Text Mining at the Term Level

Ronen Feldman[1], Moshe Fresko[1], Yakkov Kinar[1],Yehuda Lindell[1], Orly Liphstat[1], Martin Rajman[2], Yonatan Schler[1], Oren Zamir[3]

[1] Department of Mathematics and Computer Science, Bar-Ilan University, Ramat-Gan, Israel
{feldman, fresko, kinary, ylindell, okatz, schler}@cs.biu.ac.il
[2] Artificial Intelligence Laboratory (LIA), Swiss Federal Institute of Technology, Lausanne, Switzerland
martin.rajman@epfl.ch
[3] Department of Computer Science, University of Washington, Seattle, WA
zamir@cs.washington.edu

Abstract. Knowledge Discovery in Databases (KDD) focuses on the computerized exploration of large amounts of data and on the discovery of interesting patterns within them. While most work on KDD has been concerned with structured databases, there has been little work on handling the huge amount of information that is available only in unstructured textual form. Previous work in text mining focused at the word or the tag level. This paper presents an approach to performing text mining at the term level. The mining process starts by preprocessing the document collection and extracting terms from the documents. Each document is then represented by a set of terms and annotations characterizing the document. Terms and additional higher-level entities are then organized in a hierarchical taxonomy. In this paper we will describe the Term Extraction module of the Document Explorer system, and provide experimental evaluation performed on a set of 52,000 documents published by Reuters in the years 1995-1996.

1 Introduction

Traditional databases store information in the form of structured records and provide methods for querying them to obtain all records whose content satisfies the user's query. More recently however, researchers in *Knowledge Discovery in Databases* (KDD) have provided a new family of tools for accessing information in databases [1,3,15,19,20]. The goal of such work, often called *data mining*, has been defined as "the nontrivial extraction of implicit, previously unknown, and potentially useful information from given data" [15]. Work in this area includes applying machine-learning and statistical-analysis techniques towards the automatic discovery of patterns in databases, as well as providing user-guided environments for exploration of data.

Most efforts in KDD have focused on knowledge discovery in structured databases, despite the tremendous amount of online information that appears only in collections of unstructured text. Previous approaches to text mining have used either tags attached to documents [11,12,13] or words contained in the documents [21].

Standard Text Mining systems do not usually operate on unprepared documents but rather on “categorized documents”, i.e. documents that were (manually or automatically from a set of examples) tagged with terms identifying their content. Such systems can of course be extended to use the full text of the document by systematically tagging the documents with *all* the words they contain. This process however, does not provide effectively exploitable results, as has been shown for association generation [22]. For example, in the experiment just mentioned, the association generation process detected either compounds, i.e. domain-dependent terms such as [wall, street] or [treasury, secretary, james, baker], which cannot be considered potentially useful associations or extracted uninterpretable associations such as [dollars, shares, exchange, total, commission, stake, securities] that cannot be considered easily understandable.

The exploitation of untagged, full text documents therefore requires some additional linguistic pre-processing, allowing the automated extraction from the documents of linguistic elements more complex than simple words. We use *normalized terms*, i.e. sequences of one or more lemmatized word forms (or lemmas) associated with their part-of-speech tags. “stock/N market/N” or “annual/Adj interest/N rate/N” are typical examples of such normalized terms.

In this paper, we present our approach to text mining, which is based on extracting meaningful terms from documents. The system described in this paper begins with collections of raw documents, without any labels or tags. Documents are first labeled with terms extracted directly from the documents. Next, the terms and additional higher-level entities (that are organized in a hierarchical taxonomy) are used to support a range of KDD operations on the documents. The frequency of co-occurrence of terms can provide the foundation for a wide range of KDD operations on collections of textual documents, such as finding sets of documents whose term distributions differ significantly from that of the full collection, other related collections, or collections from other points in time.

The focus of this paper is on the Term Extraction module of our term-based text mining system. In particular, we will describe the term extraction algorithms and the organization of the terms in a taxonomy. We begin this paper with the description of the Term Extraction module of the Document Explorer system. We then describe how we construct a hierarchical taxonomy of the extracted terms. Next, we present experimental results from the Reuters financial news from 1995-1996. We conclude by comparing the term-based approach to the tag-based approach and outline the strength and weaknesses of each.

2. The Term Extraction Module

The Term Extraction Module is responsible for labeling each document with a set of terms extracted from the document. An example of the output of the Term Extraction module is given in Fig.1. The following excerpt is taken from an article published by Reuters Financial on 5/12/96. Terms in this excerpt that were identified and designated as interesting by the Term Extraction module are underlined.

