Multi-facet Classification of E-Mails in a Helpdesk Scenario

Thomas Beckers
Universität Duisburg-Essen
Dep. of Computer Science
47048 Duisburg, Germany
tbeckers@is.inf.uni-due.de

Ingo Frommholz
University of Glasgow
Dep. of Computing Science
G12 8QQ Glasgow, UK
ingo@dcs.gla.ac.uk

Ralf Bönning
d.velop AG
Schildarpstraße 6 – 8
48712 Gescher, Germany
ralf.boenning@d-velop.de

Abstract

Helpdesks have to manage a huge amount of support requests which are usually submitted via e-mail. In order to be assigned to experts efficiently, incoming e-mails have to be classified w. r. t. several facets, in particular topic, support type and priority. It is desirable to perform these classifications automatically. We report on experiments using Support Vector Machines and k-Nearest-Neighbours, respectively, for the given multi-facet classification task. The challenge is to define suitable features for each facet. Our results suggest that improvements can be gained for all facets, and they also reveal which features are promising for a particular facet.

1 Introduction

The impact of e-mail for business communication has grown dramatically during the last years. These e-mails have often a context in a business workflow. They may be trigger events for the start of a business process like an order request or they may be parts of knowledge intensive tasks [Abecker *et al.*, 2000] [Frommholz and Fuhr, 2006]. In this paper a case study of multi-facet e-mail classification for the helpdesk scenario of the d.velop AG is given. One major difficulty in e-mail classification research is the availability of data sets with correlations to the business workflow context. Although with the Enron data set [Klimt and Yang, 2004] a set of e-mails of a real world company is given, these e-mails have no explicitly given context in a business process.

To allow for the immediate dissemination of an incoming e-mail to an appropriate agent, it has to be classified w. r. t. the following three facets. A *topical classification* is necessary to determine what an e-mail is about and to find the right expert for it. Choosing a wrong person for a specific request results in additional waiting time for the customer. This may be crucial for high priority calls. The *type* of an e-mail is another important criterion – while actual support requests must be assigned to an expert, e-mails containing, for instance, criticism or a few words of gratitude, but no support request, may not be distributed at all in order to keep an expert from extra work. The third important facet is the *priority* of an e-mail, which is useful either for selecting the right expert (e. g. , someone who is immediately available in case of high priority) on the one hand, and for giving the associated expert a hint whether the request has to be handled immediately or not on the other hand. Service Level Agreements (SLA) exist that define response times for different priority categories.

The problem we are dealing with is thus a multi-facet classification of e-mails w .r. t. the three facets described above. While topical classification is a well-understood problem, classification w .r. t. the other two non-topical facets is a challenging and novel task.

The remainder of the paper is structured as follows. First, we discuss some related work on e-mail classification. Subsequently, we introduce the collection we are dealing with and discuss the facets in more detail. The methods and features used for multi-facet classification are presented in section 4. Section 5 shows some evaluation and discusses the results. Finally, a conclusion and an outlook on future work are given in section 6.

2 Related Work

E-mails are one of the most frequently used services of the internet. They are specified by *RFC 2822* of the Internet Engineering Task Force (IETF). E-mails can be considered as semi-structured documents. Semi-structured means that there is no full underlying data model as it is common in databases. Though, certain parts are described by a model, like the date or the content type, while other parts have no structure at all, like the body containing the actual message.

Most research focuses on the classification into an existing folder structure created by the user [Koprinska *et al.*, 2007] [Bekkerman *et al.*, 2004] [Crawford *et al.*, 2004] [Brutlag and Meek, 2000] [Rennie, 2000] [Segal and Kephart, 1999]. This folder structure is usually of topical nature, that is, a folder contains e-mails which are about the same topic. One main challenge is the continuous adding and removing of e-mails from folders. In contrast to most other classification tasks, one has to deal with dynamic classes. That is why this process is sometimes referred to as *filtering*. Koprinska et al. achieved the best results for topical folder structures. Eichler [2005] classified e-mails of a Swedish furniture retailer with regard to the classes *assortment*, *inventory* and *complaint* but only a few selected e-mails were used.

The most common application of e-mail classification in daily use is certainly the classification of unwanted e-mails (spam). Many researchers have introduced their concepts. By now it is possible to achieve an accuracy of about 90%. Blanzieri and Bryl [2006] as well as Cormack [2007] provide a survey about current techniques for the classification of spam e-mails.

