E-mail categorization using partially related training examples

Maya Sappelli
TNO and Radboud University Nijmegen
m.sappelli@cs.ru.nl

Suzan Verberne
Radboud University Nijmegen
s.verberne@cs.ru.nl

Wessel Kraaij
TNO and Radboud University Nijmegen
kraaijw@acm.org

ABSTRACT

Automatic e-mail categorization with traditional classification methods requires labelling of training data. In a real-life setting, this labelling disturbs the working flow of the user. We argue that it might be helpful to use documents, which are generally well-structured in directories on the file system, as training data for supervised e-mail categorization and thereby reducing the labelling effort required from users. Previous work demonstrated that the characteristics of documents and e-mail messages are too different to use organized documents as training examples for e-mail categorization using traditional supervised classification methods.

In this paper we present a novel network-based algorithm that is capable of taking into account these differences between documents and e-mails. With the network algorithm, it is possible to use documents as training material for e-mail categorization without user intervention. This way, the effort for the users for labeling training examples is reduced, while the organization of their information flow is still improved.

The accuracy of the algorithm on categorizing e-mail messages was evaluated using a set of e-mail correspondence related to the documents. The proposed network method was significantly better than traditional text classification algorithm in this setting.

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

E-mail classification, categorization, transductive transfer learning

IIiX 2014, Regensburg, Germany

1. INTRODUCTION

The life of knowledge workers is changing rapidly. With the arrival of mobile internet, smart phones and the corresponding "any place any time information" it becomes increasingly hard to balance work life and personal life. Additionally, knowledge workers need to be able to handle large amounts of data. Receiving more than 70 new corporate e-mail messages a day is not uncommon [17] so an effective personal information management system is required to be able to organize and re-find these messages. For this purpose 'working in context' is deemed beneficial [8, 24]. Assistance of knowledge workers with 'working in context' is one of the goals of the SWELL project[1] for which this research is executed.

One application area of interest is the e-mail domain. Associating e-mail messages with their contexts has two benefits: 1) it can help knowledge workers find back their messages more easily and 2) reading messages context-wise, for example by project, is more efficient since the number of context switches is minimized. This latter aspect is a suggestion from the 'getting things done' management method [2].

Many e-mail programs have an option to categorize or file messages, which allows for the possibility to associate messages with for example a 'work-project' context. This categorization option however, is often not used optimally, as messages are left to linger in the inbox [25] and many users do not even use category folders at all [12]. Manually categorizing the messages is too big an effort for busy knowledge workers, diminishing the actual benefits of the categorization.

Automated approaches for e-mail message classification are plentiful. The early work in e-mail classification was mostly directed towards detecting spam [18]. This was followed by work towards categorizing e-mails in order to support personal information management [21, 4]. Nowadays, work on classifying e-mails is often directed towards predicting the action required for the message [7, 1, 20]. Only automatic spam classification has become a commodity in email handling. Categorization functionality within e-mail clients often relies on hand-crafted rules.

The downside of the methods based on machine learning is that each of them still requires labeled training data. Although this training dataset only needs to be a limited but representative part of all messages, it still requires effort from the knowledge worker as they would need to label these examples. Especially the persons that receive the most

[1] `www.swell-project.net`

messages, and will most likely benefit the most from a good categorization, will probably not have the time to provide a sufficient amount of labeled examples. Furthermore, knowledge workers are often not consistent in their categorizations [25].

Sappelli et al. [19] tried to reduce the effort required from users by using existing folder structures and the documents in them as training material for supervised algorithms to classify unstructured e-mail data. This is motivated by the idea that especially in a work setting, projects are often organized in project folders. These projects would function as an intuitive context of e-mail messages and therefore would be good categorizations. However, the study demonstrated that the characteristics of documents and e-mail messages are too different to use organized documents as training examples for e-mail categorization using traditional classification methods.

The goal of the research presented in this paper is twofold: (1) we aim to improve upon the work by [19] and (2) we aim to evaluate our new supervised network-based classification method. Since traditional methods such as K-Nearest Neighbours, Naive Bayes and SVM proved unsuccessful when presented with training materials of a different type than the test data, we have developed a method that combines the specific characteristics of documents and e-mail messages and exploits these characteristics to make a more robust classifier.

In Section 2, we describe some supervised, unsupervised and semi-supervised approaches to e-mail categorization. The new network-based classification method that is proposed is described in Section 3. In Section 4 the evaluation of this algorithm is described, followed by a discussion and our conclusions in sections 5 and 6.

2. RELATED WORK

Methods for classification can be divided into supervised approaches, where training examples are provided, and unsupervised approaches, where there is typically no training involved. There are also semi-supervised approaches, where a combination of labeled and unlabeled data is used to reduce the training effort compared to supervised methods. A specific form of semi-supervised learning is transductive transfer learning [3], where knowledge from one domain is transfered to another domain, and where the source domain has an abundance of labeled examples while the target domain has none. This is the approach we take in the presented algorithm in this paper. In this section we describe a few typical e-mail categorization methods in each of the categories.

2.1 Supervised categorization

Although supervised machine learning methods require labelled data, which implicates that they need input from the user, this is the main approach for e-mail classification. Various machine learning algorithms have been proposed. For example, Segal et al. [21] use a classifier in their e-mail organization program MailCat that uses the similarity between a word-frequency vector of a message and TFIDF weighted vectors of categories to determine the correct category. Their algorithm achieves 60-80% accuracy.

Bekkerman et al. [4] evaluate Maximum Entropy (ME), Naive Bayes (NB), Support Vector Machines (SVM) and Winnow on the classification of two e-mail datasets, among which the Enron dataset. Overall, SVM has the highest performance (55-95% dependent on the persons whose messages are categorized).

