Clustering Main Concepts from e-Mails

Jesús S. Aguilar-Ruiz[1], Domingo S. Rodriguez-Baena[1],
Paul R. Cohen[2], and Jose Cristóbal Riquelme[1]

[1] Department of Computer Science, University of Seville, Spain
{aguilar,dsavio,riquelme}@lsi.us.es
[2] Intelligent Systems Division, ISI, University of South California, USA
cohen@isi.edu

Abstract. E–mail is one of the most common ways to communicate, assuming, in some cases, up to 75% of a company's communication, in which every employee spends about 90 minutes a day in e–mail tasks such as filing and deleting. This paper deals with the generation of clusters of relevant words from E–mail texts. Our approach consists of the application of text mining techniques and, later, data mining techniques, to obtain related concepts extracted from sent and received messages. We have developed a new clustering algorithm based on neighborhood, which takes into account similarity values among words obtained in the text mining phase. The potential of these applications is enormous and only a few companies, mainly large organizations, have invested in this project so far, taking advantage of employees's knowledge in future decisions.

1 Introduction

Ontologies have shown their usefulness in application areas such as intelligent information integration, information brokering and natural–language processing [15]. We can represent the knowledge existing in a domain using a conceptual diagram composed by a group of objects and the relations among them [4]. This sample of knowledge, created from a set of relational terms, from a specific vocabulary, is the aspect of ontology on which our research is focused.

Information Extraction systems were designed to filter, to select and to classify the increasing amount of information available nowadays, mainly on the Web. Most of them were based on shallow natural language processing techniques, but semantics was not really used, due to the unavailability of generic ontologies. Important efforts concentrate on developing tools for semi–automatic building of domain–specific ontologies, mainly based on text mining techniques.

Nowadays, a number of studies and techniques focus on textual information contained in electronic documents (e–mails, presentations, technical reports, etc.) [13]. Text can be a rich source of information, but this information is coded in such a way that decoding it becomes quite difficult. Learning, natural language processing, information extraction and mathematical approaches have been combined to decode and extract the content of texts [8].

The objective of this research is to extract ontological information from email using text and data mining techniques. In this paper, our goal is not to construct

R. Conejo et al. (Eds.): CAEPIA-TTIA 2003, LNAI 3040, pp. 231–240, 2004.

ontologies, but rather, to find groups of concepts that are commonly discussed in email. Email with family members involves a different set of concepts than email with colleagues in computer science, or with administrative assistants, or automobile mechanics. One way to extract concepts is to look for words that "go together" in text. If words co–occur more often than one would expect by chance, it may be because these words refer to one or more related concepts. These groups of concepts can provide us with an idea about what topics are in texts, making possible to organize the knowledge of users in order to take advantage of it for future decisions.

The document is organized as follows: in Section 2 we will briefly justify and show the current interest in this research; in Section 3 the entire system is described, using a simple example throughout every step; the experiments will be presented and discussed in Section 4; finally, the most important conclusions will be summarize in Section 5.

2 Motivation

E–mail has turned into one of the most common ways of communication in the last few years. Recent studies show that e–mail can make up to 75% of the company's communication, in which every employee spends about 90 minutes a day in organizing e–mail–tasks such as filing and deleting. The number of sent and received messages increase between 35% and 50% every year [10]. For comparison, US corporations spend roughly $1.5 trillion per year, only counting averaged salaries for the time workers spend reading, replying to, and organizing their e–mail. However, the entire US military budget in 2002 was $ 360 billion [6]. The knowledge extracted from e–mails can help us to organize, by subject or importance, the information handled by a group, or to categorize the employees of a company according to the content of their e–mails, allowing us, for instance, to locate a person specialized in data mining because the words data mart or clustering are common in his e–mails [2].

