Using file system content to organize e-mail

Maya Sappelli
m.sappelli@cs.ru.nl

Suzan Verberne
s.verberne@cs.ru.nl

Wessel Kraaij
w.kraaij@cs.ru.nl

ABSTRACT

This paper is about using existing directory structures on the file system as models for e-mail classification. This is motivated by the aim to reduce the effort for users to organize their information flow.

Classifiers were trained on categorized documents and tested on their performance on an unstructured set of e-mail correspondence related to the documents. Even though the documents and e-mails in our corpus belonged to the same categories, the classifiers showed very low accuracy on e-mail classification. More importantly, a learning curve experiment showed that initiating a model with documents can have a negative impact on the overall accuracy that could be achieved on e-mail classification. Features important for e-mail classification are inherently different than those important for document classification.

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: Design Methodology; I.6.4 [**Model validation and analysis**]; H.1.2 [**User/Machine Systems**]: Human factors

General Terms

Design, Performance, Human Factors

Keywords

Document classification, E-mail classification

1. INTRODUCTION

In the project SWELL project we aim to support knowledge workers in their physical and mental health, by coaching them and providing them with useful tools. A knowledge worker is a person who produces and processes information in his daily work, mostly using a computer. Because of the nature of their work, knowledge workers can easily get lost in the many resources that they handle each day. They need to spend precious time at organizing information sources (e.g. e-mails, documents or people) to avoid losing important information.

Knowledge workers would be helped by solutions targeted at reducing the time necessary to find and organize information. Gomez-perez et al. [5] suggest that knowledge workers can benefit from working in context. They define context as "a set of information objects that are frequently accessed concurrently or within a very short time-span". Additionally, information objects that are similar in terms of content may belong to the same context as well. Working in context refers to a method of working that improves the ability to find and access information sources. One method to improve this ability is to associate information sources with the context in which they were produced or accessed. A computer tool can then filter out distracting information sources that belong to other contexts, such that the knowledge worker remains focused on the relevant and important sources. This method is only effective when the relations between information sources in the same context is meaningful to the worker. Additionally, the worker should not spend more time at defining contexts than at finding information himself, otherwise there would be no benefit for introducing contexts.

Organizing information sources into "context" is nothing new. One example is the organization of e-mails. Many applications for e-mail classification, such as spam-detection [12] , prioritization [1] and folder prediction [9, 10] have been described in the literature. One thing that these approaches have in common, is that they require the user to provide an initial sample of e-mails organized in the desired structure. This requires effort of the user and the structure the user comes up with may not be optimal[14]. Nevertheless, a classification algorithm needs training data. We think that the process of defining context and using this context to organize information sources should require as little effort from the user as possible. This way, the user can focus on their work tasks, while still exploiting the benefits of organized information. This requires the need for an approach that recognizes context and organizes information in a manner that is meaningful to the user but is not limited by the time and imagination that the user has available.

An intuitive source of "context" for knowledge workers is the pool of projects the user is currently working on. This would be an intuitive classification of work-related e-mail. However, most users will not consistently categorize their e-mail into projects. This is because of the nature of e-mail. Many people do not categorize their e-mail immediately but

IIiX 2012, Nijmegen, The Netherlands

let messages linger in their inbox [14].

The structure of projects a user is working on is often already used in the organization of project documents other than e-mail. This is especially the case in projects that require team work and use shared project directories. We think that it may be an interesting addition to existing work on e-mail classification to see whether these existing project structures can be exploited for the organization of e-mails into folders.

In the research presented in this paper we investigate whether we can use existing folder structures to classify unstructured e-mail data. For this purpose we train classifiers on categorized documents and test the performance of the classifiers on an unstructured set of related e-mail correspondence.

2. RELATED WORK

Organization of e-mail messages into project categories can be seen as adding context to the message. There are several approaches to recognize these contexts that lead to message sending. Shen et al. [13] present the task predictor that aims to predict a user's current activity from computer events (e.g. application activity and content) using machine learning techniques. They use Mutual Information as criterion for term selection.

Granitzer et al. [6] compare a task-centric and a user-centric scenario for supervised classification models for classifying work tasks. In the task centric approach they asked each participant to perform five typical task models (Such as planning an offical journey), which are used as labels, while collecting application names, content, window tiles and semantic type. In the user centric approach the users were free to choose their own labellings. They compared the performance of Naïve Bayes, linear Support Vector Machines and k-Nearest Neighbour classifiers. Naïve Bayes performed best for the task-centric approach, while KNN with $k = 1$ performed best for the user-centric approach.