On the other hand, Chakravarty et al.[5] provide a graph-based approach to email classification which they also evaluate on the Enron dataset. Their performance varies with the number of classes that need to be recognized (60-90% accuracy)

Krzywicki et al. [13] present a method for incremental e-mail categorization. This is based on the idea that the categories in a changing dataset like e-mail change over time. New topics are introduced and older topics can become irrelevant. Their 'clumping' method looks at local coherence in the data. They evaluate their results on the Enron dataset and obtain comparable results as SVM on that dataset (58-95%). Their method however is less complex and therefore has a lower execution time.

Interestingly the variation in classification accuracies presented in existing literature is large. Furthermore, each of these methods requires a large dataset. Usually 70-80% of the data is used as training material. For the 7 largest users in the Enron dataset this corresponds to more than 2000 messages on average that need to be labelled.

2.2 Unsupervised categorization

In unsupervised machine learning methods, usually clustering techniques are used. Xiang [26] presents a non-parametric clustering algorithm using Hubert's γ. They report an accuracy of 70% on average, measured on two personal datasets whereas K-means achieves 47% and Hierarchical Agglomerative Clustering obtains 60% accuracy.

Furthermore, Kulkarni et al.[14] present their system SenseClusters. The authors see an e-mail message itself as a context and seek to group these contexts. Grouping is based on the similarities in content using occurrence and co-occurrence matrices. Labels are given using descriptive and discriminating terms in the clusters. They test their algorithm on the 20-NewsGroups Corpus and report a F-score of 61–83%. The quality of the labels is not evaluated.

These performances are comparable to the supervised setting, and these approaches require no labelling effort of the user. However, there is still an open issue as sometimes the clusters are not labelled or the labels might not be meaningful enough to the user. Additionally, the clustering is based on similarities between messages, and it is by no means certain that these clusters are the clusters the user is looking for.

2.3 Semi-Supervised categorization

There are some approaches that try a combination of supervised and unsupervised learning. Kiritchenko et al.[11] try to reduce the number of required training examples for SVM and Naive Bayes by using co-training. In this technique they separate the features in 2 sets, and train one weak classifier on one set and one on the other. For new examples, each classifier labels the example, and the most confidently predicted positive and negative examples are added to the set of labeled examples. In essence the two classifiers train each other, since when one classifier is confident about a new example, this information can be taught to the other classifier. The results show that this technique can improve SVM classifiers, but also that it has a negative impact when using Naive Bayes classifiers.

Huang and Mitchell [10] propose a mixed-initiative clustering approach for e-mail. The algorithm provides an initial clustering of the messages and the users can iteratively review and edit the clustering in order to constrain a new iteration of automatic clustering. The required effort from the user is halved using this approach, but no interpretative labels are provided for the clusters.

In the e-mail classification method by Park et al.[16] the categories result from clustering, but category labels are obtained from a set of incoming e-mails using either latent semantic analysis or nonnegative matrix factorization. When users are unsatisfied with the category hierarchy derived from the semantic features, they can opt for a dynamic category hierarchy reconstruction which is based on fuzzy relational products. Park et al. did not test their algorithm on an e-mail dataset, but rather on the Reuters document collection. Also, they did not evaluate the quality of the labels that their algorithm provides. It is possible that although their approach is interesting, it might not work as well on e-mail compared to the documents of the Reuters corpus, considering the differences that Sappelli et al.[19] have found. Moreover, it is not certain that the category labels proposed by the algorithm are meaningful to the user.

Two transductive-transfer learning examples come from Koren et al.[12] and Sappelli et al.[19]. Koren et al. [12] propose to use other user's folders to suggest categories, such that users that do not have time to categorize messages themselves can benefit from the categories that others make. Although this would be a solution to reduce the effort for some people, it would require the access to data of other users, which poses serious privacy issues. Also, it would be much harder to use social features such as sender and receiver, since it is unlikely that multiple users have the same social dynamics.

Sappelli et al. [19] compared traditional supervised algorithms such as K-nearest neighbours, SVM and Naive Bayes on the task of e-mail categorization, but provided categorized documents as training data instead of e-mail messages. They tested the algorithms on a personal set of e-mail messages. The authors found that the algorithms were not successful in categorizing e-mail messages when they were trained on related documents. An analysis of models trained on e-mail messages showed that the features required for successfully categorizing messages (such as names and addresses) are too different from the features that are extracted from the categorized documents (content words in general). In fact, the documents do contain the features that are needed for the categorization of e-mail messages (e.g. in the form of author names), but the traditional classification methods are not successful in extracting these features.

3. OUR MODEL FOR E-MAIL CATEGORIZATION

In [19], the authors found that common machine learning algorithms such as Naive Bayes, K-Nearest Neighbours (k-NN) and SVM are not successful in using documents as training examples for classifying emails. The main reason is that for email categorization, contact details, such as the sender or recipient of a message, are the most distinguishing features, while for documents the topic is much more important. In Naive Bayes, k-NN and SVM there was no distinction between the type of features as they were all unigram (bag-of-words) based.

