USING GRAMMARS FOR TEXT CLASSIFICATION

P. Kroha and T. Reichel
Department of Information Systems and Software Engineering, TU Chemnitz, Strasse der Nationen, 09111 Chemnitz, Germany

Keywords: Text mining, text classification, grammars, natural language, stock exchange news, market forecast.

Abstract: In this contribution we present our experiments with using grammars for text classification. Approaches usually used are based on statistical methods working with term frequency. We investigate short texts (stock exchange news) more deeply in that we analyze the structure of sentences and context of used phrases. Results are used for predicting market movements coming from the hypotheses that news move markets.

1 INTRODUCTION

Currently, a growing amount of commercially valuable business news becomes available on the World Wide Web in electronic form. However, the volume of news is very large. Many of them have no importance but some of them may be very important for predicting market trends. The question is how to filter the important news from the unimportant ones and how much of this kind of information moves markets.

In our previous works we have been experimenting with methods of text classification that are based on frequency of terms to distinguish between positive news and negative news in terms of long-term market trends.

In this paper, we present how we have built a grammar that describes templates typical for specific groups of news stories. Each sentence in a news story is analyzed by a parser that determines the template to which the sentence belongs. Sentences and news are classified according to these assignments.

The novel approach is in using grammars and templates for text classification. Papers already published use statistical methods of classification based on term frequency. We discuss them and their shortcomings in Section 2.

The most crucial question is, of course, how to preprocess the news before extraction and before inputting the results into the classification engine. We investigate market news but our method of text classification presented in this paper can be used for any other purpose, too.

The rest of the paper is organized as follows. Related work is recalled in Section 2. Section 3 introduces concerned problems and Section 4 describes the implementation, Section 5 introduces our experimental data. Section 6 presents our experiments and achieved results. Finally, we conclude in Section 7.

2 RELATED WORKS

In related papers, the approach to classification of market news is similar to the approach to document relevance. Experts construct a set of keywords which they think are important for moving markets. The occurrences of such a fixed set of several hundreds of keywords will be counted in every message. The counts are then transformed into weights. Finally, the weights are the input into a prediction engine (e.g. a neural net, a rule based system, or a classifier), which forecasts which class the analyzed message should be assigned to.

In papers by Nahm, Mooney (Nahm, 2002) a small number of documents was manually annotated (we can say indexed) and the obtained index, i.e. a set of keywords, will be induced to a large body of text to construct a large structured database for data mining. The authors work with documents contain-

Kroha P. and Reichel T. (2007).
USING GRAMMARS FOR TEXT CLASSIFICATION.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - AIDSS*, pages 259-264
DOI: 10.5220/0002406302590264

ing job posting templates. A similar procedure can be found in papers by Macskassy (Macsakssy and Provost, 2001). The key to his approach is the user's specification to label historical documents. These data then form a training corpus to which inductive algorithms will be applied to build a text classifier.

In Lavrenko (Lavrenko et al., 2000) a set of news is correlated with each trend. The goal is to learn a language model correlated with the trend and use it for prediction. A language model determines the statistics of word usage patterns among the news in the training set. Once a language model has been learned for every trend, a stream of incoming news can be monitored and it can be estimated which of the known trend models is most likely to generate the story.

Compared to our investigation there are two different approaches. One difference is that Lavrenko uses his models of trends and corresponding news only for day trading. The weak point of this approach is that it is not clear how quickly the market responds to news releases. Lavrenko discusses this but the problem is that it is not possible to isolate market responses for each news story. News build a context in which investors decide what to buy or sell. Fresh news occur in the context of older news and may have a different impact.

In (Kroha and Baeza-Yates, 2005), the relevance of properties of large sets of news and long-term market trends was investigated using bags of news for classification. In (Kroha et al., 2006), the method was improved so that all news stories were separated from each other and the fine-grain classification was provided. The obtained results were of a new quality but the problems of statistical methods of classification still remain. In the next chapter we present our new solution.

3 GRAMMAR FOR NEWS TEMPLATES

Some features that are important for the classification are given by the sentence structure and not by the term frequency. In the example bellow, the both news stories have the same term frequency but completely different meaning.

Example 1:

News story 1: "XY company closed with a loss last year but this year will be closed with a profit".

News story 2: "XY company closed with a profit last year but this year will be closed with a loss.

(End of example)

There are grammatical constructions changing the meaning of a sentence that would be derived from phrases.