> Profits at Canada's six big banks topped C$6 billion ($4.4 billion) in 1996, smashing last year's C$5.2 billion ($3.8 billion) record as Canadian Imperial Bank of Commerce and National Bank of Canada wrapped up the earnings season Thursday. The six banks each reported a double-digit jump in net income for a combined profit of C$6.26 billion ($4.6 billion) in fiscal 1996 ended Oct. 31.
>
> But a third straight year of record profits came amid growing public anger over perceived high service charges and credit card rates, and tight lending policies.
>
> Bank officials defended the group's performance, saying that millions of Canadians owned bank shares through mutual funds and pension plans.

Fig. 1. Example of the output of the Term Extraction Module. Terms chosen to label the document are underlined.

The overall architecture of the Term Extraction module is illustrated in Fig.2. There are three main stages in this module: Linguistic Preprocessing, Term Generation and Term Filtering.

The documents are loaded into the system through a special reader. The reader uses a configuration file that informs it of the meaning of the different tags annotating the documents. In such a way, we are able to handle a large variety of formats. The TPL reader packages the information into a standardized SGML file.

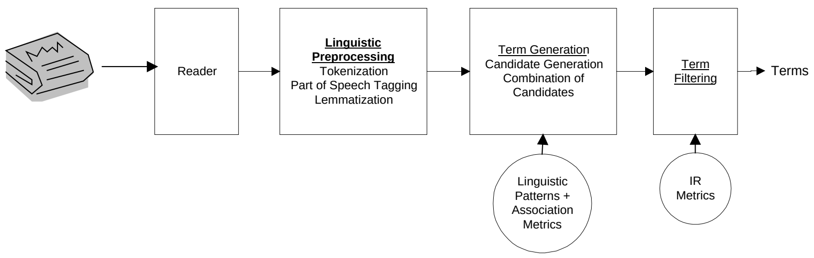

Fig. 2. Architecture of the Term Extraction Module

The next step is the Linguistic Preprocessing that includes Tokenization, Part-of-Speech tagging and Lemmatzations (i.e., a linguistically more founded version of

Stemming [17]). The objective of the Part-of-Speech tagging is to automatically associate morpho-syntactic categories such as *noun, verb, adjective*, etc., to the words in the document. In our system, we used a rule-based approach similar to the one presented in [4] which is known to yield satisfying results (96% accuracy) provided that a large lexicon (containing tags and lemmas) and some manually hand-tagged data is available for training.

The Term Generation and Term Filtering Modules are described in the following subsections.

Term Generation

In the Term Generation stage, sequences of tagged lemmas are selected as potential term candidates on the basis of relevant morpho-syntactic patterns (such as “Noun Noun”, “Noun Preposition Noun”, “Adjective Noun”, etc.). The candidate combination stage is performed in several passes. In each pass, association coefficient between each pair of adjacent terms is calculated and a decision is made whether they should be combined. In the case of competing possibilities (such as (t_1 t_2) and (t_2 t_3) in (t_1 t_2 t_3)), the pair having the better association coefficient is replaced first. The documents are then updated by converting all combined terms into atomic terms by concatenating the terms with an underscore. The whole procedure is then iterated until no new terms are generated.

The nature of the patterns used for candidate generation is an open research question. In [8] and [9] specific operators (such as overcomposition, modification, and coordination) are proposed to select longer terms by using combinations of shorter ones. In [7] candidate terms are taken to be Noun-Noun* sequences (i.e. Noun sequences of length 2 or more). This technique improved the precision but reduced recall. [18] suggests to accept prepositions as well as adjectives and nouns. This approach generate a much larger number of term candidates, while in [14] only (Noun|Adjective)-Noun sequences are accepted to reduce the amount of “bad” terms.

In Document Explorer we used two basic patterns: Noun-Noun and Adjective-Noun, but we also allowed the insertion of any kind of Determiner, Preposition or Subordinating Conjunction. Therefore sequences such as “health program for the elderly”, “networking software for personal computers”, “operating system of a computer” or “King Fahd of Saudi Arabia” are accepted as well.

We have tested 4 different association coefficients: co-occurrence frequency, ϕ^2, Association Ratio [5], and Log-Likelihood [9,10]. The co-occurrence frequency is the simplest association measure that relies on the number of times that the two terms match one of the extraction patterns. ϕ^2 has been used to align words inside aligned sentences [16] and for term extraction [9]. The Association Ratio was used for monolingual word association and is based on the concept of mutual information. Log-Likelihood is a logarithmic likelihood probability. Our candidate combination phase uses two thresholds. The first is a threshold T_{freq} for the co-occurrence frequency. The second is a threshold T_{metric} for additional filtering on the basis of a complementary association coefficient.

