Patronizing and Condescending Language Detection

Name:	**Manas Dubey**
Registration No./Roll No.:	19184
Institute/University Name:	IISER Bhopal
Program/Stream:	DSE
Problem Release date:	September 02, 2021
Date of Submission:	The date you submitted

1 Introduction

In this project we deal with the class of NLP classification problems called Patronizing and Condescending Language (PCL) Detection. Before that i would like to talk in brief, about the motivation behind these sets of problems. When one engages in PCL ,the language use shows a superior attitude towards others or depicts them in a compassionate way. This effect is not always conscious and the intention of the author is often to help the person or group they refer to (e.g. by raising awareness or funds, or moving the audience to action). However, these superior attitudes and a discourse of pity can routinize discrimination and make it less visible. [1]

Moreover, general media publications reach a large audience and the unfair treatment of vulnerable groups in such media might lead to greater exclusion and inequalities.Some of the characteristics PCL might be as such, (i)it fuels discriminatory behaviour by relying on subtle language (ii) It creates and feeds stereotypes , which drive to greater exclusion, discrimination, rumour spreading and misinformation .[2] Therefore, in the light of this challenge of identifying the usage and curb the pervasion of such language in social media platforms, NLP frameworks would be essential.

1.1 Dataset

The data set we are using has been provided on request with the original authors of the data set called "The Don't Patronize Me!" data set.The file 'dontpatronizeme_ pcl.tsv' contains 10469 paragraphs annotated with labels 0 and 1 where 0 denotes that no PCL was detected in the given paragraph and 1 denotes high levels of PCL detection. This will be used in the binary classification task.Figure 1

The file 'dontpatronizeme_ categories.tsv' is the PCL multilabel classification dataset. It contains 993 paragraphs again annotated with labels in a one hot vector array form. There are seven categories namely :-(i) Unbalanced Power Relations (ii)Shallow Solution (iii)Presupposition (iv) Authority Voice (v)Metaphor (vi) Compassion (vii)The poorer the merrier. The i'th entry in the array is 1 if that category is detected in a text span, otherwise it's 0.Each paragraph might contain one or more text spans with PCL, which may be assigned to the same or to different categories. 3.

2 Methods

2.1 Text Preprocessing

Our data was mostly clean, it had no missing paragraphs and labels. Although it contained what we believed were irrelevant prepositions, articles etc. Therefore in order to test the performance of classification we tested on both stop words removed and un-removed data.Further, we proceeded to do lemmatization and stemming also ,whose performance was also compared. We used the nltk.corpus library to removed stopwords.[3] We did the aforementioned cleaning(s) for both the binary label data and multi label data.

	text	label
0	We 're living in times of absolute insanity , ...	0
1	In Libya today , there are countless number of...	0
2	"White House press secretary Sean Spicer said ...	0
3	Council customers only signs would be displaye...	0
4	""" Just like we received migrants fleeing El ...	0
...	...	...
10464	"Sri Lankan norms and culture inhibit women fr...	0
10465	He added that the AFP will continue to bank on...	0
10466	""" She has one huge platform , and informatio...	1
10467	""" Anja Ringgren Loven I ca n't find a word t...	1
10468	""" Guinness World Record of 540lbs of 7-layer...	1

10469 rows × 2 columns

Figure 1: Binary Label Classification Data

	text	label
0	We also know that they can benefit by receivin...	[1, 0, 0, 1, 0, 0, 0]
1	Pope Francis washed and kissed the feet of Mus...	[0, 1, 0, 0, 0, 0, 0]
2	Many refugees do n't want to be resettled anyw...	[0, 0, 1, 0, 0, 0, 0]
3	"Budding chefs , like "" Fred "" , "" Winston ...	[1, 0, 0, 1, 1, 1, 0]
4	"In a 90-degree view of his constituency , one...	[1, 0, 0, 0, 0, 0, 0]
...	...	...
988	Citing the fact that these kids who died at Go...	[1, 0, 0, 0, 1, 0, 0]
989	Fern ? ndez was a well-known philanthropist wh...	[1, 0, 0, 0, 0, 0, 0]
990	Touched much by their plight , Commanding Offi...	[1, 0, 0, 0, 0, 1, 0]
991	She reiterated her ministry 's commitment to p...	[1, 0, 0, 0, 0, 0, 0]
992	Preaching the sermon , the Dean of the St. Pet...	[0, 0, 0, 1, 0, 0, 0]

993 rows × 2 columns

Figure 2: Multiple Label Classification Data

Further preprocessing such as tokenization ,ngram splitting etc. is automatically handled by the built in functions in TfidfVectorizer etc.[4]

2.2 Feature Extraction

2.2.1 Multi-Label

We proceeded with using Bag of Words for feature extraction using TF-IDF vectorizer. We also used the pre-trained skipgram word embeddings from the google news corpus.[5] Now we wanted to experiment ,as a novel contribution to the area of research in NLP, whether empirically the process of removing stop words, Stemming and Lemmatization have any considerable effects on the performance. We used the Binary Relevance classifier to tackle the multi-label classification problem.[6]The Porter Stemmer and Wordnet Lemmatizer were used for these purposes.[7] It works by decomposing the the multi-label learning task into a number of independent binary learning tasks (one class per label) and taking the union of the predictions. Inside Binary Relevance, we can feed in a collection of classifiers ,such as SVC ,MNB etc. but due to some problems with sklearn ,we weren't able to hyperparameter tune anything apart from these two classifiers.

