Joanna Symonowicz
T21 - Capstone Project - NLP Applications

Task: write a brief report or summary in a PDF file.

Content

5.1. A description of the dataset used

The used dataset is: "1429_1.csv" downloaded from https://www.kaggle.com/datasets/datafiniti/consumer-reviews-of-amazon-products, and renamed as: 'amazon_product_reviews.csv'.

I had three datasets to choose from at this URL. I selected this one because it had the most product reviews, as shown in the table below.

Dataset name	Number of product reviews	Chosen?
Datafiniti_Amazon_Consumer_Reviews_of_Amazon_Products_May19.csv	28,332	No
Datafiniti_Amazon_Consumer_Reviews_of_Amazon_Products.csv	5,000	No
1429_1.csv	34,660	Yes

Table 1: Datasets available under URL provided in the task.

5.2. Details of the pre-processing steps

Pre-processing of the analysed review text consisted of:

1. Converting to lowercase using the lower(),
2. Removing any leading or trailing whitespace characters using the strip(),
3. Removing stop words with exceptions – see analysis in 5.3.
4. Removing punctuation using .is_punct,
5. Removing spaces between words using .is_space.

I am using the **'en_core_web_md'** model to compare sentence similarities, as **Task 20** demonstrated its superiority for this purpose. However, for assessing polarity scores, I am following the instructions and utilizing the **'en_core_web_sm'** model.

5.3. Evaluation of results

The output of the **predict_sentiment** function is a polarity score (between -1 = most negative and 1 = most positive, where 0 = neutral). I am not employing the subjectivity assessment included in the **.sentiment** attribute because all reviews are subjective.

5.3.1 Text pre-processing

I performed a long and detailed analysis of whether removing punctuation, spaces and stop words damages review's sentiment, see additional codes for testing:

- **Testing_text_preprocessing.py (Testing Cases 1, 2, 3)**
- **Testing_exception_stop_words.py**

First, I analysed whether the removal of stop words, punctation and spaces affects the polarity score at all, see **Testing Case 1**. **Table 2** contains some exemplary reviews. It is clear that the removal of stop words significantly affects the polarity score, while removal of punctuation and spaces plays only a minor role.

5.3.2 Stop words removal

Testing Case 2 compares processed review text (without stop words) with its original form for reviews which caused the above polarity sign change error. The conclusion is that removal of the word "not" plays an important role, see **Table 2**.

Therefore, words "not" and "no" are added in **Testing Case 3** as exceptions not to be filtered as stop words. This reduced the polarity sign change error from 564 to 356 for 34,659 reviews, see examples in **Table 2**.

Moreover, **Testing Case 3** analyses texts of reviews where the polarity sign change error remains. The conclusions are:

1. Spelling mistakes and ungrammatical writing lead to errors (e.g. *My god sister has been using these for years.What to read more!)*. Hence, implementing a grammar correction function is necessary, but it falls outside the scope of this project.
2. The disparities between polarity scores which remained are smaller (+/- 0.25), changing review from slightly positive to slightly negative, which is a big improvement in the function performance.
3. Actually, in half of cases, when stop words are removed (excluding "not" and "no"), polarity scores reflect the true sentiment more accurately, which is a further improvement and not an error, e.g.:
    a. <u>Original:</u> *My daughter loves her tablet. I had to make sure I brought a case because it is very thin. No problems and I can see everything she downloads* (Score: -0.01)
       <u>Processed:</u> *daughter loves tablet sure brought case thin no problems downloads* (Score: 0.05)
    b. <u>Original:</u> *Bought it for the kids to play with. I like it so far. Bought s few accessories to allow them to play with it* (Score: -0.05)
       <u>Processed:</u> *bought kids play like far bought s accessories allow play* (Score: 0.1)
    c. <u>Original:</u> This is a good tablet for your everyday used!!!!!! (Score: -0.03)
       <u>Processed:</u> good tablet everyday (Score: 0.25).

