WS 2016 Project 2

Anonymous

1. INTRODUCTION

There are many examples of people extracting the sentiment of sentences using machine learning. The goal of of this report is to examine how these methods translates into product specific contexts and compare the results. This is done using product reviews from brickset.com[1], using the data set found here[4] and using the python package textblob[6] and the deep learning tool nlp[5].

2. METHODOLOGY

All preprocessing has been done in Python[3]. Training and testing using textblob has also been done in python. The deep learning tool provided by Stanford nlp does most of the processing for us after the data have been preprocessed by Python. The code consist of the modules that correspond to the following sections

2.1 General pre-processing of data

The product review data provided in the google sheet[4] is missing data. The data that is not complete is filtered so only valid data, with a valid sentiment data point is used for further investigation.

This step reduces the data from 4901 data points to 3893 data points

2.2 Textblob sentiment

Before the Textblob model could be trained, the data had to go through some extra preprocessing. To speedup the training process, every product review is converted to a textblob object.

The data is then split using Numpys[2] Kfold function to prepare for 3 fold validation. The model is then trained using the three data subsets provided by the Kfold function, and then tested using the test subsets provided by the same Kfold call.

2.3 Stanford recursive deep learning for sentiment analysis

This sections has not been comleted

3. FINDINGS

Looking at the data, it can be observed, that 55.89% of the data is positive. So naively guessing that everything as positive, would yield this approximate result, assuming the data we have represents the data.

3.1 Textblob based

Textblob only manages to match the baseline approximated above. There is close to no preprocessing done on the data, so there is a high probability that this can be improved.

3.2 Stanford coreNLP

No completed model

3.3 External sentiment calculator

I started out by measuring how a normal not product related sentiment analyser worked, i used uClassify[7] combined with it's python api[?].

Using this the result is slightly better than the Textblob based one, with a 60% success. This is mostly due to the fact, that most of the result is ignored and treated as neutral. The reason for this is, that some reviews may be negative or positive, but not towards the product.

4. CONCLUSIONS

5. REFERENCES

[1] Brickset reviews. http://brickset.com/reviews/eurobricks, 2016. [Online; accessed 19-July-2008].

[2] Numerical python. http://www.numpy.org/, 2016. "[Online; accessed 20-mar-2016]".

[3] Python interpreter. https://www.python.org/, 2016. "[Online; accessed 20-mar-2016]".

[4] Review data. https://docs.google.com/spreadsheets/d/1Bfq2fPRiwwCGS6JgpyrVP4OSyPuhuU_sO77xnFKWSoQ/edit?usp=sharing, 2016. "[Online; accessed 20-mar-2016]".

[5] Stanford coreNLP, deep learning sentiment analysis. http://nlp.stanford.edu/sentiment/, 2016. "[Online; accessed 27-mar-2016]".

[6] Text analysis for python. https://textblob.readthedocs.org/en/dev/, 2016. "[Online; accessed 20-mar-2016]".

[7] uClassify. https://uclassify.com/, 2016. "[Online; accessed 20-mar-2016]".