Researchers at Hewlett Packard have been experimenting with analyzing the flow of 185.773 e–mails among 485 users in an organization over a two–month period, concluding that it is possible to identify the power structure of an organization, communities (both known and unknown), and the leadership within these groups [1]. On a practical level, managers, for example, might use information gleaned from email studies to help businesses run more smoothly by making sure teams are communicating effectively and determining who is collaborating on certain projects. That study only examined the headers of emails, as already did Schwartz and Wood in 1993 [14], by mining 1.2M email headers to detect interests between people using graph theory.

The potential of the applications derived from obtaining useful knowledge from the textual information contained in e–mails is enormous [9]. For instance, KnowledgeMail [7], is able to create a user profile that can locate an expert on a specific topic when his/her knowledge is needed by other member of a company. Logically, these types of applications start to make sense in large companies, in which the use of the knowledge generated can result in an important time and capital saving for the company. That is the case of the Central Intelligence Agency, that invested $1 million in knowledge–management software developer Tacit Knowledge Systems, and between $1 and $5 millions in Stratify's software to mine unstructured data from e–mail systems, web pages, etc., at the end of 2001 [3].

Not only private companies, but also some universities are becoming interested in this field; for example, Carnegie Mellon University is currently involved in a research project dealing with the intelligent treatment of e–mail.

3 Description

We process the information in email messages in a sequence of steps, beginning with text mining and then data mining. The steps of the system are the following:

1. Preprocessing: our knowledge base will be composed of words in messages so we will have to filter the least relevant elements from texts: punctuation marks, language elements such as articles, pronouns, conjunctions, etc.
2. Text Mining: we will apply a method designed to obtain relations among words to the data set we have obtained from the previous filtering. To do this, it is necessary that we observe the similarity between words. The calculation of similarity is based on proximity frequency of words in text. That is, two words are similar if they appear one next to each other more than it is statistically expected.
3. Data Mining: given a set of lists, obtained in the previous step and formed by pairs of words and their similarities, we will apply data mining techniques to generate conceptual groups. In particular, we will use a clustering technique, specifically developed for this project, based on the neighbourhood concept.

In Figure 1 we show the process described previously, which has a set of e–mails as input and ontologies as output.

3.1 Preprocessing

The goal of this step is to remove irrelevant information, so a simple filter algorithm is applied. A set of strings with little semantic information is deleted from the text. Among them, are punctuation marks, articles, pronouns and some high-frequency, low content words. Afterwards, the text is semantically denser, as very related words will be closer to each other due to the deletion of other irrelevant words in between.

3.2 Text Mining

There are many ways to associate words within a text [8]. We will use a sliding window of size K throughout the text. The content of each window will inform about the statistical relationship among its words.

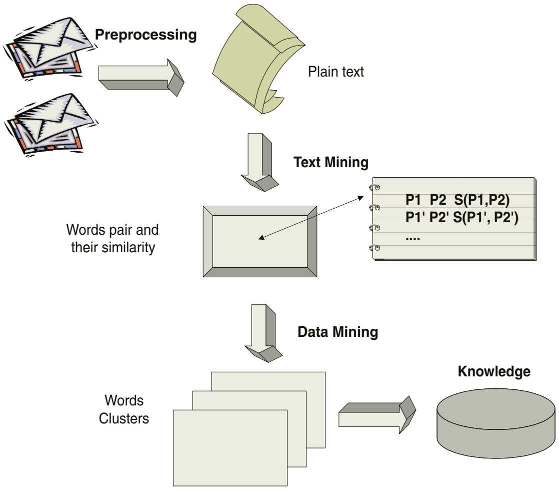

Fig. 1. Providing knowledge from e–mails.