Term Filtering

The Term Generation stage produces a set of terms associated with each document without taking into account the relevance of these terms in the framework of the whole document collection. A consequence of this is a substantial over-generation of terms. Additional filtering is therefore necessary and several approaches can be tested.

The goal of the Term Filtering stage is to reduce the number of term candidates produced by the Term Generation stage on the basis of some statistical relevance-scoring scheme. After scoring all the terms generated in the Term Genration stage, we sort them based on their scores and select only the top M terms.

For Example, the following are all two-word terms that were identified in the Term Generation stage but later filtered out in the Term Filtering stage: `right direction`, `other issue`, `point of view`, `long way`, `question mark` and `same time`. These terms were determined not to be of interest in the context of the whole document collection either because they do not occur frequently enough or because they occur in a constant distribution among different documents.

We have tested 3 approaches for scoring terms based on their relevance in the document collection:

1. **Deviation-Based approach:** The rationale behind the deviation-based approach is the hypothesis, often used in lexicometry, that terms with a distribution uniform over a collection of documents correspond to terms with few semantic content (i.e., “uninteresting” words to be filtered out [2]). We use the standard deviation of the relative frequency of a term *t* over all the documents of the collection as its score.
2. **Statistical Significance approach:** The underlying idea is to test whether the variation of the relative frequency of a given term *t* in the document collection is statistically significant. This is done using the χ^2 significance test on the relative frequency of a given term *t*.
3. **Information Retrieval approach:** The notion of term relevance with respect to a document collection is a central issue in information retrieval [23]. We assign each term its score based on maximal *tf-idf* (term frequency - inverse document frequency, maximal with respect to all the documents in the collection).

An example of the results of the Term Filtering stage for tf-idf scores is given in Fig.3. The table shows the scores of the terms found in the excerpt given in Fig.1. Terms appearing in the shaded region were discarded in the Filtering stage.

Term	Score	Term	Score
net_income	17.17	record_profit	5.63
bank	14.88	canadian_imperial_bank_of_commerce	5.56
earnings	11.41	big_bank	5.39
canada	10.39	canadian	5.29
mutual_fund	8.22	lending	4.73
national_bank_of_canada	7.68	credit_card	4.56
bank_official	6.56	jump	4.03
pension_plan	6.34	season	3.86
profit	6.20	group	3.77
performance	6.09	policy	2.84
anger	5.82	share	1.50

Fig. 3. Scores of the terms found by the Term Generation stage for the document in Fig.1.

3. Taxonomy Construction

One of the crucial issues in performing text mining at the term level is the need for a term taxonomy. A term taxonomy enables the production of high level association rules which are similar to General Association Rules [24]. These rules capture relationships between groups of terms rather than individual terms. A taxonomy is also important in other text mining algorithms such as Maximal Association Rules and Frequent Maximal Sets [12].

A taxonomy enables the user to specify mining tasks in a concise way. For instance when trying to generate association rules, rather then looking for all possible rules, the user can specify interest only in the relationships of companies in the context of business alliances. In order to do so, we need two nodes in the term taxonomy marked "business alliances" and "companies". The first node contains all terms related to alliance such as “joint venture”, “strategic alliance”, “combined initiative” etc., while the second node is the parent of all company names in our system (we used a set of rules and knowledge extracted from WWW directories to generate company names).

Building a term taxonomy is a time consuming task. Hence we provide a set of tools for semi-automatic construction of such a taxonomy. Our main tool is a taxonomy editor. This tool enables the user to read a set of terms or an external taxonomy, and use them to update the system's term taxonomy. The user can drag entire subtrees in the taxonomies or specify a set of terms via regular expressions. In our case, the initial set of terms is the set of all terms extracted from the Reuters 52,000 document collection. The terms matching any user-specified pattern is represented as a subtree, and can be dragged to an appropriate place in the target taxonomy.

The taxonomy editor also includes a semi-auomatic tool for taxonomy editing called the Taxonomy Editor Refiner (TER). The TER compares generated frequent sets against the term taxonomy. When most of the terms of a frequent set are determined to be siblings in the taxonomy hierarchy the tool suggests adding the remaining terms as siblings as well. For example, if our taxonomy currently contains 15 companies under the "tobacco companies" and the system generated a frequent set containing many tobacco companies, one of which does not appear in the taxonomy, the TER will suggest adding this additional company to the taxonomy as a tobacco company. The TER also has a term clustering module again suggest that terms clustered together be placed as siblings in the taxonomy.

