Adding Semantics to Email Clustering

Hua Li[1], Dou Shen[2], Benyu Zhang[1], Zheng Chen[1], Qiang Yang[2]

[1]*Microsoft Research Asia, Beijing, P.R.China*
{huli, byzhang, zhengc}@microsoft.com

[2]*Department of Computer Science and Engineering,*
Hong Kong University of Science and Technology
{dshen, qyang}@cse.ust.hk

Abstract

This paper presents a novel algorithm to cluster emails according to their contents and the sentence styles of their subject lines. In our algorithm, natural language processing techniques and frequent itemset mining techniques are utilized to automatically generate meaningful generalized sentence patterns (GSPs) from subjects of emails. Then we put forward a novel unsupervised approach which treats GSPs as pseudo class labels and conduct email clustering in a supervised manner, although no human labeling is involved. Our proposed algorithm is not only expected to improve the clustering performance, it can also provide meaningful descriptions of the resulted clusters by the GSPs. Experimental results on open dataset (Enron email dataset) and a personal email dataset collected by ourselves demonstrate that the proposed algorithm outperforms the K-means algorithm in terms of the popular measurement F1. Furthermore, the cluster naming readability is improved by 68.5% on the personal email dataset.

1. Introduction

Emails become an important medium of communication. A user may receive tens or even hundreds of emails every day. Handling these emails takes much time. Therefore, it is necessary to provide some automatic approaches to relieve the burden of processing the emails.

A straightforward method is to group the similar emails by supervised classification. Supervised methods need a predefined taxonomy. The taxonomy may evolve over time with the change of the users' work, which requires the users to update the taxonomy manually. What is more, whenever the taxonomy is changed, a considerable amount of training data is indispensable for building an effective classifier. However, the preparation of training data is time-consuming and expensive. Thus, an unsupervised technique such as clustering is an attractive alternative.

Conventionally, email clustering is based on the representation of bag-of-words. This simplistic approach cannot take full advantage of valuable linguistic features inherent in the semi-structured emails, which may result in unsatisfactory performance. In this paper, we present a novel technique to cluster emails according to the sentence patterns discovered from the subject lines of the emails. In this method, each subject line is treated as a sentence and parsed through natural language processing techniques. After that, the terms in the subject lines are converted to generalized terms such as "person", where "person" can be instantiated as different people names in different emails. Based on the generalized terms, we mine some patterns called generalized sentence patterns (GSP) to indicate the overall meaning of the subject lines. An example of the GSPs is {"person", "seminar", "date"} which means that someone ("person") gives a "seminar" on someday ("date"). It is clearly that the GSPs can help summarize the subjects of a large number of similar emails which results in a semantic representation of the subject lines. To mine GSPs, we utilize the existing frequent closed itemset mining techniques. However, some redundancy still exists in the set of closed GSPs. Grouping similar GSPs is a simple way to tackle this problem. GSPs in the same group will represent the same cluster. The similarity between two GSPs is defined based on their subset-superset relationship and their supports. Meanwhile, the number of GSP groups can be several times larger than the actual number of desired clusters. A heuristic rule based on the length and the support of the GSPs is applied to select GSP groups.

Once the GSPs are discovered, we can leverage them for improving the clustering performance. We consider a novel unsupervised approach which treats GSPs as pseudo class labels and clusters the emails through a supervised learning algorithm (although no human labeling is involved). Our experimental results show that the proposed algorithm substantially improves the clustering performance in terms of some popular measurements as compared to other clustering approaches that do not consider GSPs. Besides that, our proposed method can also provide meaningful and

Proceedings of the Sixth International Conference on Data Mining (ICDM'06)
0-7695-2701-9/06 $20.00 © 2006 IEEE

precise descriptions of the resulted clusters. This is an important side-product since it is often considered as a challenging task to generate descriptions for document clustering, while a precise description of a cluster can help users understand a large collection of documents easily.

The main contributions of this paper can be summarized as follows:

(1) A novel algorithm is proposed to automatically mine the semantic knowledge from subject lines of emails in terms of generalized sentence patterns (GSP);

(2) A novel clustering algorithm is proposed to leverage the discovered GSPs. Experiments on both an open email dataset and a dataset collected by ourselves show that our method achieves significant improvement as compared to the K-means clustering algorithm without using the GSPs.