Let P_1 and P_2 be two words. If they appear near each other more than one would expect by chance, we say that P1 and P2 are similar. To measure this similarity Ψ, let us suppose that P_1 and P_2 are separated by K words in a text sequence. Suppose also that there are $f_1 = 1$ appearances of P_1 in a sequence of length N. Then we can model the probability that P_2 falls within K words of P_1 as follows: the probability that a random location in the sequence falls within K words of P_1 is $p = \frac{2K}{N}$. Suppose there are $f_2 = 5$ occurrences of P_2. The probability that exactly δ of them will fall within K words of P_1 is just a binomial probability where the number of events is $f_2 = 5$, the number of successes is δ, and p is as described earlier. That is $P(X = \delta) \sim Binomial\left(N, \frac{2K}{N}\right)$. This is easily generalized to $f_1 > 1$ because $p = \frac{2Kf_1}{N}$, and then $P(X = \delta) \sim Binomial\left(N, \frac{2Kf_1}{N}\right)$.

We take $K = 10$ and then use a window with $2K+1$ positions (K on both the left and the right hand of the word being analyzed). This method will generate the similarity values for each pair of words. At the end, a file composed by a set of (P_1,P_2,Ψ), in which P_1 and P_2 are the two words and Ψ is the similarity value calculated as described earlier.

3.3 Data Mining

The final goal of this process is to obtain sets or clusters of words, so the existing relationship among members of the same cluster is based on the similarity. This phase can be divided in two steps:

1. Preprocessing: in which the similarity file generated in the previous step is processed to create a data structure that contains the information organized in such a way that the data mining technique can deal with it properly.

2. Clustering: we have developed a new neighborhood–based clustering technique adapted to the features of these data. This technique will provide a set of related words clusters from which we will study their inter–relationships to provide ontologies.

Preprocessing. The similarity value Ψ provides us an idea about the relationship between two words in the text. Because Ψ is usually very small we will use logarithms, so the numbers will be negatives. That is, larger negative value means greater similarity. After a preliminary study, we observe that there are pairs of words with very high similarity. This words used to appear at the bottom of messages as the signature (sender name, address, organization, etc.). We set a threshold, obtained from the normal distribution, based on the mean and the deviation, and eliminate a number of pairs composed by frequents words in texts, mainly associated with the sender.

In the next step, we generate a data structure in which every word is added to a word list, ordered by similarity. These lists have a variable length and give us an idea about the context of each word in the text. This data structure will serve as input for the clustering algorithm, called Snn (Similar Nearest Neighbour), which is based on the word neighbourhood and is described next.

Clustering. Our approach to clustering, the Snn (Similar Nearest Neighbour) algorithm has three main features:

- Snn is deterministic and its results do not depend on the order in which it is presented items. Many clustering algorithms do not have this property.
- Snn starts taking as initial set of representative patterns all of them at once. Next, it will join related patterns until the algorithm ends, following an incremental and hierarchical criterion. Other algorithms take a subset of representative patterns, so results might vary depending on the quality of the initial selection.
- Snn has no input user–defined parameters. The vast majority of clustering algorithms need user parameters. The number of clusters required by the user is the most common, as in K–means clustering algorithm [12] (this non–hierarchial method initially takes the number of components of the population equal to the final required number of clusters). In fact, some works have tried to find good methods to initialize the K–means algorithm [11]. Nevertheless, there are some others like the number of representative examples, as in Cure [5]. Our algorithms Snn provides automatically the most suitable number of clusters.

To describe the clustering algorithm, we will first provide some definitions.

Let e be a word, we say that the enemy of e is the first word in the list of associated words ordered by similarity that surpasses the threshold λ, previously set. The neighbourhood N_s of a word e is the set of words which are nearer e than the enemy of e, that is, their similarities are lower than enemy's similarity. The neighbourhood N_C of a cluster C is the set composed by all the neighbourhoods of each word belonging to the cluster C. $N_C(C) = \bigcup_{e \in C} N_s(e)$. Two clusters,

```
Procedure S–NN(in: $E$, $\lambda$; out: $RSC$)
  $SC := \emptyset$
  for each $e_i \in E$, $1 \leq i \leq |E|$
    $RSC[i] := \{e_i\}$
    $NSC[i] := N_s\{e_i, \lambda\}$
  end for
  $SC := RSC$
  $RSC := Reduction(SC, NSC)$
  while $SC \neq RSC$ do
    for each $C_k \in RSC$, $1 \leq k \leq |RSC|$
      $NSC[k] := N_c(C_k)$
    end for
    $SC := RSC$
    $RSC :=$Reduction$(SC, NSC)$
  end while
end S–NN

Function Reduction(in: $C$, $S$; out: $R$)
  $R := C$
  for each pair (i,j), with $1 \leq i \leq j \leq |C|$
    if $S[i] = S[j]$
      $R[i] := R[i] \cup C[j]$
      remove $R[j]$
    end if
  end for
end Reduction
```

Fig. 2. S–NN algorithm.

