification of E-Mails for Automatic Answering by a Government Agency

Hercules Dalianis[1], Jonas Sjöbergh[2], and Eriks Sneiders[1]

[1] Department of Computer and Systems Sciences (DSV),
Stockholm University, Forum 100, SE-164 40 Kista, Sweden
{eriks,hercules}@dsv.su.se
[2] KTH CSC,
SE-100 44 Stockholm, Sweden
jsh@kth.se

Abstract. E-mails to government institutions as well as to large companies may contain a large proportion of queries that can be answered in a uniform way. We analysed and manually annotated 4,404 e-mails from citizens to the Swedish Social Insurance Agency, and compared two methods for detecting answerable e-mails: manually-created text patterns (rule-based) and machine learning-based methods. We found that the text pattern-based method gave much higher precision at 89 percent than the machine learning-based method that gave only 63 percent precision. The recall was slightly higher (66 percent) for the machine learning-based methods than for the text patterns (47 percent). We also found that 23 percent of the total e-mail flow was processed by the automatic e-mail answering system.

Keywords: automatic e-mail answering, text pattern matching, machine learning, SVM, Naïve Bayes, E-government.

1 Introduction

Many governmental agencies and companies are today overwhelmed with e-mails with queries from citizens or customers that need an answer. Many of these e-mails are easy to reply to and do not need more advanced manual processing. The reply can even be made available on the web site of the government agency or the company. We studied the Swedish Social Insurance Agency (SSIA) (in Swedish “Försäkringskassan”[1]).

SSIA receives 350,000 e-mails per year, which are answered by 640 handling officers who also answer phone calls, use Internet chat, meet citizens and make decisions. The e-mail answering work in total corresponds to 25 full-time employees. If we could automatically answer even a fraction of these e-mails then much would be gained: citizens would obtain immediate answers and the workload of the handling officers would be reduced as they would not need to answer the most basic and monotonous e-mail queries and could focus on the more demanding ones and help citizens more effectively.

[1] http://www.forsakringskassan.se

A. Gelbukh (Ed.): CICLing 2011, Part II, LNCS 6609, pp. 234–243, 2011.

We have a joint research project with the SSIA within an E-government framework, where one of the goals is to help SSIA to answer some of these e-mails automatically. We have received 4,404 e-mails sent from citizens to the SSIA. These e-mails contain questions regarding parental benefit, housing allowance, pensions, sickness benefit, etc. Most questions are about the amount of money involved or when it will be paid to the individual, but there are also more general questions such as where one can find the correct application forms. We believe that around 20 to 30 percent of the e-mails can be answered automatically. A pattern-matching system called the *E-mail interceptor* can answer e-mails in categories similar to these automatically. We wanted to evaluate the precision and recall of the previously constructed *E-mail interceptor* system in a new domain, improve it, and compare it with standard machine learning methods.

2 Related Research

The research area of automatic e-mail answering is a rather novel research area, but work has been carried out for example by Busemann et al. [1], who constructed an automatic mail answering system for a German call centre. They used 4,777 e-mails that were manually divided into 47 categories with at least 30 e-mails in each. The average length of a document was 60 words. A number of natural language processing techniques were used to identify the core of the e-mails and to find the answer to be used for the automatic e-mail answering system. Techniques such as stemming on the e-mails using a lexicon of 100,000 stems were used to normalise the contents of the e-mails. Note that German is a highly inflected language. Shallow parsing techniques, negation detection, yes-no question detection, and wh-question detection were also used. A number of machine learning techniques were used to train the system. The best performance was given by SVM (Support Vector Machines); SVM-light obtained 56.2 percent accuracy and a top five accuracy of 78.2 percent [1]. The classification tool described in Busemann et al. [1] was included in an e-mail client where categorised messages were assigned a standard answer that could be further edited by a human.