Kellar and Watters [8] identify a user's web tasks automatically using decision trees on user-labelled data. Their results show that the effectiveness of the decision tree varies greatly over participants.

In this research we will use existing folder structures as labels for categories in a set of e-mail correspondence. This makes our problem essentially an e-mail classification or text classification problem. Many researchers have tackled this problem using statistical approaches for automatic classification of text and e-mails. Dumais et al. [4] compared Naive Bayes models, SVM models and decision trees on their performance on the Reuters21578 text collection. They conclude that linear SVM is a simple, fast and effective model for text classification. It was the most accurate classifier, followed by decision trees. The authors also concluded that the bag-of-words document representation was as least as good as more complicated representations.

Yang [15] performed another evaluation study of several classification algorithms. They indicated that KNN, neural network approaches, Widrow-Hoff inductive learning and linear least squares fit (LLSF) are valuable text classification algorithms as well.

In later research [7, 3, 9] KNN, Naive Bayes, SVM and decision trees are often used for e-mail classification. These are the classifiers that we will be using as well.

3. EXPERIMENTS

In this experiment we compare the performance of classifiers trained on documents and tested on e-mails with classifiers trained and tested on documents and classifiers trained and tested on e-mails.

3.1 Method

We collected data from a student who provided documents and e-mails corresponding to 43 courses. We only used documents that had the extension .doc, .docx, .pdf, .odt or .txt. All documents were converted to raw text before further processing. 8 pdf-documents were excluded, because their content could not be converted to text. The documents were hierarchically organized in a category structure as depicted in figure 1.

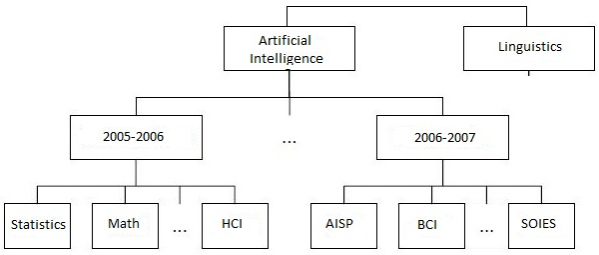

Figure 1: Example of document category structure

Although there is a hierarchical order, we considered curriculum (i.e. AI and Linguistics) and year as meta data. All documents belong to a course within the curriculum, therefore we only consider the course names as categories.

First, we manually labelled the e-mail data with labels corresponding to the course names in the category structure. This resulted in a set of 43 categories and the following distributions: 354 documents with a median of 17 per category and 874 e-mails with a median of 4 per category.

The documents were fed into the framework RapidMiner (previously Yale [11]). Since the documents contained text in both English and Dutch, stop-words of both languages were removed from the documents. A tf-idf matrix was calculated from the term frequencies in the documents. We ran three classification experiments, using multiple classifiers for comparison.

In the first experiment, the classifiers were trained on documents and tested on documents. In the second experiment, the classifiers were trained on e-mails and tested on e-mails. In the final experiment, the classifiers were trained on documents, but were tested on e-mails. The classifiers we used were SVM, Naive Bayes, Decision Tree J48, K-Nearest Neighbours (with $k = 1$ and $k = 5$, similarity measured by cosine similarity) and ZeroR (i.e. estimate of the baseline performance by majority class prediction). All classifiers were initialized with their default settings in Rapidminer. The features were pruned: words that occurred in less than 3% or more than 30% of the documents were excluded from the feature vectors. For Naïve Bayes, we did not prune the features because it is known to work better with all features, while the other classifiers work better when a subset of features is used [7]. There were 108461 features in the

Table 1: Classification Accuracy.

Classifier	Document—Document	Document—E-mail	E-mail—E-mail
ZeroR (majority baseline)	15.3%	15.3%	27.1%
Naive Bayes (no pruning)	**74.6%**	1.7%	**89.9%**
KNN, K=1	66.4%	17.4%	87.4%
KNN, K=5	66.2%	**20.9%**	86.6%
Decision Tree J48	53.0%	N/A	81.9%
Linear SVM	69.0%	7.8 %	84.7%

document set and 15812 in the e-mail set. The pruned sets consisted of 2770 and 593 features respectively. Performance of classifiers was estimated using 10-fold cross validation.