C_1 and C_2, will be cluster–neighbours if $N_C(C_1) = N_C(C_2)$. A reduced set of clusters RSC is a subset of clusters from the original set of clusters in which any two clusters are not neighbours, i.e, any two cluster–neigbours have been joined. We will identify each cluster in the reduced set of clusters RSC as the patterns that belong to the cluster, i.e., the patterns that were obtained by means of unions of neighbour–clusters.

Once the necessary definitions to support the algorithm have been presented, we will describe the algorithm depicted in Figure 2.

The input parameter E represents the structure obtained by the similarity search algorithm applied to the initial plain text, that is, pairs of words with their similarity value, and the output parameter is RSC, the reduced set of clusters, where each one comprises a group of instances.

$$P_1, P_2, \Psi_{1,2} = Similarity(P_1, P_2)$$

$$P_1, P_3, \Psi_{1,3} = Similarity(P_1, P_3)$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$P_{n-1}, P_n, \Psi_{n-1,n} = Similarity(P_{n-1}, P_n)$$

The input parameter is E, containing the instances, SC is an auxiliary set of clusters and RSC is initially set with clusters containing only one instance (lines 2–6). After initializing RSC and obtaining every word's neighbourhood, we apply a first reduction of this set of clusters. This is done because we need to take into account the value of λ in the first reduction. As we can see, the first time NSC is calculated (line 5), λ is present. However, next calculations do not take into account this value. This is not a parameter of the clustering algorithm per se, but a threshold to filter some words in the initial lists. When K is large, it is recommended to reduce the value of λ.

The process is repeated until RSC has no change at an iteration (line 9). The neighbourhood of every cluster is calculated (lines 10–12) in order to analyze the possible reduction of the set of cluster, task done by the Reduction function (line 14). The reduction of a set of clusters follows the next criterion: one cluster will be removed (line 22) if there exists another cluster which has exactly the same neighbourhood (line 20). In this case, the members of both clusters are joined (line 21).

The idea behind the algorithm is very simple: two clusters are neighbours if they have exactly the same neighbours. Obviously, the concept of neighbourhood is limited by the participation of the enemy, which indicates what neighbours each cluster has. The criterion used in this paper could be relaxed in two ways: considering the reduction when the neighbours of a cluster are a subset of the neighbours of the other or when the intersection among the neighbours of the two clusters is non–empty. As for the first as the second variation, the number of clusters is even smaller. The experiments shown in this paper were carried out by using the original criterion: the equality. However, we are going to study these other two criteria in further research.

4 Experiment

To illustrate the method, we have designed a simple practical example based on two emails. They represent the conversation between a professor of a university department and his secretary about making a reservation to a flight. The aim is to obtain relevant words within clusters generated with the proposal technique. The e–mails are the following:

> *Good Morning Maggie,*
> *I'd like to book a flight from London to NYC for Thursday evening, about seven. I must be at the University to give a talk related to the Argos project the following morning. I'd really prefer a nonstop flight, because the last time I took a connecting flight, it left late and I missed the connection.*
> *Don't worry about the hotel reservation. I'm going to spend the weekend in a friend's house. Thank you. –P*
>
> *Good Morning professor,*
> *I have been talking to the travel agency and there is a nonstop flight at 9:31 from Heathrow. Is that too late? One more question, will you be flying coach or you prefer business class? Well, I'll try to find a seat available in business class, ok? And finally, I charge it to the Argos Project, I suppose. Well, let's see what I can come up with. Maggie.*

Firstly, the textual information of both emails is filtered, eliminating elements not interesting such us commas, points, articles, pronouns, conjunctions, etc. The result of this operation is a plain text in which all the relevant words are put together, keeping the same order than in the original emails.