Scheffer [2] constructed a system to reply to frequently answered questions for a German education provider (TELES European Internet Academy). Scheffer used only 528 e-mails in German for training and evaluation. 72 percent of the e-mails could be answered using the nine pre-defined standard answers. The classification of the e-mails was based on a combination of Naïve Bayes- and SVM-based classification.

Mercure is an automatic e-mail answering system developed as a research system for the customer service of a Canadian telecom company. The system is described by Lapalme and Kosseim in [3]. Lapalme and Kossiem used 1,000 e-mails in English for training and evaluation of the system. There was great variation in the complexity of the queries in these e-mails, ranging from basic factual queries to complex queries needing several sources and research before a reply could be given. Lapalme and Kosseim focused on a small topic area regarding investor relations. They tested e-mail classification with K-nearest neighbours, Naïve Bayes, and Ripper, with and without stop word removal, with and without stemming and truncation of words. The success rate was 90 percent for five categories, 80 percent for ten categories, and 67 percent for 22 categories. The Mercure system experienced difficulties with messages that covered several topics; performance measurements cover single subject messages only. It is not clear how the results of the classification were used. Apparently messages in some categories were forwarded to domain experts, and 'messages of the report category are answered by simply mailing the desired report'.

Sneiders [4] describes a text pattern-based e-mail answering system that is applied to two types of e-mail: e-mails from customers to a Latvian telecom operator, and e-mails from customers to a Swedish insurance company. The system for Latvian was semi-automatic, preparing answers for the support officers to send out, whereas the Swedish system was fully automated. Generally the e-mails to the two customer services were fairly uniform in style and thus suitable for automatic e-mail answering.

3 De-identification and Ethics

The e-mails sent to SSIA from citizens may contain sensitive information that should not be divulged outside the SSIA. Sensitive information is information that can reveal the sender of the e-mail.

Information that can identify the sender includes for example social security numbers, phone numbers, e-mail addresses, web addresses, street addresses and postal codes as well as personal names. Before SSIA handed over the e-mails to our research group the e-mails were de-identified by a de-identification program. The de-identification program was executed on 4,404 e-mails from the period from March to August 2009 and the de-identified e-mails were handed over to our research group by SSIA.

4 Data Collection: E-Mails from SSIA

Analysing 4,404 de-identified e-mails from SSIA we found that they were of varied length and complexity. Most of the e-mails were written in Swedish, which is a Germanic language with rich morphology and very productive compounding (creation of new long compound words).

Some e-mails generate very long threads with several tens of e-mails with queries and replies, sometimes up to 40 iterations. The majority, 96.2 percent, of the e-mails had only up to four threadings. The e-mail texts were no more complex than those processed in [4] but the topic diversity here was slightly larger. In [5] an experiment was carried out on clustering e-mails. E-mails were clustered both with and without the threads but also with the query and the first answer in the thread. The authors did not find any difference between using the whole e-mail with all threadings or just the query when using the K-Means algorithm.

We therefore decided to cluster the e-mails using only the query (without the threadings). A clustering process where the e-mails were clustered using the K-Means algorithm was carried out with the aim of identifying similar and relevant query groups. Eleven clusters of frequent queries were identified, hereafter called categories (see Fig. 1).

• **When will you decide my housing allowance?**	**138**
• **I want an estimate of my future pension.**	**59**
• I want to change the taxation on my pension. (To avoid tax arrears.).	39
• **When do I get the money?**	**631**
• **How many days of parental benefits remain for my child?**	**100**
• Questions concerning child allowances.	125
• **Want a form (application form or otherwise).**	**170**
• Want a beneficiary certificate (used to get discounts).	61
• Want an EU card (entitles the holder to medical care in the EU).	32
• A question in any language other than Swedish.	11
• **Miscellaneous**	**3,205**
SUMMARY	4,571[2]

Fig. 1. Eleven answering categories (with the five selected for automatic answering in bold). The number represents the number of categorised e-mails.

Of these eleven categories, five categories were selected for automatic answering (Fig. 1 in bold); questions in these categories could be answered with a short answer that included a redirection to the SSIA website, which was convenient for demonstration purposes. In these five categories several similar e-mail queries could be answered using our text pattern matching system, the E-mail interceptor (see Section 6).