The rest of the paper is organized as follows. Section 2 describes the generation of GSPs and the proposed email clustering algorithms based on GSPs; Section 3 presents the experimental results and analysis; Section 4 concludes the paper.

2. GSP based Email clustering

In this section, we describe the proposed GSP based email clustering algorithm and the steps to generate generalized sentence patterns (GSP) and GSP groups.

2.1. Generalization of Terms in email subjects

To mine GSPs from email subjects, the first step to generalize the terms in the subjects. In this paper, a natural language parser, Microsoft's NLPWin [4], is employed for this purpose. The NLPWin tool takes a sentence as input and builds a syntactic tree for the sentence. Figure 1 is an example syntactic tree generated by NLPWin tool for the sentence (email subject) "Welcome Bob Brill."

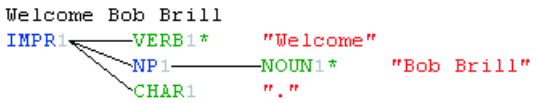

Figure 1. An example of syntactic tree generated by NLPWin.

NLPWin tool can generate factoids for noun phrases, e.g. person names, date and places. Part of the nodes in the syntactic tree has an attribute "FactPred" to specify their factoids predicted by the statistical language model of NLPWin. The factoid of a node is essentially a generalization of the word/phrase represented by the node, which captures its semantic meaning. The original words in the email subjects are replaced as the factoids to help mine sentence patterns as described in next subsection. For example, in the above example syntactic tree, the predicted factoids of the node "NOUN1" is "person" as shown in Figure 2.

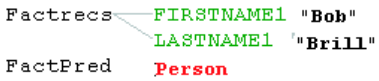

Figure 2. Factoids Predicted for the node "NOUN1".

2.2. Mine generalized sentence pattern

For each email subject, after stop words are removed, NLPWin is used to generate its syntactic tree, and the factoids of the nodes are added into the email subjects. The resultant email subjects are called as generalized sentences. For example, from the syntactic tree shown in Figure 1, {welcome, Bob, Brill, person} is the generalized sentence. The subsets of a generalized sentence are also called generalized sentences. Given two generalized sentence s_1 and s_2, if s_1 is a subset of s_2, then we say that s_2 contains s_1, which is similar to that of frequent itemset. The formal definition of generalized sentence pattern is as follows:

Definition: Generalized Sentence Pattern (GSP). Given a set of generalized sentences $S = \{s_1, s_2, ..., s_n\}$ and a generalized sentence p, the support of p in S is defined as $\sup(p) = |\{s \mid s \in S \cap p \subseteq s\}|$. Given a minimum support *min_sup*, if $\sup(p) \geq \min\_\sup$, then p is called as a generalized sentence pattern, or GSP for short.

For example, {*seminar*, *person*} and {*interview*, *date*} are two GSPs discovered in the experiments, where *person* and *date* are factoids generated by the NLPWin tool. Frequent itemset mining techniques can be directly applied on the generalized sentences to mine GSPs.

To reduce the number of generated GSPs, only closed generalized sentence patterns are to be mined in our experiments. The definition of closed GSP is similar to that of frequent closed itemsets [6][7]. In the rest of the paper, whenever we mention GSPs, we refer to closed GSPs. All existing frequent closed itemsets mining algorithms can be naturally applied on the generalized sentences to mine closed GSPs.

2.3. GSPs grouping and selection

Although mining closed GSPs can reduce the number of generated GSPs substantially, some redundancy still exists in the set of closed GSPs.

Grouping similar GSPs together is a simple way to tackle the above problem. GSPs in the same group will represent the same cluster. The similarity between two GSPs p and q is defined based on their subset-superset relationship and their supports. A single-link clustering algorithm is applied to group similar GSPs together. Each group is called as a GSP group. The similarity function is defined as follows:

$$sim(p,q) = \begin{cases} 1, & p \subset q \cap \dfrac{\sup(q)}{\sup(p)} \geq \min\_conf \\ 0, & otherwise \end{cases} \quad (1)$$

in which, *min_conf* is a parameter to control when to put two GSPs into the same group. Intuitively, a high *min_conf* may fail to group similar GSPs together; while a low *min_conf* threshold will group more GSPs together but it may introduce some noises. Experimental results showed that *min_conf* value between 0.5 and 0.8 were safe for all the tested data sets, and grouping similar GSPs together improves the performance substantially.

