NLP Deliverable

Classification of sarcastic comments in Reddit

Judith Marrero Ferrera

Motivation

Natural Language Processing (NLP) is a technique that aims to computationally understand, interpret and manipulate human language. One of NLP's fields of work is sentiment analysis, which consists of identifying the tone of texts and seeing whether they are positive, neutral or negative. Identifying sarcasm is a particular case of sentiment analysis where instead of detecting a sentiment in the whole spectrum, the focus is on sarcasm [1]. According to the Oxford dictionary, sarcasm is "*the use of language that normally signifies the opposite in order to mock or convey contempt*" [2]. Therefore, its detection is hard because its boundaries are not well defined, and an apparently positive comment can actually have a negative meaning if it is sarcastic, and vice versa.

Furthermore, Reddit [3] is a social news aggregation, web content rating, and discussion website where members submit contents such as links, text posts, images and videos that are voted up or down by other members. Posts are organized by subject into user-created communities called "subreddits", which cover a variety of topics such as news, politics, science, movies, etc. It is a particularly good site for studying sarcasm because members often use the tag "`\s`" when their post is not meant to be taken seriously but in a sarcastic way, significantly easing the identification and construction of a corpus that can be used to build AI models that automatically detect (or try to) sarcasm in texts.

The original dataset belongs to a study carried out by Khodak et al. [4]. However, to carry out this deliverable, I have used a cleaned and structured version uploaded to Kaggle (the link is at the end of this document and on GitHub). The analysis has been made over the file train-balanced-sarcasm.csv[1], which contains over 1 million comments in total (sarcastic and non-sarcastic) and other related features such as upvotes, downvotes, subreddit, etc. Sarcastic comments are labeled with a 1 and non-sarcastic comments with a 0. The full structure and the transformations made to the dataset can be seen in the `.Rmd` file.

Exploratory Data Analysis

After loading the dataset into the workspace, removing the missing values and some unwanted columns, the length of the comments is analyzed. Since the dataset is quite large

[1] The authors provide train and test sets (with balanced and unbalanced options). However, since the test sets are codified and I could not find how the codification was made, I have used their balanced train set as a whole and divided it into train and test for the classification as part of the code.

and the operations would take a while, 2 lists of 10k comments (one list for sarcastic and other for non-sarcastic) have been sampled from the original data. Then, a histogram has been plotted with the number of characters of both types of comments as their length. Looking at *Figure 1*, we can see that sarcastic comments tend to be a little longer, although the difference is not too high compared to the length of non-sarcastic comments.

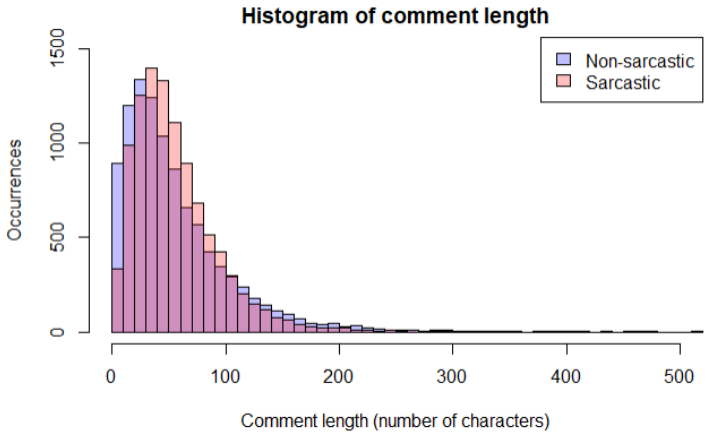

Figure 1. *Histogram of sarcastic (pink) and non-sarcastic (blue) comments length.*

Next, we have analyzed the appearances of some special characters in sarcastic and non-sarcastic posts. We could hypothesize that sarcastic comments are an "exaggerated" modality of language, and that they need to be emphasized somehow to make the sarcasm perceptible in written short texts as these Reddit comments are. To check this, the number of some special characters such as exclamation marks, question marks and asterisks has been counted and compared in both types of comments with the function `count_character()`. The results, shown in *Table 1*, show that sarcastic comments contain more than twice the number of exclamation marks than non-sarcastic posts. Therefore, they can be a clear indicator of sarcasm. On the other hand, the other analyzed characters do not show significant differences between the two groups. Other special characters can be analyzed using the same function and process.

Character	Non-sarcastic	Sarcastic
Exclamation mark (!)	565	1301
Question mark (?)	1313	1119
Asterisk (*)	555	588

Table 1. *Count of special characters in sarcastic and non-sarcastic comments.*

Afterwards, the subreddits the comments belong to have been studied. As another hypothesis, we can expect some contexts or topics to be more prone to jokes and sarcasm than others. Thus, we have ranked the top 10 most common subreddits in which the sarcastic and non-sarcastic posts of the dataset appear to see if there are any changes. As seen in *Figure 2*, there are actually some interesting differences that are worth mentioning.

	subreddit <chr>	n <int>	subreddit <chr>	n <int>
1	AskReddit	39309	AskReddit	26365
2	politics	15586	politics	23907
3	funny	9840	worldnews	16947
4	leagueoflegends	9627	leagueoflegends	11407
5	worldnews	9429	pcmasterrace	10759
6	pics	8329	news	10193
7	pcmasterrace	8228	funny	8099
8	nfl	6935	pics	7823
9	nba	6698	todayilearned	7753
10	news	6698	GlobalOffensive	7584

Figure 2. *Ranking of most common subreddits in non-sarcastic (left) and sarcastic (right) comments.*

We can see that the first 2 positions are *AskReddit* and *politics* in both cases, although the difference between them is way more significant in non-sarcastic posts and *politics* appears more with sarcasm. Then, it is interesting how *funny* goes from the 3rd to the 7th place in non-sarcastic and sarcastic, respectively, and *news* and *worldnews* rise a few places in sarcastic posts. This means that in the subreddit *funny*, the members post a kind of humor that can be described as more literal or obvious, non sarcastic, while subreddits about current affairs and news gather a lot of sarcastic posts. Indeed, sarcasm is not always used for humoristic purposes but as a defense or coping mechanism, so it is not weird for it to be very common in contexts that are related to current (and sometimes shocking) affairs.