> GOOD MORNING MAGGIE LIKE BOOK FLIGHT LONDON NYC THURSDAY EVENING SEVEN MUST BE UNIVERSITY GIVE TALK FOLLOWING MORNING RELATED ARGOS PROJECT PREFER NONSTOP FLIGHT LAST TIME TOOK CONNECTING FLIGHT LEFT LATE MISSED CONNECTION WORRY HOTEL RESERVATION GOING SPEND WEEKEND FRIEND HOUSE THANK GOOD MORNING PROFESSOR ....

In the next step, we calculate the similarity by passing a window of length $2K+1$ (with $K = 10$) through the plain text. For each pair of words we provide a value of similarity. The final result is other file with the structure shown in Table 1: two words and the similarity between them.

Table 1. Table with pairs or words and their similarity.

word	word	similarity
GOOD	MORNING	-18.086
GOOD	FLIGHT	-4.873
PROJECT	TALK	-3.023
CLASS	BUSINESS	-9.620
FLIGHT	PROFESSOR	-6.413
TRAVEL	UNIVERSITY	-3.880
FLIGHT	NONSTOP	-3.560
GOOD	BOOK	-3.009
COACH	SEAT	-5.324
PROFESSOR	MORNING	-6.837
...	...	...

Now, we apply the first part of the algorithm, as a previous step for the clusters generation by Snn. We calculate the initial neighbourhoods, which is shown in Table 2.

Table 2. Words and associated neighborhood.

[*GOOD*]	[MORNING, FLIGHT,BOOK]
[*FLIGHT*]	[GOOD, FLIGHT, NONSTOP,...]
[*PROFESSOR*]	[FLIGHT, MORNING, BOOK, ...]
...	...

Next, clusters with similar neighbourhood are joined and the new neighbourhood of each cluster is calculated. For example, [GOOD] and [PROFESSOR] have the same neighbourhood, so they will pass joined to the next iteration. In addition , the neighbourhood of [GOOD, PROFESSOR] will contain the instance NONSTOP (because it is neighbour of FLIGHT at iteration 1). For this reason we will find that [GOOD, PROFESSOR]=[MORNING, FLIGHT,BOOK]+[NONSTOP]. Iteration 3 does the same: firstly it searches for possible joining and afterwards calculates the new neighbourhood for each cluster. In this way the iterations are repeated: calculating similarity between clusters, reducing the number of clusters and increasing the neighbourhood of the new clusters generated. The process ends when there is no modification of clusters at one iteration, so the termination criterion is natural and totally independent of the user, removing this parameter, very common in the great majority of clustering algorithms.

Table 3. Clusters from 57.817 words collected from e–mails.

C1: [ARRIVE BALTIMORE DEPART ECONOMY COACH **FLIGHT** HARTFORD TERMINAL WASHINGTON]
C2: [ABILITY FIELD **INVESTIGATION** LANGUAGES MAINTAINING PROFESSIONAL PROGRAMMING PUBLICATIONS SPECIFIC THEORETICAL]
C3: [ACCOUNT **FUNDING** GOVERNOR GRANT INVESTIGATOR REQUIRED]
C4: [ACQUIRING BEHAVIOR DIRECTLY **MILITARY** TACTICS]
C5: [ASSISTANTS ASSOCIATION CONDUCT DESIGNS **FACULTY** IMPLEMENTS MEMBER]
C6: [ASSISTS COORDINATION EQUIPMENT **LAB** RESPONSIBLE]
C7: [**CARD** CREDIT DEBIT REMEMBER]

Finally, we have a reduced set of clusters as a result, each cluster containing the words that have certain degree of similarity. For instance, in our example there would be the following final clusters:

[GOOD PROFESSOR FLIGHT BOOK HOUSE...]
[TALK ARGOS UNIVERSITY PROJECT...]
[HOTEL FRIEND WORRY ...]
...