One observation is that 30 percent of the e-mails (see Fig 1.) fall into one of the nine top categories that can be answered automatically (excluding the categories ‘language other than Swedish’ and ‘Miscellaneous’) and 24 percent of the e-mails fall into the categories handled by the E-mail interceptor (see Fig. 1).

5 Annotation

To make it possible to evaluate the E-mail-interceptor and to the train the machine learning systems we needed annotated e-mails. We extracted the last message of the citizen from each e-mail, i.e., stripped the text from the previous conversations, and annotated the extracted texts. Four annotators started the annotation process with annotating the same small set of e-mails containing only 100 e-mails and then met for a discussion on how the annotation should be carried out and obtained a consensus. We finally annotated a total of 4,404 e-mails in eleven classes (categories).

The 4,404 e-mails (with only queries) encompass 296,855 tokens, i.e., an average of 65 tokens per e-mail. The e-mail tokens are on average 4.5 characters long.

We used part of the annotated e-mails as a training set (2,437 e-mails) and the remaining part (1,967 e-mails) as an evaluation set.

[2] The sum is 4,571 classifications of e-mails since the 4,404 e-mails can be in more than one category.

6 The Text Pattern-Based System

Our E-mail interceptor uses a set of FAQs (Frequently Asked Questions) specifying the questions that are to be answered automatically [4]. In this paper, the FAQ are the five categories detailed in Fig. 1, Section 4. For each question in the FAQ there is a set of hand-crafted text patterns that match wordings in query e-mails. The strengths of these patterns are the following:

- the text patterns capture relevant phrases, not just a set of keywords;
- each concept in a text pattern is described by a set of synonyms, generalisations, specialisations, etc., which can be single words or phrases;
- the synonyms are narrow context-dependent, rather than general, as in synonym dictionaries;
- since text patterns do not depend on each other, e-mails containing several questions can be assigned to several categories;
- the technique has been tested for three languages.

Before the E-mail interceptor can start operating, it is 'trained' to recognise e-mail texts that fit a given standard answer. We put 'trained' in quotes because this is not training as understood in machine learning. The training e-mails, in total 2,437, were analysed by a human and the text patterns linked to each text class were created manually. Currently, there is no method or tool for creating these text patterns automatically.

There were 1,967 e-mails in the evaluation set for the E-mail interceptor. 250 e-mails matched the patterns for at least one of the five relevant text classes, and three e-mails matched two, which makes 256 emails placed into an automated answer category, and 1970 total email placements.

Table 1 (and Table 2 for graphical form) shows the number of messages in each text class, the number of messages that the E-mail interceptor placed into each text class, and the precision and recall for each class. For the five relevant classes, precision ranges from 84 to 97 percent. Recall is just above 50 percent, except for one class with 41 percent.

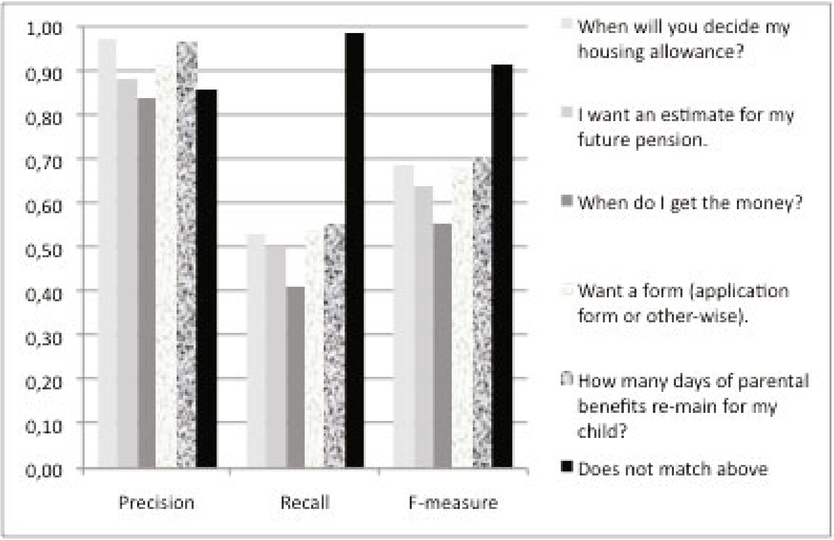
The average precision and recall for the five classes, calculated by dividing all the correctly placed messages by the total number of relevant messages for these five classes, were 89 percent and 47 percent, respectively. The reason of such low recall is a lack of opportunity to perform iterative improvement of the text patterns. That is, we did not have an opportunity to observe what mistakes the system makes on a larger test corpus and correct these mistakes in the patterns. We believe our recall values would have been higher than the current 50 percent if several 'training' and testing iterations had been performed.