3.2 Results and Discussion

Table 1 reports the classification accuracy of the classifiers. It shows that overall Naive Bayes has the best performance in the document—document and e-mail—e-mail experiments, while KNN with $k = 5$ shows the best performance in the document—e-mail experiment. KNN is actually the only classifier that performs better than the baseline in the document—e-mail experiment.

There are several important findings in these results. First, there are big differences between the performance of the classifiers. All classifiers seem to have decent accuracy ($> 80\%$) on classifying e-mail. However, the accuracy of document classification is much lower. Especially Decision Tree does not seem to work that well for document classification in this dataset with the default parameters.

Secondly, there are big differences between the performance results when trained on homogeneous train-test sets (document—document and e-mail—e-mail) versus heterogeneous sets (document—e-mail). This is mainly due to the large number of words in the documents that do not occur in the e-mail message set, explaining why a model trained on document words might not test well on e-mail messages. This also explains why the decision tree algorithm could not determine the accuracy on the test set, as the rules were mostly not applicable. A Naïve Bayes model trained only on the intersected set of features (i.e. the set of words that occurred in both the documents and the e-mails) showed an accuracy of 10.8%, which is still below baseline accuracy, but much better than the original score of 1.7% for a model trained on all features. A KNN model with $k = 5$ trained on the intersected set showed an accuracy of 39.34% which is an improvement over the model with all features (20.9%).

An important difference between the document and the e-mail set that could influence the results is the language use. The document set contained many English documents, while the e-mail message set consisted of mostly Dutch messages. We analysed the number of Dutch, English and non-words in both sets by comparing the words to the CELEX wordlists [2]. Non-words included conversion artefacts from converting pdfs to text, spelling errors and names. Table 2 shows the number of Dutch and English words in the different sets. Most of the Dutch and English words from the e-mail messages are preserved in the intersected set of words. The number of non-words is however drastically lower than in the original set.

We manually analysed the three most important features in each of the Naïve Bayes class models trained on documents and compared those to the important features in the models trained on e-mails. Table 3 presents the percentage

Table 2: Frequency of Dutch, English and non-words

	In documents	In e-mails	In both
Dutch	17281	4632	3987
English	21587	3051	2995
Non Words	71386	8479	2116

of feature types in the models trained on documents and on e-mails. The feature types consisted of conversion artefacts (e.g. ÿÿ), course identifiers (e.g. AISP, HCI), names of persons, category related words (e.g. agents, grammar) and other words (e.g. give, why) We see that for documents, category related words are most important, while for the e-mail model names are most important.

Table 3: Feature type distributions of the most predictive features for each category

Feature type	Document Models	E-mail Models
Artefacts	30%	7%
Course identifiers	4%	7%
Person names	2%	48%
Category related words	41%	16%
Other words	23%	21%

These differences in word types in the top 3 of important features indicate that words that are good features to distinguish document classes, may not be good features for distinguishing e-mail classes. Even though most names are preserved in the set of intersected words (which are important features for e-mail classification), they only make up 2% of the features and do not seem to play an important role in document classification. This explains why a model trained on documents with features that occur in documents does not perform well on e-mail classification.

To see whether we can still benefit from a model trained on documents, we investigated how many e-mail training examples are needed to reach a classification accuracy of 80%. We updated the Naïve Bayes document model with increasingly more training examples from e-mails, while testing on the remaining set of e-mails, to estimate the learning curve. The training examples were chosen using stratified sampling. The first learning step consisted of all training data from the document set together with 5% of the e-mail training examples and 95% test examples. On each learning step, the number of e-mail training examples was increased with 5% until the training set consisted of 90% of the examples and the test set of 10%. We compared this to the learning curve of a Naïve Bayes model that was trained on e-mails from the beginning and to the learning curve of a KNN classifier ($k = 5$) that was initially trained on documents. The accuracy of each learning step was estimated using 10-fold cross validation. Figure 2 shows the results. The learning curve for e-mail data only reaches the 80% point after

having seen 20% of the examples, this is the same for the KNN model with document data. The learning curve for the Bayesian model that includes the document data needs more e-mail examples and reaches the 80% point after having seen around 40% of the examples. Using a Bayesian model trained on documents has a negative impact on the maximum accuracy that can be achieved on e-mail classification, while using a KNN model does not show any benefits over using e-mail examples only.

