# IBDB

**Bas van der Borden** s.c.van.der.borden@student.vu.nl

**Barry Oosterhuis** b.a.oosterhuis@student.vu.nl

**Romano Vacca** r.d.vacca@student.vu.nl

## Abstract

The purpose of this project is creating an easily usable application that provides users with the most positive and most negative reviews from books by using text mining. Text mining is a promising new technology that extracts structured data from unstructured text so that computers can help us to deal with vast amounts of rapidly changing knowledge and information, (Indurkhya and Damerau 2010). This paper describes the design and implementation of sentiment analysis on book reviews from bol.com. Preprocessing the data, exporting the results to JSON and developing a website with HTML/CSS were some of the tools necessary.

## 1. Introduction

In current society, people expect to be able to have the information they are seeking to be provided immediately. According to the CEO of Chartbeat, Tony Haile(2014) , people spend less than 15 seconds actively on a webpage. This means that if one does not succeed to satisfy the user with the information they are seeking within this short time-frame, they will probably leave the website. This could mean the website does not contain the information the visitor is looking for or the website does not show the information fast enough due to long loading times. This is were IBDB steps in. In an age where technology is very important, paper books are used less. By making an application that enables anyone to quickly find reviews about books, it is hoped to contribute in making people read more books. The concept of a website with reviews about books or even movies is not new, however, the way the reviews are rated in IBDB is. In this paper, the application and its scoring function will be explained, as well as the implementation of the semantic analysis and the design choices that were made.

### 1.1. Research Question

Will scoring the reviews based on the amount of positive and negative words give acceptable results when displaying the most positive review as well as the most negative review? The expectation prior to this project is that this method will indeed yield an appropriate result, but it could probably use a lot of improvements when used on a large database.

## 2. Methods

In this section, data usage and used methods are explained. These involve the bol.com book review database used for the IBDB application and software such as Python[1] and JSON[2].

### 2.1. Data Usage

The dataset used for this project, was the bol.com book reviews. The dataset consisted of a CSV file with about 800.000 LOC and around 5000 unique books. Also a dutch lexicon of 10000 words was used for semantic analysis. How these datasets were adjusted to work for this project will now be further discussed.

[1] https://www.python.org

[2] http://www.json.org

### 2.2. Preprocessing the Data

The first step of preprocessing the data was giving the rows a name. The rows of the reviews were divided in; "UID", "Book Title", "Grade", "Summary Title", "Summary".

UID being the unique ID of each review, Book titles is used to properly group reviews about the same book. The Grade is the amount of stars users gave the book on bol.com, the grade can be any number from 1-5, 1 being low and 5 high. The short summary displays the title people gave to their review. This column is not used in the semantic analysis. Last but definitely not least, the Summary column is the actual review written by users on which the semantic analysis is conducted.

Table 1: Example output of the dataset after dividing it in multiple categories

| | UID | Book Title | Grade | Short Summary | Summary |
|---|---|---|---|---|---|
| **1** | 25372538 | Echte mannen eten geen kaas | 2 | Beetje tegenvallen! | Ik had meer van het boek verwacht. In de media zeiden |
| **2** | 1835283 | Echte mannen eten geen kaas | 4 | Goed boek. | |
| **25** | 372651863 | Haar naam was Sarah | 5 | Geweldig boek! | Wat een top boek. Geweldig! De schrijver is een van mijn |

The dutch lexicon contained information about each individual word. Each word was followed by what type of word it was (adjective, noun, verb) and if the word can be viewed as positive or negative. This lexicon did contain some errors. Some of these included: duplication of words, words not in dutch, lack of positive or negative mark. The application already applies pos-tagging so it was not necessary to keep track of the type of words in the lexicon. To be able to use the dutch lexicon the lay-out had to be changed.

Each line containing a word had to be rewritten to properly represent an useable Yaml dictionary so it could be used to test for positive and negative words, e.g. "jammer: [negatief]"

After solving the lexicon problems and running the script, it became apparent that some changes had to be made. The first change that needed to be made was the introduction of rating the reviews per book. With this introduction it became possible to calculate the most positive and negative review per book which is part of the goal. A second change made was the placement of the scoring of reviews. This change needed to be made to make sure that the script would return the most positive and negative review per book and not of all the reviews. A more in-depth explanation about the scoring algorithm can be found in the Software/Techniques used section.

### 2.3. Software/Techniques used

For this research, Python 3.0 was used. The main reason for choosing Python and in particular Python 3.0 was that this version of python works faster with the sentiment analyses tools necessary for this project. For the sentiment analyses a script made by F. Javier Alba(2012) in combination with NLTK[3] was used. NLTK is a platform where one can build python programs to work with human language data. Also provided for this research were Yaml dictionaries. In these dictionaries, words are marked as positive or negative.