We also have the category of 'Miscellaneous' e-mails, with quite high precision and recall values. In the context of automated e-mail answering, these would be the messages sent on to manual processing, and therefore the precision and recall values of this text class are not particularly interesting.

Table 1. Results from the E-mail interceptor, the text pattern-based system

No	Category	Placed in category	Placed in category, relevant	Total relevant	Precision	Recall	F-measure
1	When will you decide my housing allowance?	34	33	62	0.97	0.53	0.69
2	I want an estimate of my future pension.	17	15	30	0.88	0.50	0.63
3	When do I get the money?	132	111	269	0.84	0.41	0.55
4	Want a form (application form or otherwise).	45	41	76	0.91	0.54	0.68
5	How many days of parental benefits remain for my child?	28	27	49	0.96	0.55	0.70
	Total	256	227	486			
	Average				**0.89**	**0.47**	0.61
6	Does not match above	1714	1467	1490	0.86	0.99	0.91
	Total	1970	1694	1976			
	Average				0.86	0.86	0.86

Table 2. Bar chart table presentation of Table 3, the results from the E-mail interceptor



7 Applying Machine Learning Techniques

Apart from the handwritten pattern matching rules in the E-mail interceptor, we also applied machine learning methods for classifying the 4,404 e-mails. We used the WEKA framework [7].

We used Naïve Bayes and Support Vector Machines in WEKA in a standard text classification setting, i.e., the features for the machine learning were word vectors (TF/IDF), evaluated using tenfold cross-validation on the whole data set in Fig 1.

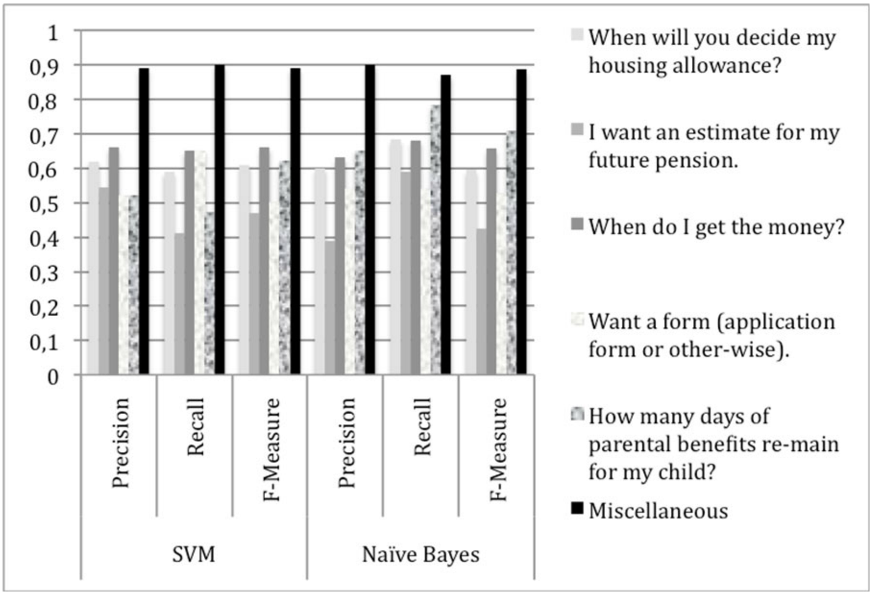
We classified the e-mails into six categories, the five used by the E-mail interceptor and one large category termed 'Miscellaneous'; see Table 3 (and Table 4 for graphical form) for the results. We used splitting of compound words into their components, lemmatisation of words, shallow parsing (chunking) of the text into phrases, and automatic spelling correction of misspelled words. Again, there was no real difference between the two machine learning methods SVM and Naïve Bayes or language technology preprocessing.