Example 2:

"Lexmark's net income rose 12 % but the company warned that an uncertain economy and price competition could weigh on future results."

(End of example)

Example 3:

"Lexmark's net income rose 12 % but it did not achieved the earnings expectations."

(End of example)

To overcome the problem presented above, i.e. two sentences may exist that have the same term frequency but completely different meaning, we have written a grammar describing grammatical constructions in English that usually bring positive or negative meaning to a sentence.

We collected news stories that can have an important influence (at least in our opinion) on markets and divided them accordingly to their content and structure into positive and negative ones.

Investigating the positive news we found phrases like e.g. new contracts, cost cutting, net rose, profit surged, jump in net profit, net doubled, earnings doubled, jump in sales, huge profit, income rose, strong sales, upgraded, swung to a profit.

In negative news we found e.g. accounting problems, decline in revenue, downgraded, drop in earnings, drop in net income, expectations down, net fell, net loss, net plunged, profile plunged, profit reduced, earnings decline, profits drop, slashed forecast, lowered forecast, prices tumbled.

We also investigated the sentence structure to classify news like: "Dell's profit fell 11% due to a tax charge, but operating earnings jumped 21%." or "EBay said earnings rose 44% but narrowly missed Wall Street expectations, sending shares down 12% in after-hours trading." We focused the parts of sentences that denote future event before last events because this is the way the investors do it.

In general, we can observe some repeating templates in news stories that can be used for association of news into groups and estimation of their meaning.

For example, we can distinguish the following templates:

1. New big contract
2. Acquisitions
3. Income rose
4. Income fell
5. Bad past but good future

6. Good past but bad future
7. .... etc.

For simplicity we can suppose that news that cannot be assigned to our templates are not interesting enough for forecasting. Based on templates we constructed our grammar.

A part of our grammar that solves the problem given in Example 1 is given bellow for illustration.

```
Start = Message;

Message = (Sentence)*;

Sentence = '.'  {c.add("empty");} | ...  |
  BadPastGoodPresent|GoodPastBadPresent|
  ...  | NotClassified ;

BadPastGoodPresent =
  Company NegSubject PastWord 'but'
  PresentWord PosSubject '.'
  {c.add("bPgP");} ;
  // match News story 1 as positive

GoodPastBadPresent =
  Company PosSubject PastWord 'but'
  PresentWord NegSubject '.'
  {c.add("gPbP");} ;
  // match News story 2 as negative

Company = ...  | 'XY' | ...  ;
PosSubject = ...  | 'profit' | ...  ;
NegSubject = ...  | 'loss' | ...  ;
PastWord = ...  |'last' 'year'| ...;
PresentWord = ...|'this' 'year'| ...;

allWord = Company | PosSubject | NegSubject
  | PastWord | PresentWord;
```

4 IMPLEMENTATION

In the implemented system, messages are classified in three steps that will be described in next subsections.

4.1 From the Grammar to the Parser

The BNF-form of the grammar has to be transformed into an executable version of parser. We used the tool Bex (Franke, 2000) that produced the source code of the corresponding ll(k)-parsers. This source code was adapted by a classification object ClassifyObj(`c`) that completed the grammar by semantic rules describing the classification of messages.

The classification object ClassifyObj(`c`) includes two associative arrays (Hashtables). One of them (TEMP) contains the temporary classification of the current sentence, the next one (FINAL) contains the final classification of the message. The array TEMP receives values during the parsing process (`addTmp("x")`).

If the end of a sentence will be obtained (`'.'`) then the temporary classification form the array TEMP will be added to the existing FINAL classification (`addTmp()`).

If the end of the sentence will not be obtained because only the first part of the sentence match a rule of the grammar, the TEMP-classification will be deleted. In the example given above the TEMP-classification is not used because the classification found can be added directly and definitely (`add("x")`).

4.2 Lexical Analysis

The lexical analysis has to prepare the textual messages into a form suitable for processing by the parser, i.e. into a form that will be accepted by all means (the case `notClassified` is a part of the grammar). It is necessary for every sentence to be classified because the parser should not interrupt the classification.

At the very beginning, each message will be decomposed into sentences. Because of that all abbreviations are investigated for change (to delete the dot) to guarantee that sentences are identified correctly.

Then all words will be removed from each sentence that are not terminals of the given grammar, i.e. not included in `allWord`.

This process of text message preparation can be called a normalization. In the next step normalized texts are processed.