We propose to bridge the domains of emails and documents by introducing contact names as an additional category of features for the joint space of documents and emails. Our model can be viewed as a transfer learning approach where labeled data in the domain of documents is used to learn a classifier for emails, for which no labeled training items are available. In the approach proposed in this paper, the contact-type features in the data play an important role. In e-mail messages, these contact type features are often e-mail addresses, while in documents these features are usually (author) names. To connect the names and contacts, together with other information, we use a network based approach which is based on the interactive activation model by McClelland & Rumelhart [15]. The original interactive activation model is a cognitive model based on theories on neural activity in the brain. The idea is that the model consists of nodes and connections and that activation spreads through the network to activate other nodes. These nodes are comparable to neurons in the brain and the notion of spreading activation is similar to neurons transmitting electrochemical signals to each other. As in the brain, nodes in the model can send inhibitory or excitatory signals. Where this model was originally used to assess validity of cognitive theories, it can also be used as a method for context recognition [22]. In contrast to typical neural networks, there are no hidden layers in the model for context recognition and the connection weights are not learned.

There are two phases in the interactive activation approach. First the network with nodes and connections need to be constructed (Section 3.1). Secondly, to obtain a classification for an input, the activation needs to be spread through the network (Section 3.2).

3.1 Constructing the Network

The network consists of 3 layers as depicted in Figure 1:

- the input layer; the e-mails that need to be categorized
- the context information layer; the various elements that can be extracted from e-mails and documents. These tell us something about the context of the message or document. For the current problem we focus on social context (person names), topics or terms and location information. These information types were chosen because they have a relation to both e-mail messages and documents.
- the output layer; the categories that the user is interested in

Each of these layers consists of nodes. Each node can have one or more weighted connections to other nodes.

First, the contact nodes in the context-information layer are created. We do this by using the knowledge worker's address book on the computer. Names are divided into first names and last names. Only the actual names are kept, words like 'van de' are removed. Each first and last name and each e-mail address receives a node in the context information level. Names and addresses are connected using the address book. Names can be associated with multiple addresses. For example 'John Doe' is divided into 'John' and 'Doe'. The name is associated with 'john.doe@email.com'. Both the node 'John' as well as the node 'Doe' are connected to 'john.doe@email.com'. If there is also a person named

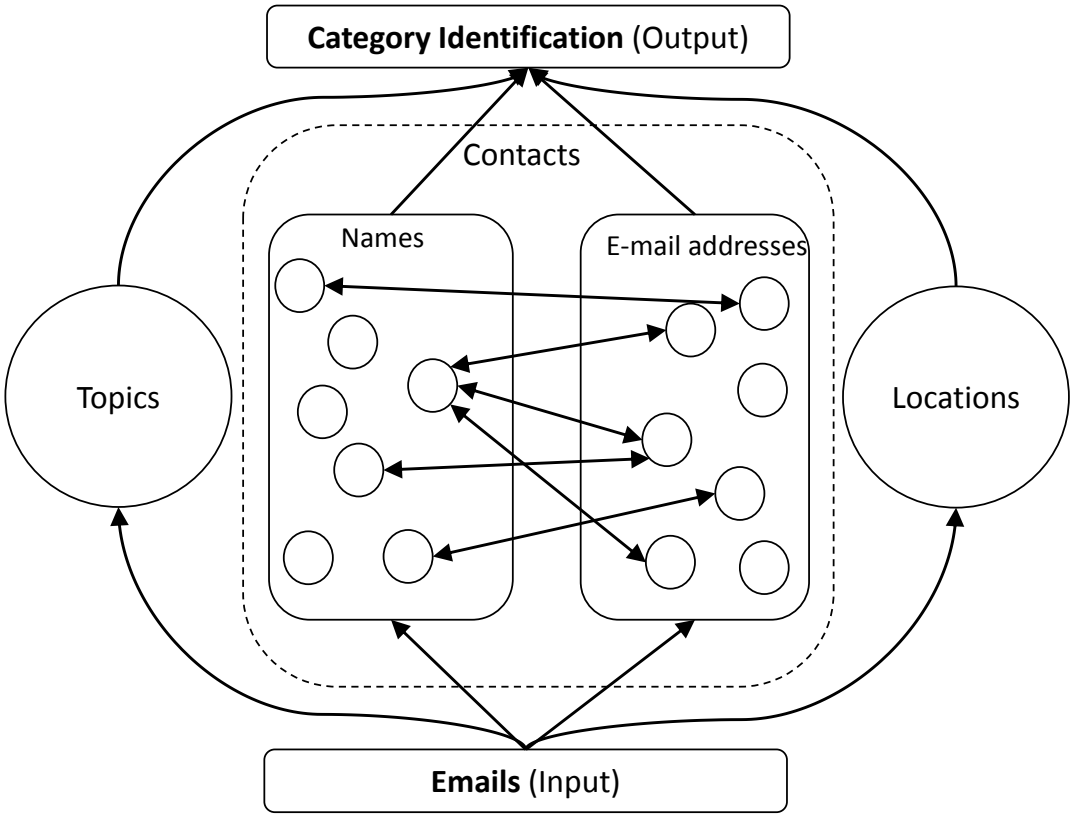

Figure 1: Representation of the network

'John Peters' with the address 'myaddress@email.com' in the addressbook, the node 'john' is connected to the node 'myaddress@email.com' as well. Furthermore, e-mail addresses can be connected to multiple nodes as well, for example when an e-mail address is shared between 2 persons. This idea of connecting nodes is cognitively plausible: When you just read the name 'John' you do not know yet whether this refers to 'John Doe' or 'John Peters' and thus you can think of both e-mail addresses as possible senders.

Additional context information can be added during this phase as well. For constructing the topic nodes, terms or phrases in the documents that are descriptive of the categories are extracted using a weighted combination of three term scoring methods as described in Verberne et al. [23]. The 100 most important terms for a category are extracted, resulting in a set of 19658 important terms and phrases for all categories together when duplicates are removed. These terms and phrases are interesting as they could contain project names, but in any case they represent the content of the documents.