Table 3. The top five categories classified with both SVM and Naïve Bayes, using ten-fold cross-validation

		Manually	**SVM**			**Naïve Bayes**		
No	**Categories**	**Classified E-mails**	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**
1	When will you decide my housing allowance?	138	0.62	0.59	0.61	0.60	0.68	0.64
2	I want an estimate of my future pension.	59	0.55	0.41	0.47	0.39	0.59	0.47
3	When do I get the money?	631	0.66	0.65	0.66	0.63	0.68	0.65
4	Want a form (application form or other-wise).	170	0.52	0.47	0.50	0.54	0.54	0.54
5	How many days of parental benefits remain for my child?	100	0.63	0.60	0.62	0.65	0.78	0.71
	Weighted (by #mails in cate-gory) **Average**	220	0.63	0.60	0.62	0.60	0.66	0.63
6	Miscellaneous[3]	3473	0.89	0.90	0.89	0.90	0.87	0.89
	Summary	4571	0.82	0.83	0.83	0.83	0.82	0.83

[3] Compared to Fig 1 there are more e-mails in the Miscellaneous category, since all e-mails in the five unused classes are now classified as Miscellaneous.

Table 4. Graphical overview of the results of Table 3, showing the differences between SVM and Naïve Bayes using tenfold cross-validation



8 Error Analysis

One problem is that the ‘miscellaneous category’ is large compared with all other categories. This makes the machine learning methods focus on this (least interesting) category and perform poorly on the others. Other problems include questions from non-native speakers that contain very many writing mistakes, thus making it hard for the system to understand what the user means. Native speakers also make many mistakes, and there are very many ‘creative’ abbreviations that make simple word matching as well as other language processing methods difficult.

The precision and recall values do not seem to be directly connected to the number of messages in the category ‘When do I get the money?’ turned out to be a rather broad category in which the question was raised in many different contexts and different ways. Thus, the recall for the E-mail interceptor is low, at 41 percent, whereas precision is still high, at 84 percent (see Table 1).

The E-mail interceptor generally does better than the machine learning methods but there are of course also e-mails that the machine learning methods get right but where the E-mail interceptor fails. Most such examples stem from for example the SVM being much more aggressive in classifying e-mails into the smaller classes than the E-mail interceptor that is tuned for high precision on these. Another fairly typical example is an e-mail talking about how the person found the correct form online but does not have a printer so he would like to know if the form can be sent to him through normal post instead. The E-mail interceptor classified it as a request for information on finding forms online since the text is very similar to such e-mails while it is mentioning that this did not work. The SVM correctly classified it as ‘Miscellaneous’ based on there also being a lot of text not fitting any category in particular.

9 Conclusions

We ported a system for automatic e-mail answering [4] to a new domain and compared it with standard machine learning text classification methods. The method based on manually-created text patterns had very high precision, 89 percent, for the categories that would be answered automatically, but the recall was quite low at 47 percent.

With machine learning, the recall was slightly higher, from 60 to 66 percent (depending on the method and settings), but precision was much lower, 60 to 63 percent. These figures are not directly comparable since the machine learning methods are evaluated using tenfold cross-validation and the text patterns are evaluated on a different set of the e-mails as a test set. Using tenfold cross-validation favours the machine learning methods over the manually-created patterns, so that they still perform better than the machine learning methods is an even stronger result for the manual patterns.