The number of GSP groups can be several times larger than the actual number of clusters. A heuristic rule based on the length and the support of the GSPs is applied to select GSP groups. First, we sort the GSP groups in descending order of length. Second, we sort them in descending order of support. Finally, we select the first *sp_num* GSP groups for clustering. The length of a GSP group is defined as the maximal length of the GSPs in that group and the support of a GSP group is defined as the maximal support of the GSPs in that group. A parameter *sp_num* is used to control how many GSP groups are selected for clustering. The rationale behind this simple heuristic rule is that a longer GSP is more confident than a shorter one in deciding the membership of the emails.

2.4. GSP-PCL: GSP as pseudo class label

Based on GSPs, we proposed a novel clustering algorithm to form a pseudo class for the emails matching the same GSP group, and then use a discriminative variant of Classification Expectation Maximization algorithm (CEM) [2] [5] to get the final clusters. When CEM is applied to document clustering, the high-dimension usually causes the inaccurate model estimation and degrades the efficiency. Here linear SVM is used as the underlying classifier.

As shown in the following pseudo code of the GSP-PCL algorithm, only the emails not matching any GSP group are classified. The algorithm would stop when it converges or the predefined iteration limit is reached. A threshold is defined to control whether an email is put into a pseudo class. Only when the maximal posterior probability of an email is greater than the given threshold, the email will be put into the class with the maximal posterior probability, otherwise the email will be put into a special class D_{other}.

Algorithm: GSP-PCL

GSP-PCL (k, GSP groups $G_1, G_2, \ldots G_{sp\_num}$, email set D)

1. Construct *sp_num* pseudo classes using GSP groups, $D_i^0 = \{d \mid d \in D \text{ and } d \text{ match } G_i\}$, $i = 1, 2, \ldots, sp\_num$;
2. $D' = D - \bigcup_{i=1}^{sp\_num} D_i^0$;
3. Iterative until converge. For the j-th iteration, j>0:
   a) Train a SVM classifier based on D_i^{j-1}, i=1,…, *sp_num*;
   b) For each email $d \in D'$, classify d into class D_i^{j-1} if $P(D_i^{j-1}|d)$ is the maximal posterior probability and $P(D_i^{j-1}|d) \geq min\_class\_prob$;
4. $D_{other} = D - \bigcup_{i=1}^{sp\_num} D_i^j$;
5. Use basic K-means to partition D_{other} into (*k-sp_num*) clusters.

The proposed GSP-PCL algorithm uses GSP groups to construct initial pseudo classes. SVM classifier is fed by the classification output of the previous iteration. The *sp_num* parameter in GSP-PCL should be no greater than the desired number of clusters k.

3. EXPERIMENTS

To demonstrate the effectiveness of our proposed GSP-PCL clustering algorithm, we conduct several experiments on two Email datasets: the open dataset Enron email dataset and a private email dataset collected by ourselves. On the private data, we conduct a case study on clustering naming.

3.1. Email datasets

In this section, we describe the open Enron email dataset and the private email dataset used in our experiments.

3.1.1. Enron email dataset. The Enron email dataset [8] is the archive email from many of the senior management of Enron Corporation, and is now the public record. The dataset is provide by SRI after major clean-up and removed of attachments.

The Enron email dataset used here is a subset of the original Enron email dataset, which is generated by Bekkerman, McCallum, and G. Huang [1].

Table 1. Statistics on private email dataset.

User	#Folder	#Message	#Smallest	#Largest
User1	25	1178	7	260
User2	15	719	9	176
User3	11	1500	8	522
User4	5	476	33	144

3.1.2. Private email dataset. Four volunteers (named as user1, user2, user3 and user4 for privacy in table 1) in our organization provide us with their personal email for experiments. Each of them has manually organized his/her emails into self-defined folders before this research work began. The similar preprocessing as Enron email dataset is conducted on this private email dataset to clean the data.

3.2. Evaluation Criteria

The clustering performance of the proposed GSP-PCL clustering algorithm is evaluated against the manually generated class labels based on external criteria [3].