Classification in regard to non-topical criteria is also possible. Cohen et al. [2004] classified e-mails according to so-called *acts of speech*. An act of speech is a pair of a verb and a noun, like *deliver information* or *request meeting*. Bennett and Carbonell [2005] tried to recognize e-mails that require an action by the user (*action items*). Their classification was performed on document and sentence level whereas classification on sentence level achieved the best results. Nenkova and Bagga [2003] analysed e-mails of a contact center if they require an immediate reply or not (*root messages* vs. *single messages*). Antoniol et al. [Antoniol *et al.*, 2008] classified texts posted in bug tracking systems – which are similar to e-mails – into different kind of activities, like *bug*, *enhancement*, *refactoring* etc.

Most research focuses on term features only and performs classification with respect to a single facet. Furthermore, only corpora in English are used. Our approach takes also non-term features into account and is evaluated with a German language corpus.

3 Collection

The d.velop AG[1] is a German software company for solutions and systems in the area of document management. In order to give their customers support on their products the support department has implemented a helpdesk solution in which customer requests are stored, tracked and routed to appropriate agents.

The d.velop AG has made a subset of support e-mails – so-called *tickets* – available for this research work. One motivation for this step is the question whether the helpdesk process can be made more efficient with the support of automatic classification. Furthermore, the d.velop AG has developed commercial solutions for managing e-mails and aims at improving their products based on research work.

Our collection consists of 2000 e-mails that were received from October 2007 to May 2008 by the support system at d.velop AG. A multi-facet classification was performed by the employees of the helpdesk which we used for training and testing of classifiers.

Every incoming e-mail is stored in two files by the support system. One file contains the actual content while the other file contains metadata. The metadata includes the classifications in respect to the three facets which were performed by the employees of the helpdesk. This classfication is used for the training of the classifiers. In order to handle the data more easily we created a single XML file which comprises all relevant data. Furthermore, we cleaned up the classes and deleted tickets internal to d.velop. Some rare classes were also removed while some other classes were merged.

In the following the classes of each facet are described.

Topic Each product or service forms a class. A class consists of three parts, namely the *product*, the *module* and the *component*. The *module* and the *component* part can also be empty. Thus, the classes build several hierarchies. Figure 1 illustrates an extract of the class hierarchies.

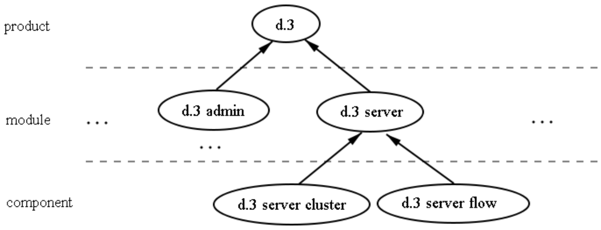


Figure 1: Extract of the class hierarchies of the facet *topic*

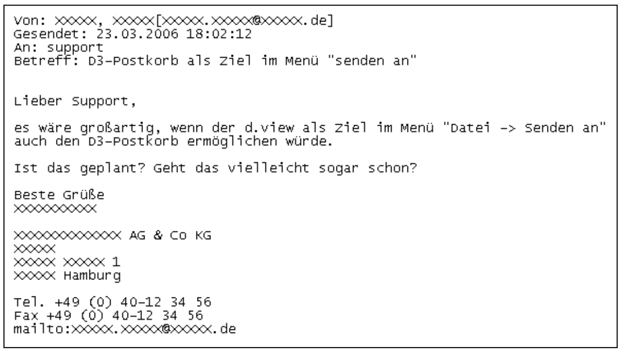
Some classes occur quite often, such as *d.3 explorer*, while others occur very rarely, like *d.3 admin ldap*. 31 classes remained after cleanup.

Support Type The classes of the support type are

- *error report*, if an error occurs in the software (merged from 2 very similar classes);
- *questions concerning handling/config/doc*, if there are questions about the handling, configuration or documentation of the software;
- *consultation*, if a customer requests consultation about products or services and
- *miscellaneous*, for all other possible types of requests.

Three other classes were ignored since they were rarely used. About 70% of the tickets belong to the class *error report*. Note that a classification as error does not make a statement about the severity or if it is a product error at all.