Another type of context information are the location nodes. For each document the filename without the path is extracted and added as a location node. The motivation for these nodes is based on the possibility of attaching files to messages. Since these attachments do not have a path, only the filename itself is informative. If one of these file attachments matches a filename in the location nodes, this is strong evidence for the category of the document from which the location node is created.

The connections from the nodes in the context information layer to the category nodes in the output layer are made based on the analysis of documents on the user's computer. The connections are made by creating a category node in the output layer for each of the project-folders that the user selects. This selection of relevant folders is the only 'supervision' that is required from the users. After the creation of these category nodes, they are connected to the context information layer by analyzing the documents in the corresponding project folder. From each document, names and e-mail addresses that occur in the context information layer are extracted. Each name or e-mail address that occurs is connected to the category name that the document belongs to. The same process is repeated for the terms. If one of the important terms is found in a document, then a connection is made from the term to the category of the document. For the locations, the documents do not need to be analyzed, simply a connection can be made from the location node to the category of the document from which the location node was originally created.

The connections from input layer to context information layer are created during run-time.

Each connection from node n1 to node n2 has an inverse document frequency style connection weight:

$$\frac{1}{\#outputconnections_{n1}} \tag{1}$$

This means that if a node has only a few connections and it is activated it has a high impact, but if the node is connected to many other categories it becomes less important.

For the purpose of the experiments in section 4 the identification layer and the context information layer are fixed at this point; only connections between input and context information layer are added during run-time. However, in a learning setting as presented in section 4.3.2, the network can be adapted further as new connections between context information layer and identification layer can be made given the input message and the corrected or confirmed category. Additionally, it would be expected in a real application that the network is updated regularly, to allow new categories or remove obsolete categories.

3.2 Running the Network

To obtain a category for an input the network needs to be activated. First the e-mail message is added to the input layer as a node. Next, the names, e-mail addresses, topics, file attachment names and other potential sources of information are extracted from the message and connections to the corresponding nodes are created. Then the activation of the input node corresponding to the message is set to 1.0.

First, the weighted excitatory input ex and weighted inhibitory input in are calculated. These are weighted sums of each of the excitatory or inhibitory input connections to a node:

$$ex_j = \sum_{c_{i,j} \in C_{excitatory}} \alpha \cdot a_i \cdot w_{c_{i,j}} \quad (2)$$

where $c_{i,j} \in C_{excitatory}$ is a connection in the set of input connections to node j where $a_i > 0$: the activation of the from-node i is greater than 0. α is the parameter for the strength of excitation and $w_{c_{i,j}}$ is the connection strength between node i and j.

$$in_j = \sum_{c_{i,j} \in C_{inhibitory}} \gamma \cdot a_i \cdot w_{c_{i,j}} \quad (3)$$

where $c_{i,j} \in C_{inhibtory}$ is a connection in the set of input connections to node j where $a_i <= 0$. γ is the parameter for the strength of inhibitions.

Activation of each of the nodes in the network is updated using Grossberg's activation function [9]:

$$\delta a_j = (max - a_j)ex_j - (a_j - min)in_j - decay(a_j - rest) \quad (4)$$

where a is the current activation of a node, ex is the weighted excitatory input of the node (2), in is the weighted inhibitory input (3) and min, max, $rest$ and $decay$ are general parameters in the model (see also Table 1). This function ensures that the activation of a node will go back to the resting level when there is no evidence for that element, and towards 1.0 when there is a lot of evidence for that element.

Normally the network would be run for the number of iterations required to stabilize the activation in the network. However, for pragmatic reasons the network is run for 10 iterations for each input message. This is enough to activate the network properly (i.e. activate all levels) and keeps the running time low. This would be a realistic requirement when the algorithm would be put to use in an actual application. Moreover, more than 10 iterations did not improve accuracy in the experiments described in section 4. This suggests that the network stabilizes quickly.

To obtain the label for the input message, the activation of the category nodes in the output layer can be read. Each node starts with the same resting level, but the variation in number of input connections to a node together with the excitation and inhibition parameters can alter this resting level slightly. Therefore, the increase in activity of a node is compared to its individual start level and the node with the highest increase in activation will be selected as label for the input message.

4. EXPERIMENTS

In a series of experiments we compare our method for e-mail categorization using documents as training data to the previous approach by Sappelli et al. [19]. In the first experiment we look at a network with only contact nodes. In a second and third experiment we add topic nodes and location nodes respectively to see whether this enhances the network.

4.1 Data

We obtained the personal email and document dataset from Sappelli et al. [19]. This dataset consists of 354 documents and 874 e-mails.The documents as well as the emails were provided in raw text form. This data had been manually categorized into 43 categories corresponding to 43 different courses followed by the single student who provided the dataset. These courses were followed in 4 years time and are part of 2 different curricula; Linguistics and Artificial Intelligence (AI). A third curriculum-type category was the Thesis category, as this was a combination of both the Linguistics and AI curriculums. The data is hierarchically ordered based on curriculum and year (See Figure 2).

Our aim is to support knowledge workers in their working life by categorizing messages to projects. Although at first sight a dataset of a student's course related documents and e-mail messages might not seem relevant for the knowledge worker's life, there is a clear link. Both courses and projects have contextual elements. They both have topics, they both have documents related to the topics, and in both projects part of the work or all the work can be executed in collaboration. Thus, both courses and knowledge worker projects have a social and topical context. In fact, a course can be seen as a project that the student is working on.