4. Experimental Evaluation

We have used 51,725 documents from the Reuters financial news of years 1995-1996. This collection is 120M RAM in size and contains over 170,000 unique words. Each document contained on average 864 words. In the Term Generation stage, 1.25M terms were identified (25M RAM), 154K of them unique. After the Term Filtering stage we were left with 975K term (approximately 45 terms per document), 16,847 of them unique. Our feature space was therefore reduced by more than a factor of 10 and the average document length was reduced by a factor of 20.

In the example presented below the user is interested in business alliances between companies. She therefore specifies a filter for the association rules generation algorithm, requesting only association rules with companies on the LHS of the rule and business alliance topics on the RHS.

Using the Reuters document corpus described above, Document Explorer generates 12,000 frequent sets that comply with the restriction specified by the filter (with a support threshold of 5 documents and confidence threshold of 0.1). These frequent sets generated 575 associations. A further analysis removed rules that were subsumed by other rules, resulting in a total of 569 rules. A sample of these rules is presented in Fig.4. The numbers presented at the end of each rule are the rule's support and confidence.

america online inc, bertelsmann ag ⇒ joint venture 13/0.72
apple computer inc, sun microsystems inc ⇒ merger talk 22/0.27
apple computer inc, taligent inc ⇒ joint venture 6/0.75
sprint corp, tele-communications inc ⇒ alliance 8/0.25
burlington northern inc, santa fe pacific corp ⇒ merger 9/0.23
lockheed corp, martin marietta corp ⇒ merger 14/0.4
chevron corp, mobil corp ⇒ joint venture 11/0.26
intuit inc, novell inc ⇒ merger 8/0.47
bank of boston corp, corestates financial corp ⇒ merger talk 7/0.69

Fig. 4. A sample of the association rules found by TextVis with companies on the LHS of the rule and business alliance topics on the RHS.

The example above illustrates the advantages of performing text-mining at the term level. Terms such as "joint venture" would be totally lost if we worked at the word level. Company names, such as "santa fe pacific corp" and "bank of boston corp", would not have been identified as well. Another important issue is the construction of a useful taxonomy such as the one used in the example above. Such a taxonomy cannot be defined at the word level as many logical objects and concepts are, in fact, multi-word terms.

5. Conclusions

Previous approaches to text mining have used either tags attached to documents or words contained in the documents. Tags were either assigned manually like in some of the on-line services (Dialog, Reuters), which is a very expensive and time consuming process, or by using machine learning algorithms. These text-categorization algorithms must be provided with a training set of pre-tagged documents. The main drawbacks of the machine learning approach are that it requires an expert to go and tag hundreds of training documents and that the accuracy is not high enough. The break-even point of these algorithms is below 80% [6]. This tag-based approach is characterized by a relatively small and controlled vocabulary. This has many implications on the results of the mining operations. On the one hand, the tags are meaningful and are often organized in a taxonomy, thus the mining results are often of high quality. On the other hand, much of the information present in the documents is not captured by the tags and thus is lost for the mining process.

Systems that use the full texts of the documents tend to produce a huge number of often meaningless results. In one example, the association generation process detected either compounds, i.e. domain-dependent terms such as {treasury,secretary,james} $\Rightarrow$ {baker}, or extracted uninterpretable associations [22]. There is an additional disadvantage when using the full text of the documents and that is the execution time and the memory requirements of the mining algorithms.

Term level text mining attempts to benefit from the advantages of these two extremes. On the one hand there is no need for human effort in tagging document, and we do not loose most of the information present in the document as in the tagged documents approach. Thus the system has the ability to work on new collections without any preparation, as well as the ability to merge several distinct collections into one (even though they might have been tagged according to different guidelines which would prohibit their merger in a tagged based system). On the other hand the number of meaningless results and the execution time of the mining algorithms are greatly reduced. Working on the term level also enables the construction (with the help of semi-automatic tools) of a hierarchical taxonomy which is extremely important to a text mining system. We are currently working on an empirical evaluation of the Term Extraction process in which we shall compare the results obtained by different methods to a set of terms designated as important by Human indexers.

One of the future directions is to use a hybrid approach that represents the document as a combination of tags and terms. In such a way we can benefit from both approaches.