Priority A ticket can be classified as *urgent* if it requires an immediate response, like an error report about a problem that crashes a complete system. Otherwise, a ticket is classified as *not urgent*. Both classes comprise two of the original four priority classes. Most tickets are *not urgent*.



```
Von: XXXXX, XXXXX[XXXXX.XXXXX@XXXXX.de]
Gesendet: 23.03.2006 18:02:12
An: support
Betreff: D3-Postkorb als Ziel im Menü "senden an"


Lieber Support,

es wäre großartig, wenn der d.view als Ziel im Menü "Datei -> Senden an"
auch den D3-Postkorb ermöglichen würde.

Ist das geplant? Geht das vielleicht sogar schon?

Beste Grüße
XXXXXXXXXX

XXXXXXXXXXXXX AG & Co KG
XXXXX
XXXXX XXXXX 1
XXXXX Hamburg

Tel. +49 (0) 40-12 34 56
Fax +49 (0) 40-12 34 56
mailto:XXXXX.XXXXX@XXXXX.de
```

Figure 2: Example of a ticket that was received via e-mail

Fig. 2 shows an example of a ticket[2] received via e-mail. A customer asks if it is possible to add a new shortcut to a menu. With regard to the facets this ticket is classified as

- *d.view* (facet *topic*),
- *question concerning handling/config/doc* (facet *support type*) and
- *not urgent* (facet *priority*).

[1] http://www.d-velop.de/en/

[2] Some details were made irrecognisable because of privacy reasons. Telephone numbers are fictitious.

4 Multi-facet Classification

Classification was performed with Support Vector Machines (SVMs). Due to the fact that most text classification problems are linearly separable [Joachims, 1998] a linear kernel was employed. SVMs are used with a classic representation of documents, but including also non-term features besides term features. That is why this technique is called *extended indexing* in the following.

Alternatively, we utilised k-Nearest-Neighbour (k-NN) as classification techniques. For k-NN we made use of a *probabilistic, decision-oriented indexing* technique developed by Fuhr and Buckley [1991]. Features of terms and documents (tickets) x are defined as

$$\vec{x}(t,d) = (x_1(t,d), x_2(t,d), \ldots, x_3(t,d)),$$

whereas t denotes a term and d denotes a document. For example, $\vec{x}(t,d)$ could be defined as

$$\begin{aligned} x_1(t,d) &= \begin{cases} 1 & \text{if } t \text{ in } d \text{ occurs once} \\ 2 & \text{if } t \text{ in } d \text{ occurs at least twice} \end{cases} \\ x_2(t,d) &= idf(t) \\ x_3(t,d) &= \begin{cases} 1 & \text{if } t \text{ occurs in the subject of } d \\ 0 & \text{else} \end{cases}, \end{aligned}$$

with $idf(t)$ as the inverse document frequency of t. These features are used to learn an indexing function which estimates $P(R|\vec{x}(t,d))$ based on a learning sample L^x. Beckers [2008] shows in detail how L^x is constructed. This probability is then used as indexing weight for term t in document d. The terms of the tickets and their weights are used as features for classification with k-NN. Logistic regression based on a maximum-likelihood criterion was employed to learn the indexing function. Our approach is similar to that of Gövert et al. [1999], who classified web documents of Yahoo's web catalogue.

After the representations of the tickets have been created, normalisation of the data was applied. SVMs as well as k-NN require normalisation since features with large values would otherwise overlie features with small values. The preparation of the collection and the features are outlined in the following.

4.1 Features

We regard term features as well as non-term features for classification. We defined features which seem to be useful for our task. Features and groups of features, respectively, are divided into feature categories. For each of our three facets all features are regarded. We defined the following feature categories for the extended indexing.

Terms The most obvious features are the terms which appear in the tickets. They can be either represented as sets of words or the frequency of their occurrence can be regarded. Another possibility is not to consider all terms but only terms from a dictionary (*special terms*). N-grams are usually used to take the context of terms into account. We only use bigrams because n-grams with $n > 2$ excessively increase the dimensionality of the data. Thus, most n-grams would only occur rarely or once. Finally, there are some statistics features; the count of the number of terms and the number of different terms.

Term position Not only the terms can provide meaningful features for classification. A term can appear in certain fields, like the subject or the attachment and at different places of the body. Thus, the body is divided into three thirds. Also, a simple recognition of interrogative sentences is performed. A suffix representing the position is appended to each term. These terms plus suffix are used as features.

Punctuation The usage of punctuation may also be useful for classification [Nenkova and Bagga, 2003]. Consider the following sentence of a ticket: "This does not work!". An exclamation mark may be used more often in problem reports than in questions concerning the documentation. Thus, there are features about the usage (number, relative number and if there are three in a row) of exclamation and question marks.

Attachment The attachment is used to create features as well. The actual content of the attached files is ignored since it is nearly impossible to extract data from all possible attachment types. If there are attachments and the types thereof are regarded. There are binary features for each of the following file types:

- log files (*.log)
- text files (*.txt)
- XML files (*.xml)
- temporary files (*.tmp, *.temp)
- images (*.jpg, *.png, *.bmp, *.tif, *.gif, ...)
- archives (*.zip, *.jar, *.rar)
- miscellaneous

Sender The sender address is used as feature in its entirety. The domain part of the address is used as another feature. There is also some historical information about the classes of past tickets.