In this application it is important that the precision is high, since otherwise the answers sent out will be answers to the wrong question and thus frustrate the users instead of helping them. A high recall is of course also important since that means fewer e-mails need to be answered manually, but without a high precision in the automatic answering the system is not useful, so recall is a secondary concern.

Annotation of the 4,404 e-mails used took about 40 hours. Training the machine learning systems took a few minutes, whereas manually constructing the patterns for the E-mail interceptor took around 40 hours. Although more work is required for the manually-constructed patterns, this method clearly outperforms the machine learning methods, both in terms of total classification accuracy, and, most importantly, its very high precision in the categories that are answered automatically.

Both methods would benefit from more annotated e-mails. The machine learning methods lack data for many categories, and the manually-created patterns would benefit from iterations of testing on new e-mails to discover and correct mistakes made by the current patterns.

One possibility of increasing the amount of training material without the need for manual annotation is to use active learning [7] In an active learning scenario the small amount of manual annotated material would hopefully boost performance both for the machine learning-based system and for the text pattern-based E-mail interceptor. To improve the performance of our approach we will also look into [8] if we can use action request classification features. A text pattern-based system is most advantageous in settings where the correctness of replies is crucial, where we want to maximise the end-user experience, and where a list of ten candidate answers is not an option, for example in fully automated e-mail answering without any human mediation. This approach is especially advantageous for e-mail flows with a high ratio of recurring inquiries.

The text patterns however have at least two limitations. First, the technique is designed for narrow and stable domains only. It should not be considered for text classification tasks in arbitrary text collections. Second, there is insufficient automation of the text pattern generation, which lessens the practical value of the technique until at least partial automation of this process is achieved.

One of the benefits of the machine learning methods is that much less manual work is needed. The e-mails do however contain very many misspellings of important words, non-standard abbreviations, grammatical mistakes of many kinds, etc. This makes their automatic processing difficult.

Acknowledgments

We would like to thank Anne-Lie Karlsson at Försäkringskassan, SSIA, for her warm support of the IMAIL research group. We would also to give a special thanks to our annotators Viggo Kann, Ola Knutsson and Magnus Rosell for their devoted work.

Finally we would like to thank VINNOVA (The Swedish Governmental Agency for Innovation Systems) for the funding of the IMAIL-project.

References

1. Busemann, S., Schmeier, S., Arens, R.G.: Message classification in the call center. In: Proceedings of the Sixth Conference on Applied Natural Language Processing, Seattle, Washington, pp. 158–165. ACL (2000)
2. Scheffer, T.: E-mail answering assistance by semi-supervised text classification. Intelligent Data Analysis 8(5), 481–493 (2004)
3. Lapalme, G., Kosseim, L.: Mercure: Towards an automatic e-mail follow-up system. IEEE Computational Intelligence Bulletin 2(1), 14–18 (2003)
4. Sneiders, E.: Automated E-mail Answering by Text Pattern Matching. In: Loftsson, H., Rögnvaldsson, E., Helgadóttir, S. (eds.) IceTAL 2010. LNCS, vol. 6233, pp. 381–392. Springer, Heidelberg (2010)
5. Dalianis, H., Rosell, M., Sneiders, E.: Clustering E-mails for the swedish social insurance agency – what part of the E-mail thread gives the best quality? In: Loftsson, H., Rögnvaldsson, E., Helgadóttir, S. (eds.) IceTAL 2010. LNCS, vol. 6233, pp. 115–120. Springer, Heidelberg (2010)
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2005)
7. Cohn, D.A., Zoubin, G., Michael, I.J.: Active learning with statistical models. Journal of Artificial Intelligence Research 4, 129–145 (1996)
8. Lampert, A., Dale, R., Paris, C.: Detecting Emails Containing Requests for Action. In: The Proceeding of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL-HLT, Los Angeles, pp. 984–992 (2010)