Review text	Polarity				
	Original review	**No stop words**	**No punctuation nor spaces**	**No stop words nor punctuation (nor spaces)**	**No punctuation nor stop words except "not", "no"**
I love ordering books and reading them with the reader.	0.5	0.5	0.5	0.5	0.5
Not easy for elderly users cease of ads that pop up.	-0.21(6)	0.4(3) ("easy elderly users cease ads pop")	-0.21(6)	0.4(3)	-0.216
Easy to figure out and used it extensively on my vacation.	0.21(6)	0.21(6)	0.21(6)	0.21(6)	0.21(6)
Great tablet for the price. It is easy to navigate.	0.61(6)	0.61(6)	0.61(6)	0.61(6)	0.61(6
My god sister has been using these for years.What to read more!	0.625	0.0	0.5	0.0	0.0
I'm sorry I didn't buy it sooner! Very easy to use. Love it.	0.146(1)	0.102(7)	0.18(7)	0.14(4)	0.14(4)
I have never owned a tablet. Wanted an I-Pad like most, but couldn't afford the sticker price. This one fit my needs perfect. I mainly just use it for web surfing and streaming video. No complaints so far! I would recommend this device.	0.438(3)	0.42291(6)	0.4(3)	0.41(6)	0.41(6)
Not a good product short battery life i can only use for 20 minutes	-0.11(6)	0.35 ("good product short battery life use 20 minutes"	-0.11(6)	0.35	-0.11(6)

Table 2: Comparison of polarity scores when pre-processing review texts by removal of stop words, punctuation and spaces.

However, **Testing Case 3** does not take into account cases where the combined punctation and stop words removal change the polarity sign. Furthermore, the code demands significant computational time.

Therefore, an additional program **Testing_exception_stop_words.py** further analyses which other stop words should not be omitted to preserve review's sentiment, see **Table 3**. Adding more negations (isn't, aren't, don't, doesn't, can't) did not change the output, because the model cannot handle negations when "n't" is used. However, adding further exceptions (such as: only, more, really, so, much, well, first, please, many, too, full, less, just, very, mostly, most, few, all, everything) reduced the number of changes of polarity signs from 552 to 228, leaving only ‘errors’ where the change of polarity sign actually better conveys the review’s sentiment (please, see the examples above).

Exceptions	Number of errors	Computing time [s]
not, no	552	1253
not, no, isn't, aren't, don't, doesn't, can't	552	1019
not, no, only, more, really, so, much, well, first, please, many, too, full, less, just, very, mostly, most, few, all, everything	228	1044

Table 3: Analysis of how different exception words affect the polarity change sign.

5.3.3 Testing similarity

I grouped reviews into three groups based on their polarity score: (negative / neutral / positive). I then run tests to compare similarity of reviews within each group. The runtime for comparing all 34,659 reviews with one another exceeded 6 hours, so I used the first 500 reviews instead (computing time: ~33 min). The average similarity score within polarity groups is:

- ~0.784 for negative reviews,
- ~0.664 for neutral reviews,
- ~0.776 for positive reviews.

These values closely resemble the average similarity score between reviews in opposing groups (positive and negative), which is ~0.776. Hence, regardless of whether sentences convey a similar message, the similarity score between them remains unchanged.

In conclusion, it seems that the **similarity()** function analyses similarity of words rather than the similarity of meaning, which is not the best. Thus, it might be better to assess **polarity** rather than **similarity** to correctly evaluate the similarity of meaning between sentences.

5.4. Insights into the model's strengths and limitations

Limitations:

1. The model might not assess correctly the polarity score for reviews with spelling mistakes.
2. The model is prone to mistakes for individual reviews, and thus should be rather used for statistical analysis.
3. The model cannot handle negations when “n't” is used, e.g. (“isn’t, don’t).

Strengths:

1. In most cases, it correctly evaluates review’s sentiment.

2. It is handy for the automatized analysis for sentiment in big datasets, giving a general idea whether consumers are happy with a product.
3. The **predict_sentiment** function can evaluate the similarity of meaning between sentences better than the **similarity()** function, and can do so much faster.