Length The length of the subject and the length of the body are two more features. Note that these both features count the character length while the length feature from the feature category terms counts the terms.

Time The date and the time of an incoming ticket is also of potential value for classification. We use several features of time. There are 7 binary features that indicate the day of the week. The 24 hours of a day are divided into several blocks. For each of these blocks there is also a binary feature. Finally, a binary feature shows if the time is during usual labour time.

Characters Problem reports often have inlined snippets of e. g. log files or error messages which contain many special characters. Some special characters that we regard are e. g. the following:

`- [ ] ( ) { } : _ + = # * $ & % / \ ~ | @`

An overview about the features described above can be found in tables 10 and 11 in the appendix.

The probabilistic, decision-oriented indexing requires different definitions of term features and thus different defined feature categories (see sec. 4). All other non term-related feature categories stay the same. These features are used to learn an indexing function with logistic regression. The indexing function is then used to compute the weights for the terms of the tickets.

Terms The term frequency and the inverse document frequency of a term build this feature category as well as a binary feature that checks if a terms belongs to the most frequent terms. The statistics-related features are defined as stated above.

Term position All features from this feature category are defined along the lines of the feature category of the extended indexing but they corresponded to terms instead of tickets.

A more detailed description of the features as well as additional examples are provided by Beckers [2008].

5 Evaluation

We used the classic 10-fold stratified cross validation for testing of our concepts. Our main questions are:

- Is it possible to classify with higher quality compared to a given baseline?
- How do the different approaches (ext. indexing & SVM and prob. indexing and k-NN) perform?
- Which features are appropriate for the different facets? We think that not only set of words should be regarded as features, especially for non-topical facets.

In the following, we describe the implementation, the creation of the training/test collection, the selection of appropriate evaluation measures and finally the achieved results.

5.1 Implementation

Our experiments were performed with the open-source data mining framework *RapidMiner*[3]. Additional functionality has been implemented by means of *RapidMiner*'s plug-in mechanism. For classification with SVMs we selected the LibSVM operator which wraps the well-known LIBSVM[4] library. A built-in operator was used for k-NN. We applied *RapidMiner* in version 4.2. All experiments ran on a computer with AMD Phenom 9550 2.2 GHz, 4 GB RAM, Debian 2.6.18.gfsg-1-22 (Xen) and Java JDK 6 Update 7 64 bit.

5.2 Training and Test Collection per Facet

The complete collection consists of 2000 tickets. The maximum number of available tickets is used for the facet *topic*. Tickets with rare classes were removed, that is, classes that only occur less than 15 times. This results in 1643 tickets usable for classification. Due to time constraints only 1000 arbitrary tickets were selected for the facet *support type*. As there are only four different classes a smaller number of tickets is likely to be sufficient. Because of the poor quality of the classification for the facet *priority* we used 150 manually selected tickets for each class.

5.3 Evaluation Measures

Due to the different nature of the facets, a single evaluation measure for all facets would not have been appropriate. It seems reasonable to define a cost-based measure which takes the hierarchical structure of the classes from the facet *topic* in to account. $cost_i^j$ denotes the costs of classifying an instance of class c_j as instance of class c_i. The classification costs are defined as

$$cost = \frac{1}{|Test|} \cdot \sum_{t \in Test} cost_{predictedClass(t)}^{actualClass(t)},$$

whereas $actualClass(t)$ and $predictedClass(t)$ denote the index of the actual and the predicted class, respectively. $Test$ denotes the set of test instances.

	costs
$c_{predicted}$ equals c_{actual}	0
$c_{predicted}$ is ancestor of $c_{actual}(2 \rightarrow 3)$	0.3
$c_{predicted}$ is ancestor of $c_{actual}(1 \rightarrow 2)$	0.3
$c_{predicted}$ is ancestor of $c_{actual}(1 \rightarrow 3)$	0.7
c_{actual} is ancestor of $c_{predicted}(2 \rightarrow 3)$	0.3
c_{actual} is ancestor of $c_{predicted}(1 \rightarrow 2)$	0.3
c_{actual} is ancestor of $c_{predicted}(1 \rightarrow 3)$	0.7
$c_{predicted}$ and c_{actual} are siblings (3)	0.3
$c_{predicted}$ and c_{actual} are siblings (2)	0.7
otherwise	1

Table 1: The classification costs of a ticket of class c_{actual} that was predicted as class $c_{predicted}$ for the *topic* facet

Table 1 shows the costs that are heuristically defined based on experience. The numbers in brackets denote the levels in the hierarchy. A correct classification does not produce any costs while a classification that is completely wrong produces maximum costs. If a predicted class is an ancestor or a sibling the costs are somewhere in between.

