Natural Language Processing MP2 Report

Group 53
93692 Bernardo Quinteiro: 50% effort
93700 Diogo Lopes: 50% effort

Models and experimental setup

Two main/more relevant models (models that broke baseline 2) were used throughout this project:

In the first one, we used a CountVectorizer to implement a Support Vector Machine. With this, we proceeded to go for a Bag of Words approach. The pre-processing of each word is made upon creation of the vector and is defined by us, in which we do the:

- Full removal of punctuation and “\n”;
- Removal of HTML tags;
- Lemmatization (using WordNetLemmatizer);
- Stop word removal (using NLTKs preset);
- Substitution of keywords for tags.

In this last step, our intent was to replace words with similar “impact” on the perceived sentiment of the review with matching tags, so that these would be more thoroughly associated. For example, words such as “fantastic”, “love” and “awesome” would all be replaced with POSITIVE. If these words were in uppercases, they’d be replaced with VERYPOSITIVE, with the same applying to negative words. We also added tags for what we called exponential words (for example, “I REALLY liked that product”) and words that deny what follows (for example, “I DIDN’T like it at all”).

After testing, we verified this accomplished baseline2, scoring around 43%, depending on the test set.

We attempted to expand upon the tagging of keywords, by creating specific scenarios for the denial words, so, for example, in the “I didn’t like it at all” example, it would be replaced with “I NEGATIVE it all”. However, upon implementation, it reduced the effectiveness of the solution.

Afterwards, we settled on a solution in which we changed from a CountVectorizer to a TF-IDF Vectorizer, and removed the stop word removal and substitution of keywords for tags, which improved our overall results to values that varied from 47-52%, depending on the test set.

This variation happens due to two scenarios: firstly, if we change the size of the test set (which is created from the training set), we would have fewer phrases to test from, which would make it more inaccurate; secondly, by changing the random_state variable, it would change the selected reviews for the test set, which could lead to the creation of more or less “difficult” test set.

Results

(For train_test_split(load_train_info(filename=token), test_size=0.05, random_state=30))

Confusion Matrix:

```
[[71  6  3 10 14]
 [ 8 48  9 20 19]
 [ 3  5 55 24  2]
 [ 3 16 36 34  7]
 [27 17  4  9 50]]
```

Classification report:

	Precision	**Recall**	**F1-score**	**Support**
=Excellent=	0.63	0.68	0.66	104
=Good=	0.52	0.46	0.49	104
=Poor=	0.51	0.62	0.56	89
=Unsatisfactory=	0.35	0.35	0.35	96
=VeryGood=	0.54	0.47	0.50	107
Accuracy			0.52	500
Macro avg	0.51	0.52	0.51	500
Weighted avg	0.52	0.52	0.51	500

Accuracy score: 0.516

Discussion

We believe that possibly the most significant failing in our attempt at the use of tags for positive or negative words was the broadness of words that could possibly be used to describe these sentiments, as, for example, reviews that consist of the usage of interjections such as 2 similar reviews “=Unsatisfactory= blegh” would never be recognised by the dictionary we created. Another issue that may have affected our results is how many reviews aren’t written in an articulate or correct form and, sometimes, even have words written with the wrong spelling, which we can’t detect, such as “Cant”.

Future Work

If given more time, we would have first tried to implement Named Entity Recognition in our models, as that could be an effective way of “fixing” what didn’t work in our attempt at word tagging in the first model. If we could associate these entities with positive, very positive, negative and very negative sentiments, we would have a much broader pool of tagged words than the one we attempted to create ourselves.

Additionally, we would try to add deep learning or machine learning mechanics, for example, by implementing pre-trained models or neural networks. This, combined with the possibilities of NER would, most likely, create a highly effective solution.