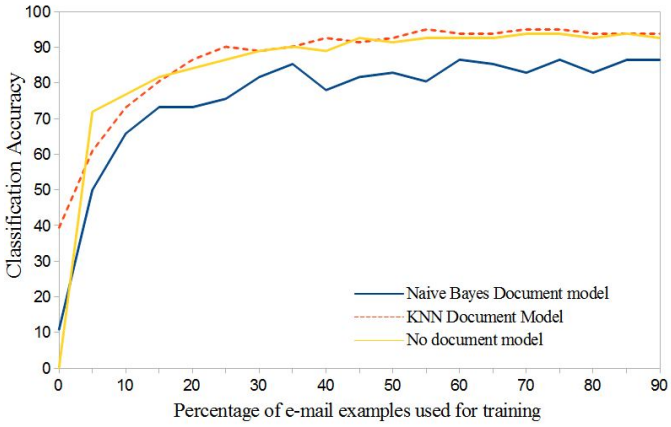

Figure 2: Learning curve for e-mail classification with models trained on documents and updated with e-mail examples compared to learning curve trained entirely on e-mail examples

4. CONCLUSION AND FUTURE WORK

In the research described in this paper we tried to exploit existing document structures for the classification of e-mail messages. The underlying idea was that these structures exist anyway and that they can provide representative meaningful categories for e-mails. This would reduce the effort required of users for e-mail classification, since users would not need to think of meaningful categories themselves and would not have to label their e-mails.

We found that training on documents leads to models that are not suitable for classifying e-mails, even though the categories are the same. This means that we can not exploit existing document structures directly for classifying e-mail. More importantly, learning curve experiments showed that using a model trained on documents as an initial training set can even have a negative impact on the overall accuracy that can be achieved in e-mail classification. We found that person names are important features for classifying e-mails. The same features are found in documents but are not important for document classification.

In future work we will investigate how combinations of supervised and unsupervised learning methods can reduce the input that is necessary from users in document organisation. Additionally, we will investigate the optimization of the balance between required user effort and received benefits. We look into the simplification of supervision by the user, and the optimization of the timing of supervision requests.

5. ACKNOWLEDGEMENTS

This publication was supported by the Dutch national program COMMIT (project P7 SWELL).

6. REFERENCES

[1] D. Aberdeen, O. Pacovsky, and A. Slater. The learning behind gmail priority inbox. In *LCCC: NIPS 2010 Workshop on Learning on Cores, Clusters and Clouds*, 2010.

[2] G. Burnage. *CELEX – A Guide for Users*. Centre for Lexical Information, University of Nijmegen, 1990.

[3] E. Crawford, J. Kay, and E. McCreath. *Automatic Induction of Rules for e-mail Classification*.

[4] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 148–155. ACM, 1998.

[5] J. Gomez-Perez, M. Grobelnik, C. Ruiz, M. Tilly, and P. Warren. Using task context to achieve effective information delivery. In *Proceedings of the 1st Workshop on Context, Information and Ontologies*, page 3, 2009.

[6] M. Granitzer, A. S. Rath, M. Kröll, C. Seifert, D. Ipsmiller, D. Devaurs, N. Weber, and S. Lindstaedt. Machine learning based work task classification. *Journal of Digital Information Management*, 7(5):306–314, 2009.

[7] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98*, pages 137–142, 1998.

[8] M. Kellar and C. Watters. Using web browser interactions to predict task. In *Proceedings of the 15th international conference on World Wide Web*, pages 843–844. ACM, 2006.

[9] B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research. In *Machine Learning: ECML 2004*, pages 217–226. Springer, 2004.

[10] G. Manco, E. Masciari, M. Ruffolo, and A. Tagarelli. Towards an adaptive mail classifier. In *Proceedings of the Italian Association for Artificial Intelligence Workshop*, 2002.

[11] I. Mierswa, M. Scholz, R. Klinkenberg, M. Wurst, and T. Euler. YALE: rapid prototyping for complex data mining tasks. In *Proceedings of the 12th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 935–940. ACM, 2006.

[12] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk e-mail. In *Learning for Text Categorization: Papers from the 1998 workshop*, volume 62, pages 98–105, 1998.

[13] J. Shen, L. Li, T. G. Dietterich, and J. L. Herlocker. A hybrid learning system for recognizing user tasks from desktop activities and email messages. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 86–92. ACM, 2006.

[14] S. Whittaker and C. Sidner. Email overload: exploring personal information management of email. pages 276–283. ACM, 1996.

[15] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1):69–90, 1999.