There are three relations between the documents and e-mails in the dataset. First, the documents and the messages are about the same courses, so there is a topical relation between them. Furthermore, the documents and messages share a time relation as the messages are sent and received during the course period. Some of the documents have been created by the student in that time period as well, while some documents, such as course materials written by the teacher, have already been obtained at the start of the course. This means that the training documents do not necessarily preceed all e-mail messages in their creation dates. Finally, documents can be send via e-mail messages, creating an attachment relation between a message and a document. However this last relation was not very common as only 5.8% of the e-mail messages contained a training document as file attachment.

4.2 Parameter Optimization

A small subset of the data (7 categories, 122 documents, 53 e-mail messages) was used to optimize the parameters

Table 1: Parameters

Parameter	Definition	Min	Max	Stepsize	Default	Optimal
α	Strength of excitation	0.0	1.0	0.1	0.1	0.2
γ	Strength of inhibition	0.0	1.0	0.1	0.1	0.0
Min	Minimal value of activation	−1.0	0.5	0.1	−0.2	−0.2
Max	Maximal value of activation	0.0	1.0	0.1	1.0	1.0
$Rest$	Resting-level of activation	−1.0	0.5	0.1	−0.1	−0.1
$Decay$	Strength of decay	0.0	1.0	0.1	0.1	0.3

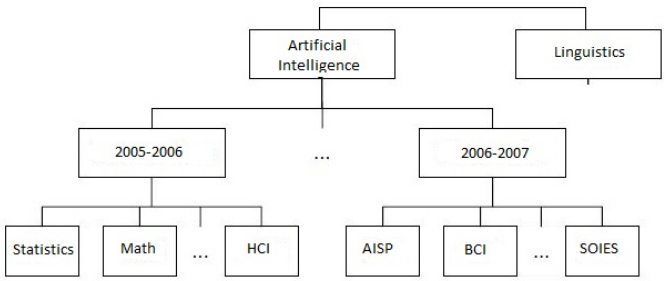

Figure 2: Example of document category structure (Adopted from [19])

in our network. A grid search optimization was executed with the minimum, maximum and step-sizes as mentioned in Table 1.

Table 1 also shows the default parameters of the original IA model, and the parameters that we used in the experiment. There were multiple sets of parameters that proved to be optimal. We have chosen to use a set that seemed logical. The strength of excitation in the network was increased while the strength of inhibition was decreased compared to the default. This boosted the impact of observed nodes, while reducing the effect of unobserved nodes. The decay parameter was also increased. The decay parameter pushes the activation back to the resting value, which happens faster with higher decay values. This was as expected for categorizing e-mail messages since there does not need to be a relation between one message and the next.

4.3 Results

In Table 2 we present the accuracy of the presented network method in various forms. We compare the accuracy of the algorithm presented in this paper to traditional algorithms such as Naive Bayes, K-nearest neighbours (k-NN) and Linear SVM. These are the baseline runs and are in the top part of table 2. These traditional algorithms are trained on a bag of word unigram model with TF-IDF weighting, where k-NN and SVM are pruned: words that occurred in less than 3% or more than 30% of the documents were excluded from the feature vectors. The accuracy that can be obtained when the most frequent class is always selected is also presented (ZeroR). The reported accuracies are adopted from Sappelli et al. [19].

In addition we improved the term selection for Naive Bayes, k-NN and linear SVM by using the same term extraction method as in the network approach, which is described in Section 3.1.

The significance of the difference between the algorithms and the best baseline run (k-NN) is measured using McNemar's test [6]. This statistical test measures whether the marginal frequencies are equal and can be used on paired nominal data. The null hypothesis is that two classifiers (C1 and C2) are equally accurate. The McNemar test statistic is:

$$\chi^2 = \frac{(b-c)^2}{b+c} \tag{5}$$

where b is the number of times C1 is correct and C2 is wrong, and c is the number of times C2 is correct and C1 is wrong. The χ^2-value is compared to the chi-squared distribution to determine the p-value.

When we look at the result of a network with only topic nodes, it already has a significantly higher performance than Naive Bayes ($p < 0.001$), while the result is comparable to SVM ($p = 0.39$). The network approach uses less features than Naive Bayes.

Table 2: Classification Accuracy. The top block shows the baseline accuracies adopted from Sappelli et al.[19]. The middle block shows the accuracies obtained with the network method. The bottom block shows improved accuracies for the supervised methods using either the same term selection as in the network (descriptive term extraction) or contact names

Classifier	Accuracy
ZeroR (majority baseline)	15.3%
Naive Bayes (no pruning)	1.7%
Linear SVM	7.8%
k-NN, K=5	20.9%
Network – Contacts only	53.7%
Network – Topics only	7.1%
Network – Location only	14.0%
Network – Contacts & Location	56.7%
Network – Contacts & Topics	57.7%
Network – Contacts, Location & Topics	**58.3%**
Naive Bayes – Descriptive term extraction	3.2%
Linear SVM – Descriptive term extraction	15.9%
k-NN, K=5 – Descriptive term extraction	32.8%
k-NN, K=5 – Contacts only	54.6%

When looking at location nodes (i.e. filenames matching file attachments), the accuracy of the network goes towards the majority baseline and is a significant improvement over Naive Bayes ($p < 0.001$) as well as SVMs ($p < 0.001$).

Using only contact nodes in the network gives a significantly better performance ($p < 0.001$) than the traditional classification methods that were explored so far, even the ones that were improved by using the descriptive term extraction method as described in Section 3.1. However, if we train a k-NN algorithm with $k = 5$ on the documents where we first extract all contact features, we can obtain a similar accuracy of 54.6%. There is no significant difference

between k-NN with only contacts and the network with only contacts ($p = 0.54$), but the network has the advantage that additional information types can be added. Adding location nodes to the network with contact nodes boosts the performance a little bit further to 56.7%. A network with contact nodes, location nodes as well as topic nodes gives the highest performance of 58.3%. Although this is a nice improvement, the difference between the complete network and k-NN with only contact nodes is not significant ($p > 0.06$).