Once the clusters have been obtained, we can analyze the relevant concepts based on words within them. In each cluster we detect the most relevant elements as the one that has the biggest value of similarity with respect to the rest of words in the cluster. In Figure 3, the two main concepts and their relations are shown: project and flight. The elements associated with them are in the cluster, not being the most relevant ones.

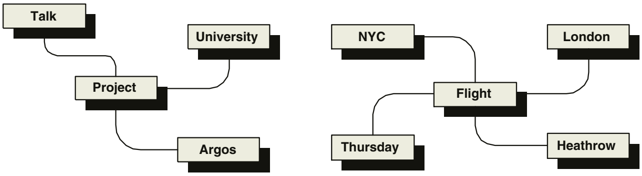

Fig. 3. Main concepts.

The example designed to explain the process has very few words. However, the system has been proven in a real organization and with real e–mail messages. The text obtained, after applying the first filter over punctuation marks, articles, etc, contained more than 10.000 different words, that generated a file with 57.817 pairs of related words, with their respective similarity values. A summary of the results is shown in Table 3, in which appear the most significant clusters. The words in bold represent the main elements of every cluster.

5 Conclusions

In this paper we addresses a problem that is becoming considered relevant in large organizations: the generation of clusters from E–mail texts. The objective of this research consists of extracting useful knowledge represented by clusters from textual information contained in a large number of emails using text and data mining techniques. Our approach consists of the application of text mining techniques to filter spurious words and find similarities among words and, later, data mining techniques, to obtain relational concepts, grouped in clusters, and extracted from the sent and received messages electronically. A new neighborhood–based clustering algorithm, Snn, is also introduced in this paper. Experiments generated from 57.817 pairs of related words show the quality of our approach.

References

1. L.M. Bowman. Email flow can show company power structure. *ZD Net UK News*, March 2003.
2. P. W. Eklund and R. Cole. Structured ontology and information retrieval for email search and discovery. *ISMIS*, pages 75–84, 2002.
3. E. Goodridge Intelligence Agency Bets On Knowledge Management. *InformationWeek*, Dec. 3, 2001.
4. T. Gruber. Towards principles for the design of ontologies used for knowledge sharing. *Intl. J. of Human and Computer Stud-ies*, 2/3(46):293–310, 1997.
5. S. Guha, R. Rastogi and K. Shim. CURE: an efficient clustering algorithm for large databases. *ACM SIGMOD International Conference on Management of Data*, pages 73–84, 1998.
6. J. Dao. Pentagon Seeking A Large Increase In Its Next Budget. *New York Times*, January 7, 2002.
7. KnowledgeMail. *http://www.tacit.com*, 2003.
8. C. D. Mannig and H. Schutze. *Foundations of Statistical Natural Language Processing.* MIT Press, 1999.
9. E. Moreale and S. Watt. Organisational information management and knowledge discovery in email within mailing lists. *IDEAL*, pages 87–92, 2002.
10. V. Murphy. You've Got Expertise. *Forbes Magazine*, February, 2001.
11. J. M. Pena, J. A. Lozano and P. Larranaga. An empirical comparison of four initialization methods for the k-means algorithm. *Pattern Recognition Letters*, 20, 1027–1040, 1999.
12. D. Pollard. Quantization and the method of k-means. *IEEE Transactions on Information Theory*, 2/3(46):293–310,28(2):199–205, 1982.
13. F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 1(34):1–47, 2002.
14. M. Schwartz and D. Wood. Discovering shared interests among people using graph analysis of global electronic mail traffic. *Communication of the ACM*, 36(8):1–47, 1993.
15. G. Wiederhold and M. Genesereth. The conceptual basis for mediation services. *IEEE Expert / Intelligent Systems*, 12(5):38–47, September/October 1997.