A ticket of class *d.view* (see fig. 2) which gets classified as *d.view admin* would cause costs of 0.3.

For the facet *support type* the well-known accuracy was used [Sebastiani, 2002]. There's no evidence that it makes sense to assign different costs for wrongly classified tickets. The accuracy a is defined as

$$a = \frac{TP + TN}{TP + TN + FP + FN},$$

with the usual definition of TP (true positive), FP (false positive), TN (true negative) and FN (false negative).

Cost matrix for priority		**predicted class**	
		urgent	not urgent
actual class	urgent	0	2
	not urgent	1	0

Table 2: Cost matrix for facet *priority*

The facet *priority* also uses a cost-based measure (see table 2). It is worse to classify an urgent ticket as not urgent than classifying a not urgent ticket as urgent. It is important that the processing of urgent

[3] `http://sourceforge.net/projects/yale/`

[4] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

tickets is not delayed by an erroneous classification. That is why the costs are twice as much for the former case as for the latter case.

5.4 Results

First, we examined the term features including term modification (stemming, stop word removal, removal of rare terms) in detail. If feature categories have shown to be useful they were kept for the following experiments. Afterwards, the non-term features were analyzed. We performed an experiment that covered all non-term feature categories and then for each non-term feature category an experiment without it was performed in order to determine its usefulness. Finally, all useful term and non-term features were regarded. The optimal parameters for SVM and k-NN were estimated after each step, namely C and k, respectively. To handle multi-class problems with SVMs we utilised the one vs. one approach [Hsu and Lin, 2002]. Beckers [2008] provides a more detailed description of the results and computed additional evaluation measures for the best results.

Baselines

We regarded two different baselines, namely a random classification and a classification taking the most common class (mode). Table 3 shows the baselines for each facet. Note that costs are the better the smaller they are while the accuracy should be as high as possible. Based on these baselines we perform t-tests with $\alpha = 0.05$ (☆) and $\alpha = 0.01$ (★). There is no mode baseline for the *priority* (see sec. 5.2).

	mode	random	measure
topic	0.7117	0.8613	costs
support type	0.7179	0.2393 0.2419[1]	accuracy
priority	–	0.7334	costs

[1] All instances were weighted inversely proportional with the occurrence frequency of their class.

Table 3: Results of the baseline experiments

Facet *Topic*

Ext. Indexing & SVM Table 4 shows the results of the experiments for this facet. The best result (0.3954) of SVMs was achieved by applying simple term features with binary weights (set of words) and term modification (printed in bold font). Only special terms as feature also achieved good results (row 4) but with a lower dimensionality of the data and thus with increased performance. So, if a slight decrease of classification quality is acceptable, then a significant faster learning of classifiers is possible. Bigrams apparently were not appropriate. The term position features were also of some value for classification. All non-term features except sender and character features provided useful information for classification. Using both term and non-term features could not increase the classification quality. All results are statistically significant w. r. t. both baselines. As expected term features are the most useful features. Non-term features decreased the costs below the baselines but they could not improve the overall classification quality.

experiment	costs	SM[1]	SR[2]
terms (binary)	0.4212	★	★
terms (binary & mod.)	**0.3954**	★	★
terms (tf)	0.5082	★	★
terms (special terms)	0.4154	★	★
terms (bigrams)	0.5424	★	★
terms	0.3957	★	★
term position	0.4454	★	★
all non-term features	0.6359	★	★
without punctuation	0.6362	★	★
without attachment	0.6779	★	★
without sender	0.6252	★	★
without length	0.636	★	★
without time	0.6363	★	★
without characters	0.6357	★	★
all	0.3991	★	★

[1] significance compared to the mode baseline
[2] significance compared to the random baseline

Table 4: Results of experiments for the facet *topic* (SVM & ext. indexing)

Prob. Indexing & k-NN The use of weights from a learned indexing function for k-NN showed better results than the use of simple binary occurrence weights (see tab. 5). Due to performance reasons and time constraints the sender feature category was ignored and only a single experiment with different features than set of words was performed (as for all other facets). The best result is slightly better than the best result of the ext. indexing with SVM. All results are also statistically significant in respect of both baselines.