4.3.1 Influence of number of classes

As discussed before, we aim to support knowledge workers by categorizing messages to the projects that the knowledge worker is working on. It is not realistic that a knowledge worker would work on 43 projects at the same time. Therefore we have looked at the influence of the number of categories.

We first try to make the task easier by categorizing the messages to curriculum rather than to course level. This is essentially a categorization higher up in the hierarchy. The curriculum level has 3 categories; *Artificial Intelligence (AI)*, *Linguistics* and *Thesis*. The *Thesis* category is not actually a curriculum, but is rather a combination of both *AI* and *Linguistics* and is therefore placed on the curriculum level. The full network (contacts, topics & locations) correctly classifies 73.5% of the messages. When we look at the confusion matrix in table 3 it becomes apparent that most mistakes are made between the *AI* messages and *Thesis*-messages (11% error). This is not strange, as the thesis was a continuation of a course in the *AI* curriculum, so there is a large overlap between contacts.

More interesting is the number of *AI* messages that are mistakenly classified as *Linguistics*, while there are no *Linguistics* messages mistakenly classified as *AI*. It seems that these errors are related to ambiguity in the names. Both categories have connections to a couple of first-names that occur often even though the actual persons to which they refer may be different. The messages that are wrongly classified, typically have these common names associated with them. Apparently the *Linguistics* category is favored in the case of these ambiguous situations.

Table 3: Confusion Matrix Network - Curriculum

	AI predicted	Thesis predicted	Linguistics predicted
AI	421	90	82
Thesis	5	104	26
Linguistics	0	24	104

A k-NN algorithm filtered on contacts actually achieves a significantly higher accuracy of 79.3% ($p < 0.001$) compared to the network algorithm when tested on curriculum categories. The confusion matrix in Table 4 shows that for k-NN there is much less confusion between *AI* and *Linguistics*. There are two possible explanations: 1) the impact of ambiguity in names is smaller because the weighting in k-NN is different or 2) the network algorithm is harmed by the influence of previous messages.

In a second attempt to make the task more realistic, we categorize messages to courses again, but the network is built year-wise, such that there are only 5-14 categories at a time. The training data then consists of only the documents corresponding to the courses that were taken in a specific year.

Table 4: Confusion Matrix k-NN - Curriculum

	AI predicted	Thesis predicted	Linguistics predicted
AI	485	61	47
Thesis	27	100	8
Linguistics	24	10	94

Table 5: Classification Accuracy: year-wise training

Year	#categories	accuracy Network	accuracy k-NN
2005-2006	5	20.5%	73.1%
2006-2007	12	20.0%	22.5%
2007-2008	14	40.8%	31.6%
2008-2009	11	73.6%	62.9%

For the k-NN algorithm trained on contacts only there is a clear advantage of a reduced number of categories (See Table 5). Interestingly this does not seem to be the case for the network algorithm. The total accuracy that could be obtained is for both algorithms lower than when all categories are classified at the same time.

When the accuracies of the network algorithm are analyzed, the accuracy for the courses in the year 2008-2009 is the highest. Most likely this is because, 2008-2009 is a particularly easy year as 84.9% of the messages can be classified in 3 out of 11 categories.

Similarly, 2005-2006 seems to be a very difficult classification. In 2005-2006 there are only 5 courses, and in 3 of them there was a collaboration with the same persons. Moreover all 3 courses were quite similar in topic. The most confusion in this year existed between the courses *Datastructures* and *HCI*: 57.1% of *Datastructures* messages were misclassified as *HCI*. There were actually 43 documents for *Datastructures* while there were only 4 for *HCI*. After inspection of these training documents, it is clear that the *HCI* documents contain more social references related to the course, making them more suited as training material for e-mail categorization.

However, this problem would be the same for the k-NN algorithm, but k-NN seems less influenced by the overlap in contacts and the lack of references in some of the training documents. Moreover, k-NN seems better at selecting the larger categories when the input is ambiguous, which improves the accuracy greatly. Nevertheless it has a class recall of 0% for 3 out of 5, which is not satisfactory. We expect that k-NN has a higher accuracy for the year 2005-2006 because the larger categories also had many more documents. This means that there are more documents that can be close to a message and a higher chance that k-NN will select that category as the class. The network is not influenced by the number of documents as there only needs to be one example to make a connection. In this particular year, this has a large influence since the network prefers a smaller class *HCI* over the larger class *Datastructures*, whereas k-NN always prefers *Datastructures*.

4.3.2 Learning Curve

The model can be improved when the content of the e-mail messages is used to make additional connections in the network. Initially there will be no labeled messages available,

but as the user uses the system it can correct or confirm categorizations. From these corrections and confirmations the network can learn, because for these messages it is absolutely certain what the label should be. New connections between context information nodes and category labels can be made. In particular, direct connections between e-mail addresses and categories can be created. In the learning curve experiment we look at a model with only contact nodes to see what we can achieve with the least complex network possible.

Figure 3 shows the learning curve of the network. Increasingly more labeled e-mail examples, are presented to the network, improving the classification accuracy. For this experiment we chose to randomly select the e-mail examples, as this would be a realistic setting when users confirm or correct labels. The figure shows the learning curve for the situation where an initial model based on documents is improved, as well as a traditional supervised situation where there is no initial model.