Example 5:

The message

"eBay's net profit doubled, citing better-than-expected sales across most of its segments."

will be converted into

"eBay net profit doubled better-than-expected sales."

(End of example)

Example 6:

The message

"XY company closed with a loss last year but this year will be closed with a profit."

will be converted into

"XY loss last year but this year profit ."

(End of example)

4.3 Classification

The process of classification runs on normalized text messages. The parser reads every message and builds a corresponding object ClassifyObj that contains the

number of classified (event. non-classified) sentences inclusive the corresponding arrays TEMP. For purposes of the later statistical investigations the mapping (message; ClassifyObj) will be stored.

5 DATA FOR EXPERIMENTS

We used about 27.000 messages of Wall Street Journal - Electronic Edition in time interval February 2001 to November 2006 (about 400 news stories in a month). The messages are short summaries of longer stock reports. Most (75 %) of them contain one sentence only (see examples above).

The grammar has been constructed on messages of the year 2004 and the validation of the classification has been done manually on messages of the year 2002.

For purposes of prediction we used stock prices of US stock index Dow Jones in the same time interval.

6 EXPERIMENTS AND ACHIEVED RESULTS

We developed a grammar based on messages of 2004. Because of the simplicity and because of the fact that only 25 % of messages consist of more than one sentence, our grammar can process only isolated sentences. We did not investigate relations between sentences in one message. This will be one of the topics of our future research.

In the following example we can see that there is not always a direct relation between more sentences of a message. Often more one-sentence-messages are composed into one composed message even if they have the same meaning being separated in different messages.

Example 7:

Techs edged higher Tuesday afternoon after the previous session's selloff. Cisco fell amid downbeat analyst remarks. Microsoft and Dell rose, but AMD fell.

At its first part, the grammar contains a filter template that identifies sentences that cannot be classified using our grammar. There are, for example, sentences consisting of less than two words or sentences that do not contain a company name. Another filter template identifies sentences that have—according to our opinion—no direct influence on markets. We also wrote filter templates to identify sentences that contain data we obtain from other sources, e.g. how many points Dow Jones declined yesterday. After using filter templates about 50 % messages were eliminated.

The next part of the grammar contains 46 rules that represent classes of messages. They were obtained from messages of 2004 that we classified manually and used as a training set. We validated them using messages of 2002 and achieved a precision in average 92 %. The recall has not been measured because for that it would be necessary to classify all 5000 messages of 2002 manually, which would be too time consuming. Using this part of grammar about 45 % of messages were assigned to known classes, the rest was assigned to the class NotClassified.

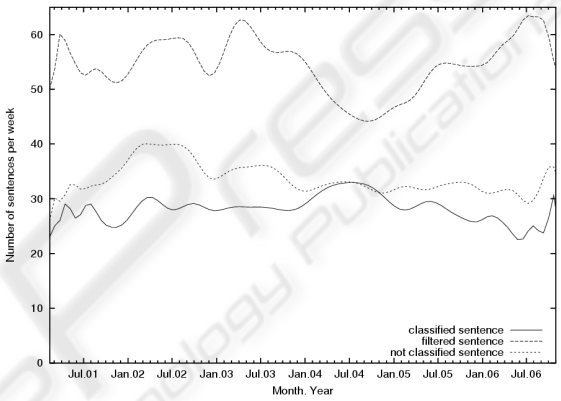

Figure 1: Number of classified, filtered and not classified sentences per week.

As mentioned above, we classified news stories into two classes: positive messages, negative messages. After the classification we used the relation between the number of positive messages and the number of negative messages as a simple indicator for prediction. Our hypotheses is that when positive messages are in a majority then the market index will go up, when the negative messages are in a majority then the market index will go down. We classified news stories in week portions and determined the relation described above. The function created in this way we compared with Dow Jones Index because the news are in English and are concerning the US-market. To eliminate the dependency from the number of all messages in weeks, which is different of course, we normalized using the following formula:

$$prediction_t = \frac{\sum_{i=1}^{n} w_i \cdot c_i}{\sum_{i=1}^{n} c_i} \qquad (1)$$

$prediction_t$ prediction at time t,
$w_i \in [-1,1]$ weighting of class i,
$c_i \geq 0$ number of classifications in class i

In the most simple case, weighting $w_i = 1$ will be assigned for all positive classified messages and

$w_i = -1$ for all negative classified messages. News stories that cannot be either positive nor negative classified will be weighted with $w_i = 0$. Using this method the majority is easy to detect. For $result > 0$, we have more positive then negative, for $result < 0$ we have more negative then positive messages. Using the weights 1,-1, 0 the computation of the formula (1) will be easy.

$$prediction_t = \frac{pos - neg}{pos + neg} \quad (2)$$

pos number of positive messages,
neg number of negative messages

In Fig. 2 we can see the Dow Jones Index and the function given by the formula (2). In this case, we have got 22 positive, 20 negative, and 4 zero classifications.