experiment	costs	SM	SR
binary weights	0.5562	★	★
binary weights & term mod.	0.5221	★	★
weights by ind. func.	**0.3909**	★	★

Table 5: Results of experiments for the facet *topic* (k-NN & prob. indexing)

Facet *Support Type*

Ext. Indexing & SVM The best result for SVMs were delivered by the term position features (0.7556). Table 6 shows all results. Term features with term modification, tf weights or bigrams worsened the accuracy in comparison to simple binary occurrence weights. Due to the skew class distribution we applied an equal weighting technique to avoid useless results (see [Beckers, 2008] for more details). Attachment features and time features of the non-term features had not proven as useful whereas the other non-term features (punctuation, sender, length, characters) are of value for classification. Most results are statistically significant while a few are only weak or not statistically significant compared to the baselines. In contrast to the facet *topic* not binary term features but term position features have achieved the best result. This supports our hypothesis that also other features should be taken into account for non-topical facets.

experiment	acc.	SM	SR
terms (binary)	0.7393		★
terms (binary & mod.)	0.7240		★
terms (tf)	0.7321	☆	★
terms (bigrams)	0.7199		★
terms	0.7403	☆	★
term position	**0.7556**	★	★
all non-term features	0.2904	★	☆
without punctuation	0.2655	★	
without attachment	0.3065	★	★
without sender	0.2712	★	
without length	0.2730	★	
without time	0.3002	★	★
without characters	0.2774	★	☆
all	**0.7556**	★	★

Table 6: Results of experiments for the facet *support type* (SVM & ext. indexing)

Prob. Indexing & k-NN The usage of weights by a learned indexing function achieved the best results for k-NN (see tab. 7). Term modification also increased the accuracy. Again, all results are statistically significant. Overall, the best result is slightly worse than the best result of the ext. indexing & SVM (0.7271 vs. 0.7556).

experiment	acc.	SM	SR
binary weights	0.72398		★
binary weights & term mod.	0.72403		★
weights by ind. func.	**0.7271**		★

Table 7: Results of experiments for the facet *support type* (k-NN & prob. indexing)

Facet *Priority*

Ext. Indexing & SVM The results of the experiments with SVMs are shown in table 8. The best result with term features only was achieved by terms with binary occurrence weights and term statistics features. As seen before for the other facets, tf weights and bigrams could not increase the classification quality. All non-term features except character features improved the classification quality. The usage of all available features resulted in the lowest costs (0.3967). Most results are statistically significant. Non-term features were able to increase the classification quality together with term features.

Prob. Indexing & k-NN The best accuracy was again achieved with term weights by a learned indexing function. Even the best result of ext. indexing with SVM is outperformed. Term modification was also useful. All results are statistically significant.

5.5 Discussion

Results that are statistically significant better than the baselines can be achieved for all of the three facets. In the following, some other observations we made are described.

experiment	costs	SR
terms (binary)	0.4033	★
terms (binary & mod.)	0.44	★
terms (tf)	0.49	★
terms (special terms)	0.6634	
terms (bigrams)	0.5567	★
terms	**0.3967**	★
term position	0.4167	★
all non-term features	0.4567	★
without punctuation	0.48	★
without attachment	0.56	★
without sender	0.4933	★
without length	0.49	★
without time	0.5067	★
without characters	0.4567	★
all	**0.3833**	★

Table 8: Results of experiments for the facet *priority* (SVM & ext. indexing)

experiment	costs	SR
binary weights	0.4003	★
binary weights & term mod.	0.3475	★
weights by ind. func.	**0.2997**	★

Table 9: Results of experiments for the facet *priority* (k-NN & prob. indexing)

- The estimation of the parameters for SVMs is a very time-consuming task. Some experiments ran several days; in particular, the *topic* facet with 31 classes. Due to the one vs. one classification approach $\frac{k\cdot(k-1)}{2} = \frac{31\cdot(31-1)}{2} = 465$ classifiers had to be learned for a single classification model. The learning of an indexing function with logistic regression took also some days.
- The best results for the facets were achieved by different sets of features. We have shown that it is reasonable to regard also other types of features than just simple set of words. This is in particular the case if classification is performed with respect to a non-topical facet. For the facet *topic* classic sets of words have been the best features.
- Bigrams and tf weights were not useful in any facet. This can be explained due to the fact that bigrams increase the dimensionality of the data. Thus, many bigrams only appear once or twice in the whole collection. Our experiments support that tf as weighting schema has proved to be important for information retrieval but for text classification no such statement can be done.
- Both extended indexing & SVMs and probabilistic, decision-oriented indexing & k-NN have produced results which are statistically significant better than the corresponding baselines. The differences between both techniques were higher for non-topical facets than for the topical facet.

