NLP MP2 Report: Classifying Beauty Products' Reviews

Group 15: Aylton Silva, 86390 Mariana Cintrão, 93737

1 Models and Experimental Setup

For training and testing, we used K-Folds Cross Validation, specifically 10-Folds which was an intuitive choice. Since the dataset was 10.000 items long, each test set contained 1.000 items and each model was trained and tested 10 times under this premise.

Initially, we implemented a simple TD-IDF based Multinomial Naïve Bayes model [1]. This alone already yielded adequate results (46.10% accuracy) so we used it as baseline.

We implemented 2 different main models: Support Vector Machines (**SVM**) and Multinomial Naïve Bayes (**NB**) and also a few different pre-processing approaches. We then tested which combinations would result in the best accuracy. The pre-processing methods we tested were Stop Word (**SW**) removal both the Porter (**PS**) and Lancaster (**LS**) Stemmers, the Word Net Lemmatizer (**L**) and for a while a Jaccard distance based autocorrect [2] that did not prove to be useful at all. Regarding SW removal, the set we used was the default english "stopword" list from NLTK.

2 Results

	NB				NB with SW removal			
	Baseline	PS	LS	L	SW	SW + PS	SW + LS	SW + L
=Poor=	54.45	53.02	49.15	53.75	51.45	49.16	49.16	51.45
=Unsatisfactory=	44.07	41.27	30.37	42.08	33.43	30.37	30.37	31.52
=Good=	41.06	41.90	38.90	42.77	39.79	38.91	38.91	39.29
=VeryGood=	46.67	45.53	39.13	44.89	40.37	39.12	39.12	40.66
=Excellent=	62.22	63.34	47.42	65.17	61.39	47.42	47.42	59.67
Accuracy	46.10	46.29	46.26	**46.32**	43.63	43.66	43.33	43.30

Table 1: F1-score $= 2 \times \frac{Precision \times Recall}{Precision + Recall}$ for respective classification and Accuracy of attempted approaches

	SVM Poly	SVM RBF
	F1-score	F1-score
Poor	49.57	41.38
Unsatisfactory	36.67	33.84
Good	38.44	45.51
Very Good	36.84	39.60
Excellent	51.05	55.38
Accuracy	41.84	42.70

Table 2: SVM Results with different kernel methods: Polynomial (Poly) and Radial Basis Function (RBF)

Our best model is NB using only the lemmatizer. For simplicity, we did not include in the results tables some intermediate experiments that we conducted. For example, we tested the removal of SWs with some added items, particularly punctuation. However this did not result in any improvement.

To cover all basis, we conducted experiments on pre-processing with SMV although SVM on its own obtained worse results than the NB baseline. However, as expected, this did not improve it above NB. Hence these results were omitted.

3 Discussion

The training dataset is considerably insconsistent. It would be expected that even if there were repeated reviews, they would have been classified in the same way, or at least in a similar way. However this is not necessarily the case. To give an example, the review “Good product” is seen 18 times and although its target is usually “=VeryGood=” or “=Excellent=” there is one sneaky instance where the provided target is “=Unsatisfactory=”. With such a straightforward sentence and the overwhelming majority of positive classifications, it is safe to assume that the “=Unsatisfactory=” is an outlier if not a plain mistake. However these kinds of errors obviously result either in a poorer training or in loss of accuracy points when used for a test set.

There are also instances that do not have any letters. Some have emojis such as “=(” which has the target “=Unsatisfactory=”, but others are more cryptic (e.g., “...” has the target “=Poor=”). We assume this is on of the reasons why removing punctuation did not help with accuracy. Disappointingly, our classifier does not seem to do well with these types of emojis. It classifies both “:)” and “:(” as “=Good=”. You win some, you lose some...

Lastly, there are multiple typos, for example what should be “anti-aging” becomes “ant aging” and “ant” in itself is still a word. But there is also “Reveiws” which really is not a word. To fill this gap we tried implementing an auto-correction mechanism based of off this article [2] however we obtained terrible results. We assumed this was because some correct words were being returned as different words.

One interesting thing that we noticed is that fortunately, the majority of incorrectly classified instances only had a minimal distance to the actual target class. Meaning, most ”mistakes” that are made in classification are not so far off from the truth (e.g., classifying as ”=Excellent=” an instance of ”=VeryGood=”).

4 Future Work

Since the lemmatizer did improve prediction, and it works by reducing words down to their “lemma” or meaning, it could be beneficial to premise it with better typo-free text. Thus, a starting point would be to make use of better text correction mechanisms.

The choice of SWs could also be something to look at since we ended up using the default set from NLTK. Analyzing the error cases in which the error distance in substantial (e.g., target “=Poor=” being classified as “=veryGood”) could give us a better insight into words or word classes which should be given a different weight in the decision making process.

References

[1] Natural Language Processing: Naive Bayes Classification in Python, https://medium.com/@eiki1212/natural-language-processing-naive-bayes-classification-in-python-e934365cf40c. Last accessed 21 Oct 2022

[2] Correcting Words using NLTK in Python, https://www.geeksforgeeks.org/correcting-words-using-nltk-in-python/. Last accessed 21 Oct 2022