Let $C = \{C_1 \ldots C_m\}$ be a set of clusters generated by a clustering algorithm on a data set X, and $B = \{B_1 \ldots B_n\}$ be a set of predefined classes on X.

Table 2. Relation between two objects

		C	
		Same	Different
B	Same	SS	DS
	Different	SD	DD

Each pair of two objects (x_i, x_j) from the data set X belongs to one of the four possible cases as shown in Table 2. After computing the four values in Table 2, precision and recall and F1-Measure are calculated as following:

$$P = \frac{|SS|}{|SS + SD|}, R = \frac{|SS|}{|SS + DS|}, F1 = \frac{2PR}{P + R} \qquad (2)$$

3.3. Clustering Performance Study

We compared the proposed GSP-PCL algorithm with the basic K-means algorithm to show its effectiveness. The basic K-means algorithm randomly selects *k* emails as the initial cluster centers. To alleviate the effectiveness of random initialization, we ran K-means clustering for 10 times and report the average performance in the following experiments. Meanwhile, to study the effectiveness of the pseudo class label generated by GSPs, we use the generated GSPs to initialize the K-means clusters. We call the resultant algorithm as GSP-means, which is another baseline algorithm.

3.3.1. Experimental results on Enron email dataset. In the experiments conducted on Enron email dataset, the minimum support threshold (*min_sup*) is set as 4 to generate the GSPs and the minimum length of GSPs is restricted to 2. The cluster number *k* on each user's email data is set as the folder number of each user.

Experimental results on Enron email dataset are reported as the F1 values over the seven users. As shown in Figure 3, GSP-PCL achieves consistently significant improvements on seven users compared with GSP-means and K-means clustering.

As shown in Figure 3, the clustering performance varies a lot on the seven users' dataset, which depends on the level of complexity and homogeneity of each dataset. Such an observation is in consistent to the classification results reported by Bekkerman, McCallum, and G. Huang in their work [1].

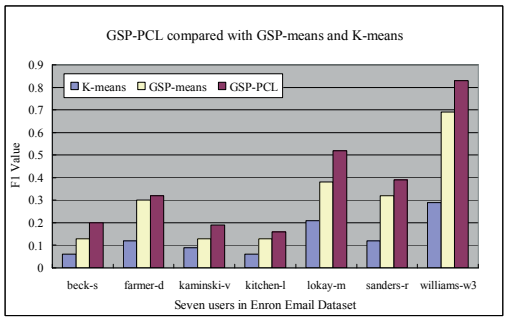

Figure 3. Performance of GSP-PCL, GSP-means and K-means on Enron email dataset (in terms of F1).

3.3.2. Experimental results on private email dataset. Experimental results on the private email dataset are shown in Figure 4 in terms of F1. Similar to the observation on Enron dataset, we can see that GSP-PCL achieves consistently significant improvements on four users compared with GSP-means and K-means clustering. Another observation from Figure 4 is that GSP-means achieves some improvements over the basic K-means algorithm, especially on user1 and user4, which proves the effectiveness and usefulness of the GSPs.

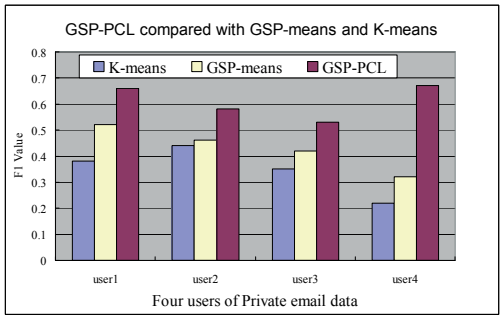

Figure 4. Performance of GSP-PCL, GSP-means and K-means on private email dataset (in terms of F1).

3.4. Cluster Naming

In GSP-PCL, cluster names are generated as follows: if emails in one cluster match one or more GSP groups, the cluster is named by the GSP with the highest support, otherwise it is named by the top five words sorted based on the scores computed as follows:

$$Score(t_k) = \frac{\sum_{d_i \in C_j} t_{ki}}{\ln\left(1 + \sum_{d_i} t_{ki}\right)} \qquad (3)$$

where C_j denotes the cluster, d_i is an email, t_k is a word, and t_{ki} is the weight of word t_k in the email d_i. For the basic K-means algorithm, clusters are named using ranked word features. Some sample cluster names generated through GSPs and word features are shown in Table 3.