To be able to answer the research question of this paper, the provided sentiment script had to be modified. This was done by adding a counter to select the most positive and most negative review. Below is some pseudo-code to describe how the sentiment analysis works:

**Algorithm 1** calculating most positive and most negative score

```
maxscore= -9999 , minscore= 9999 , currentscore = 0
FOR every review:
if curentscore > maxscore and word = positive
THEN
maxscore = currentscore
if currentscore < minscore and word = negative
THEN
minscore = currentscore
```

To be able to use the results, they were saved into a JSON database. It took Python nearly 11 hours to run the script and save all results into the JSON database. The database can be accessed with Javascript functions.

## 3. Visualisation

There were a number of ways the output of the sentiment analysis on the reviews could be presented. For example, one of these options was typing in the title of a book in the Terminal on a computer and displaying the most positive and most negative review. However, to optimise the user satisfaction, it seemed better to develop a website that can give the user a better understanding of how the application would work if actually launched online. On the website, users can search on the title of a book, see the rating of that particular book, and the most positive and most negative review belonging to that book. The benefit of displaying the results on a website was that it was possible to add extra value towards information-need of the user. This was

[3] http://www.nltk.org

possible by adding categories to the books, so users could not only search for book on titles, but also on multiple genres.

## 4. Evaluation

To see whether the application completely satisfies the research question of this paper, the web-based application has been used by multiple testers. These include people who already knew how the application would work and people who were completely new to the concept. In general, the testers agreed that the application provided them with useful results and the way the results were displayed received their approval. Most common received feedback included the need for a different text color and further optimisation of categories. All search queries the testers used on the application so far seemed to yield correct results. The use of a Yaml dictionary containing both positive and negative words did correctly classify most of the reviews as positive or negative. For all the books the application displayed correct reviews, which implicates that the method used was successful. However, to properly identify each review for a larger database the application could use further development. How this can be done will be discussed in more detail further on in this paper.

## 5. Conclusion

IBDB is now a functioning application that is ready to be used. Although, the review selection can be improved by adding more parameters other than just positive and negative words, the current IBDB already fulfils its role as a fully functioning application which displays information and opinions about popular books. This research has shown that using only words to analyse semantic values in opinions is a decent start. IBDB has shown that the word lexicon gives the application enough information to properly analyse the polarity in reviews.

## 6. Further Explorations

- At the moment, the application only shows the most positive and most negative reviews. To even better provide users with information about how good or bad the book is experienced by others, it may be valuable to show more reviews. For example, by showing the three most positive and three most negative reviews, one could better decide whether the book is worth reading or not. Considering the short time that was available for developing this project, together with the decision to focus on simplicity first, led to the choice to only incorporate 1 positive and 1 negative review for the time being.
- As can be seen in Table 1, some reviews were not fully complete. Some book titles only contained either positive or negative reviews due to a low amount of reviews for these book titles. One of the possible solutions could be excluding these books from the results. It should be further researched whether users find results with only positive or only negative results valuable enough or not.
- As explained before, the Grade for every book is determined by the average grade of all the reviews as found in the original database for one particular book. This is a very easy way to show how other people rate a book. However, from own experience, people tend to have different opinions about whether a book is worth a four or a five star rating. Including another scoring function to exclude for this could be a step in improving this project.
- As stated in the visualisation section, categories were added to the books. This was done manually for 50 books. This number was chosen because of the short time available, but still large enough to show the potential of the idea. It would be better if there were genres for every book in the dataset available. A best way to do this would be to automatically scrape the content from for example wikipedia[4] pages of every book. Another more complex way of accomplishing this would be automatically analysing the reviews, and trying to predict the genres.
- One of the most important future changes could be changing the way the books are rated. The reviews are now scored based on the most positive and negative reviews. This also means that reviews that are longer, tend to get more positive or more negative. By including incrementers and decrementers, there could be added value to the words used. This means that some words are more positive and more negative than others. This would lead to a more representative result and could better provide the users with information.

## References

Haile, T. (2014). "What You Think You Know About the Web Is Wrong." Disponível na internet por http em: http://time. com/12933/what-you-think-you-know-about-theweb-is-wrong. Acesso em **10**.

[4] https://www.wikipedia.org

Indurkhya, N. and F. J. Damerau (2010). Handbook of natural language processing, CRC Press.

Javier Alba, F. (2012). "Basic Sentiment Analysis with Python." from http://fjavieralba.com/basic-sentiment-analysis-with-python.html.

## Appendix

Table 2: Work sharing

| **Task** | **Person(s)** |
|---|---|
| Dataset adjusting | Bas |
| Coding | Bas and Barry |
| Creating yaml dictionary | Romano and Barry |
| JSON Database creation | Romano |
| Presentation/Poster | Romano |
| Website | Barry |
| Paper writing | Bas, Barry and Romano |

Code & Demo:
https://github.com/romanovacca/IBDB