6 Conclusion and Outlook

In comparison to other classification problems it is more difficult to achieve good classification results for the given task. For one thing the quality of the existing classification is rather poor for some facets, especially *priority*. For another thing the difference between two arbitrary classes is not as distinct as e. g. between *spam* and *no spam* in spam classification. Nonetheless, for all facets statistically significant results above the baselines have been achieved.

Extended indexing & SVMs as well as prob. indexing & k-NN have both shown good results. Thus, no conclusion about what technique is generally better for our task can be drawn.

Additional facets, such as e. g. *speech act* or *sentiment*, can be considered. However, our collection does not contain data that is required for these facets. Frommholz and Fuhr [2006] outline some more possible facets. The increasing spreading of d.velop products in other countries than Germany poses new challenges concerning multilingual tickets. The language of a ticket could also be meaningful for classification.

Further improvements could be made with learning techniques that take the classification costs into account during the learning phase (*cost based learning*). Furthermore, feature selection and weighting could increase the classification quality as well as the (time) performance. A more comprehensive evaluation should not only take the multi-facet classification in an isolated way into account but should also investigate whether the multi-facet classification is actually meaningful for employees of the helpdesk and supports them in their daily work.

References

[Abecker *et al.*, 2000] Andreas Abecker, Ansgar Bernardi, Knut Hinkelmann, Otto Kühn, and Michael Sintek. Context-aware, proactive delivery of task-specific knowledge: The KnowMore project. *International Journal on Information System Frontiers (ISF)*, 2((3/4)):139–162, 2000.

[Antoniol *et al.*, 2008] Giuliano Antoniol, Kamel Ayari, Massimiliano Di Penta, Foutse Khomh, and Yann-Gaël Guéhéneuc. Is it a bug or an enhancement?: a text-based approach to classify change requests. Proceedings of CASCON 2008, pages 304–318, New York, NY, USA, 2008. ACM.

[Beckers, 2008] Thomas Beckers. Multifacettenklassifikation von E-Mails im Helpdesk-Szenario. Diploma thesis, Universität Duisburg-Essen, 2008. In German.

[Bekkerman *et al.*, 2004] Ron Bekkerman, Andrew McCallum, and Gary Huang. Automatic categorization of email into folders: Benchmark experiments on Enron and SRI corpora. Technical Report IR-418, University of Massachusetts, CIIR, 2004.

[Bennett and Carbonell, 2005] Paul N. Bennett and Jaime Carbonell. Detecting action-items in e-mail. In *Proceedings of SIGIR 2005*, pages 585–586, Salvador, Brasilien, August 2005. ACM.

[Blanzieri and Bryl, 2006] Enrico Blanzieri and Anton Bryl. A survey of anti-spam techniques. Technical Report DIT-06-056, University of Trento, September 2006.

[Brutlag and Meek, 2000] Jake D. Brutlag and Christopher Meek. Challenges of the email domain for text classification. In Pat Langley, editor, *Proceedings of ICML 2000*, pages 103–110. Morgen Kaufmann, 2000.

[Cohen *et al.*, 2004] William W. Cohen, Vitor R. Carvalho, and Tom M. Mitchell. Learning to classify email into speech acts. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 309–316, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[Cormack, 2007] Gordon V. Cormack. Email spam filtering: A systematic review. *Found. Trends Inf. Retr.*, 1(4):335–455, 2007.

[Crawford *et al.*, 2004] Elisabeth Crawford, Irena Koprinska, and Jon Patrick. Phrases and feature selection in e-mail classification. In *Proceedings of the 9th Australasian Document Computing Symposium*, Melbourne, Australia, December 2004.

[Eichler, 2005] Kathrin Eichler. Automatic classification of swedish email messages. Bachelor thesis, Eberhard-Karls-Universität Tübingen, 2005.

[Frommholz and Fuhr, 2006] Ingo Frommholz and Norbert Fuhr. KI-Methoden zur Email-Archivierung – Technologische Studie zum Inbox-Szenario. Internal Report, Universität Duisburg-Essen, November 2006. `http://www.is.inf.uni-due.de/bib/pdf/ir/Frommholz_Fuhr:06b.pdf`. In German.

[Fuhr and Buckley, 1991] Norbert Fuhr and Chris Buckley. A probabilistic learning approach for document indexing. *ACM Transactions on Information Systems*, 9(3):223–248, 1991.

