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

April 2013

Contents

Chapter 1

Project Outline

1.1 What is the problem you are solving?

The project originates from a classical text mining problem. Generally speaking, we would like to deal with the question whether or not Natural Language Processing (NLP) tools are able to extract and correctly classify certain events in a given text. More specifically, our intention is machine-reading written reports on Premier League football matches. The particular event we are looking into is the scoring of a goal. At first sight, this appears rather straightforward, but by taking into account that there are numerous or even infinite ways to express that a team marks a goal, the difficulty to find as many of them as possible is challenging. Thus, a classification of text snippets based on manually found patterns indicating a goal is needed. Furthermore, we would like to enhance our search result by using learning algorithms.

While looking for goals in football match reports, we are omitting the detection of team that scored as well as the unique recognition of a goal as many reports refer multiple times to the same goal.

1.2 What data will you use? Where will you get it? How will you gather it?

Since many NLP tools work best with English language, we would like to increase our chance of success by using football reports from Englands 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 because after 32 matchdays exactly 32 * 10 = 320 reports are available. The number of reports is even growing until the end of the season. Our first project step is gathering the data

from the BBC homepage. Therefore we have to develop a crawler which crawls the results pages of the premier league and accesses every game report. A sample data set can be found on http://www.bbc.co.uk/sport/0/football/21992293.

1.3 How will you solve the problem?

1.3.1 Crawling

As mentioned above we need a crawler to gather data. Web crawlers, also known as spiders or robots, are programs that automatically download Web pages.[...] A crawler can visit many sites to collect information that can be analyzed and mined in a central location, either online (as it is downloaded) or offline (after it is stored). (Liu, 2007, p.311) In our case the crawler should start at the overview page of all games played in the current season. The element tree of the game pages are identically constructed so it should be possible for a crawler to find the element named article within the tree and extract its content while removing additional HTML tags. Each report should be saved in a separate file.

1.3.2 Seeds and patterns

At first we have to figure out a certain amount of sentences which describe a scored goal. These sentences are so called seeds. From this seeds we can infer certain patterns, e.g. from the sentence Sigurdsson scored a late equaliser for Tottenham we can infer the pattern $NamedEntityscored.*$NOUN.

There are several methods applicable to construct such patterns. Named Entity Recognition (NER) labels asequences of words in a text which are the names of things, in our application football clubs, players and coaches. The Stanford Natural Language Processing Group offers a Java software library including a Named Entity Recognizer. The Stanford Core NLP offers another useful tool for pattern detection: the POS-Tagger. POS stands for part of speech, so a POS-Tagger categorizes each word as a part of speech depending on the word and the context. Both tools will be helpful to detect patterns. After extracting patterns from some samples, we apply them to the data we have. Therefore you could, for example, use the tool RapidMiner. The created set of sentences will then be used to learn new patterns. The repetition of pattern learning and its application to data is called learning.

1.3.3 Training and Machine-Learning

To improve the effect of machine learning we have to create a second set of sentences which dont describe goals but have a similar structure. Both sets, the set including goals and the one without goals will build up our training set. The training set will be used to train a classifier, e.g. a Naive Bayes classifier. Then the data will be classified and the results will be evaluated.

1.4 How will you evaluate, measure success?

We will measure success two-dimensionally. The first dimension is the measurement of two metrics, namely precision and recall. Where precision p is the number of correctly classified positive examples divided by the total number of examples that are classified as positive. Recall r is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set. (Liu, 2007, p.103). These figures are taken for iterations all along the way of improving our search results. At least we would like to record precision and recall before and after the application of the learning algorithm. Ideally, we will be able to obtain intermediate results as well. Subsequently, the second dimension is determining a trend of the direction whereto the development of precision and results leads while the learning algorithm is applied.