Lastly, spaCyR is initialized, the comments are tokenized and the level of profanity is studied in both types of posts. To do it, a file of profane words is downloaded from GitHub [5] and the number of coincidences is counted using the function `count_profane()`. The list contains really obscene and disgusting expressions, so reader discretion is advised. Results show that sarcastic comments tend to have more profanity than non-sarcastic comments (with 1493 and 1377 coincidences, respectively), but the difference is actually smaller than expected.

Modeling and evaluation

Moving to modeling, some machine learning algorithms were applied to build models that are able to identify sarcastic and non-sarcastic texts. Prior to that, the data frame is transformed into a corpus object and the latter is tokenized to build a Document Feature Matrix (DFM) that can be used for the classification. Special characters, punctuation marks and stop words are removed from the comments and most frequent features (or words) in both groups are computed too. Regarding them, the most remarkable result is that the words “yeah” and “sure” are two of the most repeated in sarcastic posts, while in non-sarcastic comments they appear in a really smaller size in the word cloud. Finally, train and test sets are created.

The first algorithm that was applied was Naive Bayes classifier, creating models with both multinomial and Bernoulli distributions with the function `textmodel_nb()`. Then, SVM

algorithm was also applied for the classification of texts with `textmodel_svm()`. Since the dataset is quite large and the function returned an error when more than 20,000 documents were used, the function `svmClassifier()` samples the original DFM (with the size specified by the user) and creates new train and test sets to perform the classification. Lastly, the classification metrics and confusion matrices are extracted for both algorithms and the metrics using different sample sizes are compared for the SVM classifier. These results can be seen in *Table 2* and *Figure 3*, respectively.

Model	Accuracy	Precision	Recall
NB (multinomial)	0.665	0.669	0.653
NB (Bernoulli)	0.672	0.669	0.684
SVM (10k comments)	0.611	0.588	0.650

Table 2. *Classification performance metrics of the models created.*

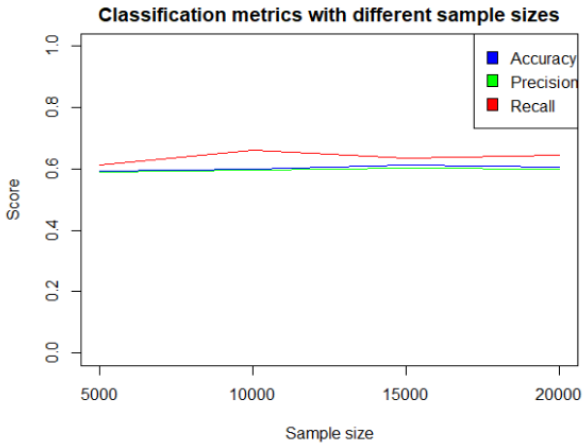

Figure 3. *SVM metrics with different sample sizes.*

Conclusions

In the different analyses that were carried out, the EDA showed some interesting patterns across sarcastic comments. For example, comments with sarcasm are a bit longer, contain more exclamation marks than usual and are more likely to appear in certain contexts or subreddits. In addition, they are more profane or disrespectful, although the small difference with non-sarcastic comments might show that the offense is sort of "hidden" with words that might seem harmless in other contexts.

On the other hand, neither of the algorithms provided particularly good results. Naive Bayes with Bernoulli distribution has the best accuracy, precision (together with multinomial) and recall, but all of them are below 70%. Furthermore, increasing the size of the sample in SVM does not improve the metrics significantly (and the execution time is much higher), at least with the studied sizes, but it would be advisable to try with bigger samples if the resources allow it. However, the classification of texts is a difficult task, especially with the particularities of sarcasm identification that were previously discussed, so these results cannot be seen as bad but as a preliminary approach to start studying and discovering this interesting field inside NLP.

Links and resources

- Code on GitHub: https://github.com/jmarrerof/NLPdeliverable
- Dataset (file train-balanced-sarcasm.csv): https://www.kaggle.com/danofer/sarcasm

References

1. Berasategi, A., 2020. *Sarcasm detection with NLP*. [online] Towards Data Science. Available at: <https://towardsdatascience.com/sarcasm-detection-with-nlp-cbff1723f69a> [Accessed 29 January 2022].

2. In: *Lexico*. n.d. Sarcasm. [online] Available at: <https://www.lexico.com/en/definition/sarcasm> [Accessed 29 January 2022].

3. Reddit.com. n.d. [online] Available at: <https://www.reddit.com/> [Accessed 29 January 2022].

4. Khodak, M., Saunshi, N. and Vodrahalli, K., 2017. A Large Self-Annotated Corpus for Sarcasm. *Cornell University*, [online] Available at: <https://arxiv.org/abs/1704.05579> [Accessed 29 January 2022].

5. Raw.githubusercontent.com. n.d. *List of dirty naughty obscene and otherwise bad words*. [online] Available at: <https://raw.githubusercontent.com/shutterstock/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words/master/en> [Accessed 29 January 2022].