[Gövert *et al.*, 1999] Norbert Gövert, Mounia Lalmas, and Norbert Fuhr. A probabilistic description-oriented approach for categorising web documents. In Susan Gauch and Il-Yeol Soong, editors, *Proceedings of the Eighth International Conference on Information and Knowledge Management*, pages 475–482. ACM, 1999.

[Hsu and Lin, 2002] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, March 2002.

[Joachims, 1998] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. Forschungsbericht LS-8 Report 23, Universität Dortmund, 1998.

[Klimt and Yang, 2004] Bryan Klimt and Yiming Yang. The Enron corpus: A new dataset for email classification research. In J. G. Carbonell and J. Siekmann, editors, *Proc. of ECML 2004*, volume 3201/2004 of *Lecture Notes in A. I.*, pages 217–226, Pisa, Italy, September 2004. Springer.

[Koprinska *et al.*, 2007] Irena Koprinska, Josiah Poon, James Clark, and Jason Chan. Learning to classify e-mail. *Information Sciences*, 177(10):2167–2187, May 2007.

[Nenkova and Bagga, 2003] Ani Nenkova and Amit Bagga. Email classification for contact centers. In *SAC '03: Proceedings of the 2003 ACM Symposium on Applied Computing*, pages 789–792, New York, NY, USA, 2003. ACM Press.

[Rennie, 2000] Jason D. M. Rennie. ifile: An application of machine learning to email filtering. In Marko Grobelnik, Dunja Mladenic, and Natasa Milic-Frayling, editors, *Proc. of the KDD-2000 Workshop on Text Mining*, Boston, USA, August 2000.

[Sebastiani, 2002] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[Segal and Kephart, 1999] Richard B. Segal and Jeffrey O. Kephart. Mailcat: an intelligent assistant for organizing e-mail. In *AGENTS '99: Proceedings of the third annual conference on Autonomous Agents*, pages 276–282, New York, NY, USA, 1999. ACM Press.

A Tables of features

Feature group	#Features	Description
TERMS	~	All terms of a ticket
SPECIAL_TERMS	~	All special terms with a postfix
SPECIAL_TERMS_COUNT	1	number of special terms
BIGRAMS	~	Bigrams from body and subject
DIFFERENT_TERMS	2	Number of different terms and their relative number in comp. to all terms
TERMS_COUNT	3	Number of terms everywhere, number of terms in body and subject
NO_CHARACTER_OR_DIGIT	2	Number and relative number of special characters
TERMS_IN_FIRST_THIRD	~	All terms in first third of the body with postfix
TERMS_IN_SECOND_THIRD	~	All terms in second third of the body with postfix
TERMS_IN_THIRD_THIRD	~	All terms in third third of the body with postfix
TERMS_IN_SUBJECT	~	All terms in subject with postfix
TERMS_IN_ATTACHMENT	~	All terms in attachment with postfix
TERMS_IN_QUESTIONS	~	All terms in questions with postfix
QUESTION_MARKS	2	Number and relative number of question marks
THREE_QUESTION_MARKS	1	If *???* occurs
EXCLAMATION_MARKS	2	Number and relative number of exclamation marks
THREE_EXCLAMATION_MARKS	1	If *!!!* occurs
HAS_ATTACHMENT	1	If there's an attachment
ATTACHMENT_TYPE	7	type of the attachment (log, text, xml, tmp, image, archive or misc.)
FROM	~	The sender of the ticket
FROM_COMPANY	~	The domain part of the sender
FROM_HISTORY	~	The last few classes of tickets from the sender
FROM_COMPANY_HISTORY	~	The last few classes of tickets from the sender (domain part)
SUBJECT_LENGTH	1	Number of characters in subject
BODY_LENGTH	1	Number of characters in body
DAY_OF_WEEK	7	The weekday of the ticket
WORKING_HOURS	1	if the time of the ticket is during usual working times
TIME_BLOCKS	~	Time block of the time of the ticket

Table 10: Table of features (ext. indexing)

Feature group	#Features	Description
termFrequency	1	Frequency of a term in a ticket
inverseDocumentFrequency	1	*idf* of a term
mostFrequentTerms	1	If a term belongs to the most common terms
...	...	...
termInSubject	1	If a terms occurs in the subject
termInAttachment	1	If a term occurs in the attachment
termPositionInBody	3	Three features to indicate where (three thirds) a term appears
termInQuestion	1	If a term occurs in a question

Table 11: Table of features (prob. indexing)

Note: ~ denotes a variable number of features, because the concrete number depends on the number of terms in a ticket or on other properties of a ticket.