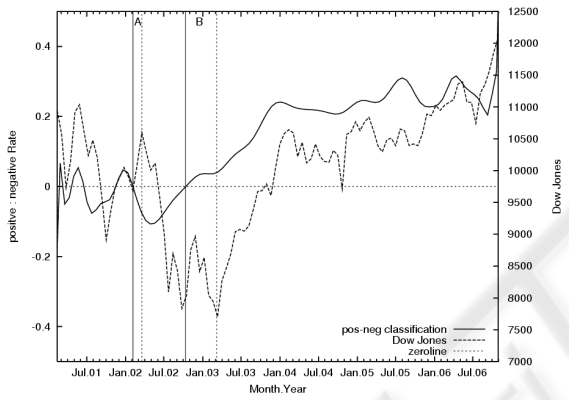

Figure 2: positive : negative classification rate and Dow Jones index.

The Dow Jones trend can be compared with the trend of our prediction function. We can investigate whether it would be possible to predict the trend of Dow Jones using the classification described above and how much importance such prediction would have.

To get a practical effect we need a prediction indicator that reacts in advance, before the market index changed its trend. To test our hypothesis we defined two time intervals *A* and *B*.

In the time interval *A* the prediction curve crosses the zero line top down at the February, 5, 2002 which predicts a long-term down trend even if Dow Jones was going up. The Dow Jones Index achieved its top at the March, 20, 2002 and then it changed to the down trend for long time. In this case the prediction came 2 months ahead.

In the time interval *B* we can see that the prediction curve crosses the zero line at the October, 12, 2002 from the bottom up. The Dow Jones Index achieved its lowest point at the March, 8, 2003 and then it changed to the up trend for long time. In this case the prediction came 4 months ahead.

To get a homogenous prediction curve we used smoothing so that the prediction is not as precise in the time interval *A*. In time interval *B* we can see the crossing having a great distance from the trend change of the market. So, we evaluate this method as interesting even if we have not more time points where the trend changes are accompanied by enough news stories in electronic form.

7 CONCLUSION

In further work we will try to collect and classify greater collections of news stories. However, to get a large collection of news stories is difficult because the electronic versions of news were not commonly used in the past.

In addition we evaluate the impact of messages in context of other economic parameters. We are just experimenting in using a neuronal network to process the classified news stories. In such a way we can include other economic influences like oil price, gold price, relation between currencies, etc. We hope to get some refinement of the prediction.

The next problem is that we also need to take into account that some of news are not true and some of them has been constructed with the intention to mystify the investors.

Our grammar already contains a class called GuessMessage that discover templates with verbs like ”to expect” or ”to plan”, but this modality is not sufficient. We will try to improve templates identifying such messages.

REFERENCES

Franke, S. (2000). Bex a bnf to java source compiler. http://www.bebbosoft.de/.

Kroha, P. and Baeza-Yates, R. (2005). A case study: News classification based on term frequency. In *Proceedings of 16th International Conference DEXA'2005, Workshop on Theory and Applications of Knowledge Management TAKMA'2005, pp. 428-432*. IEEE Computer Society.

Kroha, P., Baeza-Yates, R., and Krellner, B. (2006). Text mining of business news for forecasting. In *Proceedings of 17th International Conference DEXA'2006, Workshop on Theory and Applications of Knowledge Management TAKMA'2006, pp. 171-175*. IEEE Computer Society.

Lavrenko, S., Lawrie, O., and Jensen, A. (2000). Language models for financial news recommendation. In *Proceedings of the Ninth International Conference on Information and Knowledge Management, pp. 389-396*.

Macsakssy, H. and Provost, S. (2001). Information triage using prospective criteria. In *Proceedings of User Modeling Workshop: Machine Learning, Information Retrieval and User Learning*.

Nahm, M. (2002). Text mining with information extraction. In *AAAI 2002, Spring Symposium on Mining Answers from Texts and Knowledge Bases, Stanford*.