References

1. Anand, T.; Kahn, G.: Opportunity Explorer: Navigating Large Databases Using Knowledge Discovery Templates. In: Proceedings of the 1993 workshop on Knowledge Discovery in Databases, (1993).
2. Bookstein, A.; Klein, S.T.; Raita, T.: Clumping Properties of Content-Bearing Words. In: Proceedings of International Conference on Research and Development in Information Retrieval (SIGIR), (1995).
3. Brachman, R. J.; Selfridge, P.G.; Terveen, L.G.; Altman, B.; Borgida, A.; Halper, F.; Kirk, T.; Lazar, A.; McGuinness, D.L.; Resnick, L.A.: Integrated Support for Data Archaeology. International Journal of Intelligent and Cooperative Information Systems, (1993)2(2):159-185.
4. Brill, E.: Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. Computational Linguistics, (1995) 21(4):543-565.
5. Church, K.W.; Hanks, P.: Word Association Norms, Mutual Information, and Lexicography. Computational Linguistics, (1990) 16(1):22-29.

6. Cohen, W.; Singer, Y.: Context Sensitive Learning Methods for Text categorization. In: Proceedings of International Conference on Research and Development in Information Retrieval (SIGIR), (1996).
7. Dagan, I.; Church K.W.: Termight: Identifying and Translating Technical Terminology. In: Proceedings of the European Chapter of the Association for Computational Linguistics, EACL, (1994) 34-40.
8. Daille, B.; Gaussier, E.; Lange, J.M.: Towards Automatic Extraction of Monolingual and Bilingual Terminology. In: Proceedings of the International Conference on Computational Linguistics (COLING), (1994) 515-521.
9. Daille, B.: Study and Implementation of Combined Techniques for Automatic Extraction of Terminology. In: Resnik, P.; Klavans, J. (eds.): The Balancing Act: Combining Symbolic and Statistical Approaches to Language, MIT Press, Cambridge, MA, USA, (1996) 49-66.
10. Dunning, T.: Accurute Methods for the Statistics of Surprise and Coincidence. Computational Linguistics, (1993) 19(1).
11. Feldman, R.; Hirsh, H.: Exploiting Background Information in Knowledge Discovery from Text. Journal of Intelligent Information Systems, (1996).
12. Feldman, R.; Aumann, Y.; Amir, A.; Klösgen, W.; Zilberstien, A.: Maximal Association Rules: a New Tool for Mining for Keyword co-occurrences in Document Collections. In: Proceedings of the 3rd International Conference on Knowledge Discovery (KDD), (1997).
13. Feldman, R.; Dagan, I.: KDT – Knowledge Discovery in Texts. In: Proceedings of the First International Conference on Knowledge Discovery (KDD), (1995).
14. Frantzi, T.K.; Incorporating Context Information for the Extraction of Terms. In: Proceedings of ACL-EACL, (1997).
15. Frawley, W.J.; Piatetsky-Shapiro, G.; Matheus, C.J.: Knowledge Discovery in Databases: an Overview. In: Piatetsky-Shapiro, G.; Frawley, W. J. (eds.): Knowledge Discovery in Databases, MIT Press, (1991), 1-27.
16. Gale, W.A.; Church, K.W.: Concordances for parallel texts. In: Proceedings of the 7th Annual Conference of the UW Centre for the New OED and Text Research, Using Corpora, (1991) 40-62.
17. Hull, D.: Stemming algorithms - a case study for detailed evaluation. Journal of the American Society for Information Science, (1996) 47(1):70-84.
18. Justeson, J.S.; Katz, S.M.: Technical Terminology: Some linguistic properties and an algorithm for identification in text. Natural Language Engineering, (1995) 1(1):9-27.
19. Klösgen, W.: Problems for Knowledge Discovery in Databases and their treatment in the Statistics Interpreter EXPLORA. International Journal for Intelligent Systems, (1992) 7(7):649-673.
20. Klösgen, W.: Efficient Discovery of Interesting Statements. The Journal of Intelligent Information Systems, (1995) 4(1).
21. Lent, B.; Agrawal, R.; Srikant, R.: Discovering Trends in Text Databases. In: Proceedings of the 3rd International Conference on Knowledge Discovery (KDD), (1997).
22. Rajman, M.; Besançon, R.: Text Mining: Natural Language Techniques and Text Mining Applications. In: Proceedings of the seventh IFIP 2.6 Working Conference on Database Semantics (DS-7), Chapam & Hall IFIP Proceedings serie, (1997) Oct 7-10.
23. Salton, G.; Buckley, C.: Term-weighting Approaches in Automatic Text Retrieval. Information Processing and Management, (1998) 24(5):513-523.
24. Srikant, R.; Agrawal, R.: Mining generalized association rules. In: Proceedings of the 21st Very Large Databases (VLDB), (1995).