For Multinomial Naive Bayes, the since the input should be positive always, we applied it only for the TF-IDF features. For the embedding, we calculated the average of the word embeddings for all the words per paragraph which served as the features for classification.We calculated a 200 dimensional feature vector in the embeddings. The gensim model thus used had Stochastic Gradient Descent as the optimizer , window size of 20 and ran for 100 epochs. In TF-IDF we kept n-grams = (1,3) constant to compare results.[8] The cross validation was set to 10.

Table 1: Performance for Word2Vec embedding on Binary Relevance

Stopwordsremoved	lemmatization	Stemming	Macro f1	hamming loss	Accuracy
yes	yes	no	0.15	0.25	0.18
yes	no	yes	0.14	0.24	0.18
yes	yes	no	0.14	0.24	0.17
no	no	yes	0.17	0.22	0.19
no	no	no	0.21	0.21	0.21
no	yes	no	0.16	0.23	0.18

Table 2: Performance Of TfIdf on Binary Relevance

Stopwordsremoved	lemmatization	Stemming	Macro f1	hamming loss	Accuracy
yes	yes	no	0.15	0.24	0.20
yes	no	yes	0.12	0.24	0.20
yes	yes	no	0.18	0.20	0.19
no	no	yes	0.18	0.20	0.24
no	no	no	0.18	0.24	0.20
no	yes	no	0.11	0.24	0.19

3 Evaluation Criteria

We use the macro averaged f1 score for all classes, hamming loss and accuracy for comparison of the different permutations and combinations. The zero-one loss considers the entire set of labels for a given sample incorrect if it does not entirely match the true set of labels. Hamming loss is more forgiving in that it penalizes only the individual labels.[9]

4 Analysis of Results

From experimentation ,we can see that linear SVC classifier works best with with the binary relevance methodology, its best set of hyper-parameters being gamma = 'scale', kernel = 'rbf' and C = 100 where the input features are google word2vec embeddings without removing stopwords , without any lemmatization or stemming, if we look at average across all three metrics. Whereas just focussing on hamming loss, generally, lemmatization is performing better than stemming ,at the cost of accuracy ,on the other hand accuracy is increasing with stemming.

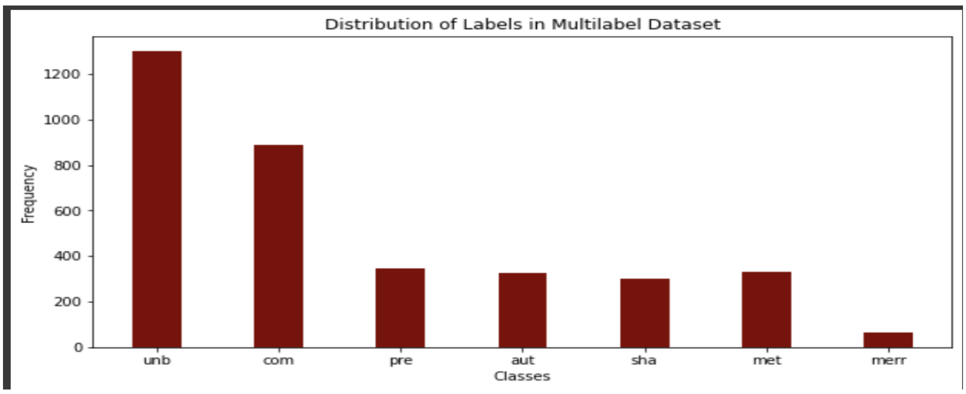

Figure 3: Multiple Label Classification Data

As to why not removing stopwords is a better option is a question that maybe related to both the structure of the data and the distribution of labels. In our case the test instances are only 993 with very subtle distinctions between relevant classes. The embeddings learned with the stopwords may be crucial in some cases for e.g. between 'Metaphor' which uses a lot of conjunctions. Similar results,although not the greatest, are observed in TF-IDF also.

5 Discussions and Conclusion

To conclude, we experimented with three different pre-processing techniques and compared the results based on certain metrics.Although, we could not conclude objectively, as to which combination of techniques are the best to use because of its dependence on a multitude of factors such as data distribution, the semantics behind the label assignment etc. but one can definitely keep an objective in mind and optimize based on our findings.

References

[1] Lilie Chouliaraki. Post-humanitarianism: Humanitarian communication beyond a politics of pity. *International journal of cultural studies*, 13(2):107–126, 2010.

[2] David Nolan and Akina Mikami. 'the things that we have to do': Ethics and instrumentality in humanitarian communication. *Global media and communication*, 9(1):53–70, 2013.

[3] Davis David. How to use texthero to prep a text-based dataset for your nlp project, Jul 2020.

[4] Cory Maklin. Tf idf, May 2019.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. 2013.

[6] Min-Ling Zhang, Yu-Kun Li, Xu-Ying Liu, and Xin Geng. Binary relevance for multi-label learning: an overview. *Frontiers of Computer Science*, 12(2):191–202, 2018.

[7] Wahiba Ben Abdessalem Karaa and Nidhal Gribâa. *Information retrieval with Porter stemmer: A new version for English*, page 243–254. Springer International Publishing, Heidelberg, 2013.

[8] Zhi Li. A beginner's guide to word embedding with gensim word2vec model, May 2019.

[9] Guoqiang Wu and Jun Zhu. Multi-label classification: do hamming loss and subset accuracy really conflict with each other? 2020.