Extraction of Goals from Premier League Match Reports using Natural Language Processing Techniques

Student Project

presented by
Jochen Hülß & Matthias Rabus
Matriculation Number 1376749 & 1207834

submitted to the
Chair of Information Systems V
Prof. Dr. Christian Bizer
University Mannheim

Mai 2013

Contents

Chapter 1

Project Summary

1.1 Application Domain and Goals

In recent years our society underwent a remarkable change in terms of the ubiquituous availability of electronically-written text. The reader might think of the rise of e-ink interfaces and the subsequent shift from print media to electronical books, papers, magazines and so forth. Apart from editorial content, a massive development of informal information sources has been taken place.
On the one hand, the blogosphere puts the right of speech on an entirely new level. On the other hand, numerous possibilitites have been emerging to customers reviewing, recommending and, thus, influencing the perception of products and services in a written way. Automatically retrieving structured information from these various sources of written text is yet a challenging task (Cellier et al., 2010, p.1). However, robust models in this domain can even help gaining a more transparent overview in emergency situations such as disaster relief by analyzing Twitter feeds.

Due these plentiful resasons, the domains of text mining and Information Extraction (IE) have drawn their attention to us. Both domains tied together enable the automatic identification of selected types of entities, relations, or events in free text (Grishman, 2005, p.545). Our research interest and also project goal is twofold. Firstly, we want to find and apply patterns extracting a particular event from an unknown textual source. Secondly, the robustness of the patterns found should be assessed by classifying an unlabeled source. According to Weiss et al. 2005 the first task can be referred to as Information Retrievel because we provide "clues" that determine the matching documents. They also argue that the second task is considered as a Document Classification one.

More specificically, we want to examplify written football match reports from England's Premiere League to investigate whether the event of a goal can be mined for automatically. With the help of manually-found seeds, our hypothesis is that a model classifying a document according to the occurence of a goal with an accuracy greater than 75% is achievable based on automatically-labeled training data.

The research report is comprised of a description of the obtained data set [1.2], our various preprocessing steps [1.3], the applied web data mining methods [1.4] as well as an evaluation of our results [1.5].

1.2 Structure and size of the data set

Since many NLP tools work best with English language, we would like to increase our chance of success by using football reports of the English Premier League. The BBC sports department offers match reports for every game of the currently ongoing season 2012/2013. The amount of reports is sufficient for the purpose of this project. A Premier League season consists of 38 matchdays with 10 games each. BBC provides one report per game. The BBC sports department is a good datasource because the reports are neutrally written and have no preference for any team. Furthermore, the used language has a higher level and should be, compared to spoken word, easier to analyse.

The first step is gaterhing the data from the BBC homepage. Crawling the data will be described more detailed in the following section about preprocessing. These data was the input for the following classification process. Thus we wanted to classify each sentence wheter it contained a goal or not, we had to generate a sentence-wise input first. The result of our crawling process was one HTML file per report. Like every HTML file, our results had a nested tree structure including some Javascript and CSS, as well. A problem could be that the text itself contained HTML tags, for example references to other reports. As mentioned before we had to extract the plain text from the rest of the document. Then we had to divide each text into the sentences itself. This could be a hard as well, because a puncuation mark could be a not sufficient seperator, for example if we consider the name of the stadium "St. James Park". The extracted sentences will be a sufficient input for the further research.

1.3 Preprocessing

As said in the previous section, the project started with gathering the data. We used RapidMiner to crawl the data for us. RapidMiner is an open source data mining tool developed by rapid-i.com. RapidMiner allows the user to define a so called process, which is a combination of certain operators. The operators are connected by defined in- and outputs and can be parameterized to perform specific actions.

RapidMiner provides an operator to crawl the world wide web. BBC's Premier League result page is structured as follows: It has a main site from which every game report is liked. The reports are grouped by the day of the match. The first task of the crawler is to find the links to the game reports within the results page and then download each linked page into a seperate HTML file. Then the text has to be extracted from the HTML files. Therefor we use RapidMiner as well. The RapidMiner process takes all HTML files and extracts the plain text. This is achieved with the Cut Document-operator and an XPath-query. The extracted text is then written to one file for all reports because the Write as Text-operator can not write into multiple files.

Afterwards we had to do parallely two steps. First we had to seperate the text file that contained the whole text from all reports and second we had to manually generate patterns that described goals.

For both tasks some helpful methods were implemented in Java. To use the text for our classifier, it had to be splitted into sentences and each sentence had to be written into an seperate file. Accomplishing this task we used a build-in Java class: BreakIterator. The class provides a method that can cut a given text into sentences given a specific language, English in this case. The previous preprocessing step created some additional lines of text, for example for every new processed file, that also have been removed within this step. As mentioned before, the results of the text splitting were saved and written into seperate text files for further use, but also processed to find some sentences containing goals and some counter examples.

If we want to train a classifier to distinguish the sentences for us we first have to create examples. Thus there are seldom two similar sentences describing a goal, you can find certain patterns if you analyze sentences using Natural Language Processing (NLP) tools. "Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications." (Liddy, 2001, p.1) To construct sufficient patterns we needed two elements of Natural Language Processing: Named Entity recognition (NER) and Part-of-Speech tagging (POS). POS assigns each word in its part of speech, it determines if it is a verb, noun, adjective etc. (Voutilainen, 2005, p.219) Named Entity recognition determines if a

word is an entity, for example a name, a place or a city. The Stanford NLP Group provides a useful Java library with many build-in language processing tools, including NER and POS tagging. Using this library, we were able to print out a sentence including named entities and part-of-speech tags. We did this for three match reports to manually detect patterns.

Jochen: Gate, Jape-Rules, ngrams

1.4 Actual Web Mining

After generating the input for our model we had to find a good model for our preprocessed data.

Matze. Verbindung zur Vorlesung. Web mining techniken (classfication: svm, naive bayes, knn). Rapidminer prozess. apply model

1.5 Evaluation

Jochen. Resultate von Precision und Recall der Classification. Diskussion. Positive und negative Einfluesse auf ergebnisse. Analog fuer ungelabelte Daten

1.6 Conclusion

Wrap-up.

Bibliography

Cellier, P., Charnois, T., Plantevit, M., and Crmilleux, B. (2010). Recursive sequence mining to discover named entity relations. In Cohen, P., Adams, N., and Berthold, M., editors, *Advances in Intelligent Data Analysis IX*, volume 6065 of *Lecture Notes in Computer Science*, pages 30–41. Springer Berlin Heidelberg.

Grishman, R. (2005). Information extraction. In *The Oxford Handbook of Computational Linguistics*, pages 545–559. Oxford University Press.

Liddy, E. D. (2001). *Natural Language Processing*. Encyclopedia of Library and Information Science. 2 edition.

Voutilainen, A. (2005). Part-of-speech tagging. In *The Oxford Handbook of Computational Linguistics*, pages 219–232. Oxford University Press.

Weiss, S., Indurkhya, N., and Zhang, T. (2005). *Text Mining: Predictive Methods for Analyzing Unstructured Information*. Springer-Verlag New York.