From the figure it is clear that the learning curves are steep. The network learns quickly when it is presented with e-mail messages, and stable 80% accuracy is obtained by the network with the initial model at around 20% training examples (about 170 messages). For the model without the initial training on documents this is obtained at approximately 35% training examples (about 300 messages). Overall, the model without the initial training obtains an accuracy of 85%, which is not significant compared to the 83% accuracy for the model with initial training on documents. The actual selection of training examples has a large impact on the performance. If all training examples that are presented come from the same category, the impact is much smaller, since less new information is introduced. Optimally, at least one example per category should be selected.

The maximum accuracy of the network model is not as high as a supervised Naive Bayes model trained only on emails (89.9%)[19]. With a Naive Bayes supervised algorithm 20% of training examples are required to achieve 80% accuracy, just like the situation with the initial network model. However, with Naive Bayes, there is actually a decreased performance when a model initially trained on documents is used (30% training examples required and a maximum accuracy around 85%). Moreover with a Naive Bayes network only trained on messages, the accuracy starts at 0%, while the user with the network method, initialized with documents, already receives a correct classification in 58% of the cases. Thus, out of the 170 messages, only 72 need to be corrected in the network method, whereas with Naive Bayes all 170 messages need to be labeled.

5. DISCUSSION

We have presented a network-based algorithm that has several advantages. First and foremost the presented algorithm is suited for transduction transfer learning since it is capable of using items as training examples that are only partially related to the examples that need to be categorized. The benefit is that using this method already existing categorizations such as folder structures on the file system can be re-used.

Traditional classification methods like K-nearest neighbour, SVM and Naive Bayes have difficulties in extracting the features from documents that are needed for e-mail categorization[19]. They can be helped by pre-filtering the documents on a specific type of features, such as contact details. However, these methods can not combine multiple feature types easily. The network-based method presented in this paper is more flexible. The method is capable of focusing on specific feature types, without the need to filter out other feature types. More importantly, the network approach is not harmed by feature types that are not very effective by themselves, such as the terms, but it can still use them for small improvements.

The classification accuracy reached with the proposed network is much higher than the accuracy of traditional supervised methods on the task of classifying e-mail messages with documents as training data. Using the proposed method, an application can provide reasonable category suggestions at first, and as the user corrects or confirms the system the accuracy can improve to levels almost as good as state of the art for supervised methods that do use labelled e-mail data. A reasonable initial classification substantially lowers the bar to use the system. Additionally it decreases the total amount of effort required from the user, as less training examples need to be labeled. With a reasonable initial classification, many items only need to be confirmed, rather than corrected, requiring less effort as well.

The presented system would be especially meaningful for situations where the effort it costs to label a message outweighs the benefits that the labeled messages has. An example stems from a person that does categorize messages for external projects, but finds it too much effort to label internal projects. The reason for this comes from the nature of the projects; external projects typically have clear bounderies, and little overlap in contacts. Also new people join as the project progresses. Thus, they are easy to label, and labeling them has advantages as it is important to keep track of all the agreements that have been made. Internal projects on the other hand have much higher overlap and are therefore more difficult to label by the user. The presented system would help the user label the external projects by reducing the number of mails that actually need to be labelled, and provides additional benefits to the internal projects that would otherwise not have been labeled. This example also demonstrates that it is not always easy for the user to label data and that help provided by suggested labels by a system could prove beneficial.

A disadvantage of the network is that it cannot discover new categories. However, the network structure allows for the possibility to use graph clustering methods to find clusters of information. This could potentially be used as a method to identify new categories, which we will look into in future work.

Other things that we will be investigating further are the flexibility of usage of the algorithm. First of all, many sources of information such as social information, location information and topic information can be combined in a simple manner. Hierarchically organized information could for example be represented naturally using additional context information levels, where each level in the network corresponds to a level in the hierarchy. Additionally, text features do not have the bulk-advantage as they have in some algorithms, making it easier to combine them with different feature types. This allows the model to be used in several research domains. Currently we are investigating the benefits and issues when using a more elaborate version of the method for context recognition, where the input data is a stream of sparse events.

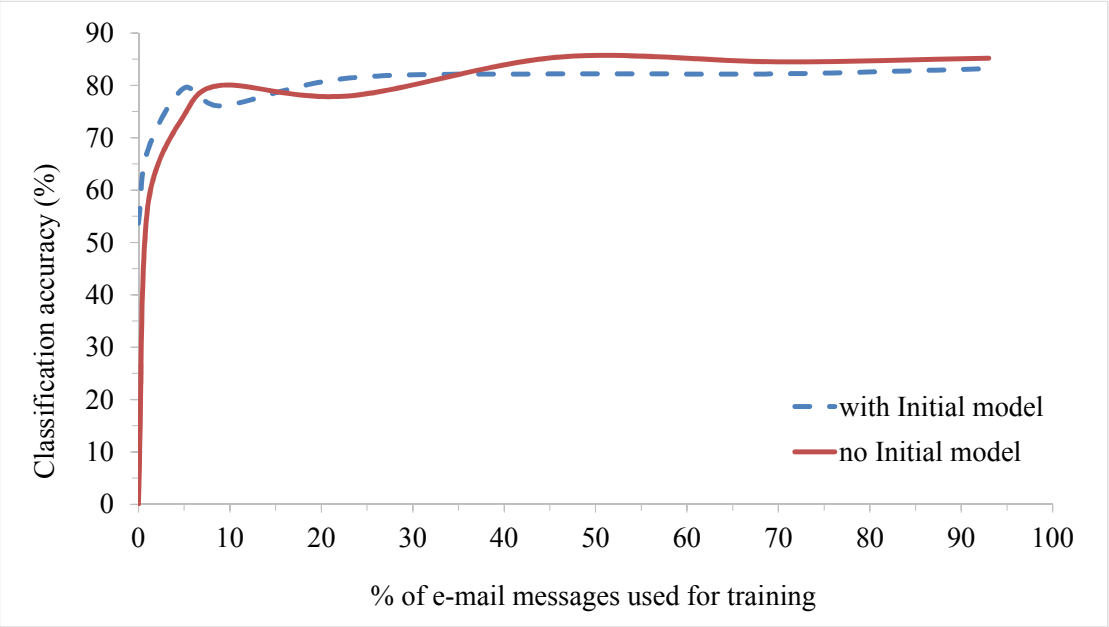

Figure 3: Learning curves of the algorithm with initial training on documents vs. without initial training.