Table 3. Sample cluster names generated from GSPs and ranked word features.

	CLASS SEMANTICS	GSP	WORD FEATURES
1	Interview schedule	Interview *date*	confirm accommodate lunch convenient attend
2	Talks given by some persons	talk *person*	venue talk speaker room Dr
3	Introducing somebody	Welcome *person*	welcome join degree joined university
4	Analysis on PSS log	analysis PSS log	associations excerpt divides structuralize surprise
5	Paper review	paper review	paper dear IEEE papers review

From Table 3, we can see that GSPs capture and summarize the contents of the clusters more precisely and compactly than word features. For sample 1, 2 and 3, the GSPs contain factoids generated by NLPWin, which not only makes the GSPs much easier to understand, but also makes the discovery of such GSPs possible. For sample 1 and 4, it is difficult to understand the names generated from word features if the users do not read the email contents. In sample 2 and 5, although the ranked word features contain enough information but they also contain some noisy words.

Table 4. Readability scores by owners and seven experimenters

Datasets		Owners	Experimenters Average
User 1	K-means	1.63	1.81
	GSP-PCL	2.80 (+71.3%)	2.44 (p=0.005782)
User 2	K-means	1.97	2.02
	GSP-PCL	2.71 (+37.7%)	2.58 (p=0.006793)
User 3	K-means	1.05	1.73
	GSP-PCL	2.42 (+130%)	2.43 (p=0.0008237)
User 4	K-means	1.72	1.72
	GSP-PCL	2.31 (+34.0%)	2.18 (p=0.0589)
Improvement		68.5%	32.4%

We conducted an experiment to evaluate the cluster names generated from GSPs and ranked word features. The cluster names generated by GSP-PCL and the basic K-means algorithm on each data set were evaluated by the owner of the dataset via readability scores from 1 (un-readable) to 3 (clear). Additionally, seven experimenters were invited to help evaluate the readability of the cluster names on all data sets. The seven experimenters are unaware of our algorithm. T-test is performed on the seven experimenters' scores. The results are shown in Table 4. We can see that the names generated from GSPs are more readable in both the viewpoints of owners and other experimenters. The improvements are statistically significant according T-test results.

4. CONCLUSIONS

In this paper, we proposed a novel approach to automatically extract embedded knowledge from the email subjects to help improve email clustering. Natural language processing technique and the frequent closed itemset mining technique are employed to generate generalized sentence patterns (GSP for short) from email subjects, which can be used to assist clustering as well as serve as good cluster descriptors. To leveraged the discovered GSPs, a novel unsupervised approach is proposed, which treats GSPs as pseudo class labels and classifies emails using a supervised learning algorithm (although no human labeling is involved). The experimental results showed that GSP-PCL obtains significant improvements both on the cluster quality and cluster name readability compared with the basic K-means algorithm.

5. Acknowledgement

The authors would like to thank Dr. Guimei Liu and the anonymous reviewers for their valuable comments and suggestions.

6. REFERENCES

[1] R. Bekkerman, A. McCallum, and G. Huang, Automatic categorization of Email into folders: Benchmark experiments on Enron and SRI Corpora. UMass CIIR Technical Report IR-418, 2004.

[2] G. Celeux and G. Govaert. Comparison of the Mixture and the Classification Maximum Likelihood in Cluster Analysis. *Journal of Statistical Computa-tion and Simulation*. 47:127-146, 1993.

[3] M. Halkidi, M. Vazirgiannis. An Introduction to Quality Assessment in Data Mining. PKDD. 2002.

[4] G. Heidorn. Intelligent writing assistance. in *Handbook of Natural Language Processing*, R. Dale, H. Moisl, and H. Somers, Eds. Marcel Dekker, New York, 1999.

[5] K. Nigam, A. McCallum, S. Thrun, T. M. Mitchell. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning* 39(2/3): 103-134. 2000.

[6] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. ICDT, 1999.

[7] J. Wang, J. Pei, and J. Han. Closet+: Searching for the best strategies for mining frequent closed itemsets. *SIGKDD*, 2003.

[8] http://www.cs.cmu.edu/~enron/