Furthermore, the network is very insightful. It is easy to discover relationships between contacts and categories. The method is not limited to categorizations, but can also be used to give insight into a person's data, which is one of the goals in the SWELL project.

The network could be used in a context-aware notification filtering setting as well. This would be a combination of using the model for context recognition and for e-mail categorization. When a new incoming e-mail message fits the current context, a notification can be shown, while if it does not fit, it can be suppressed. This could also be combined with information on the importance of a message [20].

6. CONCLUSION

We presented a novel method that is capable of exploiting data from documents as training material for classification algorithms that categorize e-mail messages. The advantage of the proposed method is that the training materials need to be only partially representative of the data that needs to be categorized. This means that more sources of information can be used. Also, categorizations that have already been made by a user in a different, but related domain (such as file organization), can be re-used. In the end this reduces the effort of the user that is typically required when dealing with supervised machine learning systems. We could reduce the number of messages that needed to be labelled or corrected by the user from 170 messages for Naive Bayes to 72 for the network method, in order to obtain a classifier with 80% accuracy in the experiment presented in this paper . This is a reduction of almost 60%.

7. ACKNOWLEDGEMENTS

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

8. 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] D. Allen. *Getting Things Done. The Art of Stress-Free Productivity*. Penguin, January 2003. ISBN-10: 0142000280 ISBN-13: 978-0142000281.

[3] M. T. Bahadori, Y. Liu, and D. Zhang. Learning with minimum supervision: A general framework for transductive transfer learning. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 61–70. IEEE, 2011.

[4] R. Bekkerman. Automatic categorization of email into folders: Benchmark experiments on Enron and SRI corpora. *Computer Science Department Faculty Publication Series*, page 218, 2004.

[5] S. Chakravarthy, A. Venkatachalam, and A. Telang. A graph-based approach for multi-folder email classification. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 78–87. IEEE, 2010.

[6] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923, 1998.

[7] M. Dredze, T. Brooks, J. Carroll, J. Magarick, J. Blitzer, and F. Pereira. Intelligent email: Reply and attachment prediction. In *Proceedings of the 13th*

international conference on Intelligent user interfaces, pages 321–324. ACM, 2008.

[8] 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.

[9] S. Grossberg. Adaptive pattern classification and universal recoding: I. parallel development and coding of neural feature detectors. *Biological cybernetics*, 23(3):121–134, 1976.

[10] Y. Huang and T. M. Mitchell. Exploring hierarchical user feedback in email clustering. In *Email'08: Proceedings of the Workshop on Enhanced Messaging-AAAI*, pages 36–41, 2008.

[11] S. Kiritchenko and S. Matwin. Email classification with co-training. In *Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research*, page 8. Citeseer, 2001.

[12] Y. Koren, E. Liberty, Y. Maarek, and R. Sandler. Automatically tagging email by leveraging other users' folders. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 913–921. ACM, 2011.

[13] A. Krzywicki and W. Wobcke. Exploiting concept clumping for efficient incremental e-mail categorization. In *Advanced Data Mining and Applications*, pages 244–258. Springer, 2010.

[14] A. Kulkarni and T. Pedersen. Senseclusters: Unsupervised clustering and labeling of similar contexts. In *Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions*, ACLdemo '05, pages 105–108, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[15] J. L. McClelland and D. E. Rumelhart. An interactive activation model of context effects in letter perception: I. an account of basic findings. *Psychological review*, 88(5):375, 1981.

[16] S. Park and D. U. An. Automatic e-mail classification using dynamic category hierarchy and semantic features. *IETE Technical Review*, 27(6), 2010.

[17] S. Radicati. Email statistics report, 2010.

[18] 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.

[19] M. Sappelli, S. Verberne, and W. Kraaij. Using file system content to organize e-mail. In *Proceedings of the fourth symposium on Information interaction in context*, 2012.

[20] M. Sappelli, S. Verberne, and W. Kraaij. Combining textual and non-textual features for e-mail importance estimation. In *Proceedings of the 25th Benelux Conference on Artificial Intelligence*, 2013.

[21] R. B. Segal and J. O. Kephart. Mailcat: An intelligent assistant for organizing e-mail. In *Proceedings of the Third Annual Conference on Autonomous Agents*, AGENTS '99, pages 276–282, New York, NY, USA, 1999. ACM.

[22] S. Verberne and M. Sappelli. D3.2 activity classification. Technical report, COMMIT P7 SWELL, 2013. Available on `http://www.swell-project.net/results/deliverables`.

[23] S. Verberne, M. Sappelli, and W. Kraaij. Term extraction for user profiling: evaluation by the user. In *Proceedings of the 21th International Conference on User Modeling, Adaptation and Personalization*, 2013.

[24] P. Warren. Personal information management: The case for an evolutionary approach. *Interacting with Computers*, 2013.

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

[26] Y. Xiang. Managing email overload with an automatic nonparametric clustering system. *The Journal of Supercomputing*, 48(3):227–242, 2009.