Imperial College: 'Don't Patronize Me!' Coursework 2024

Anton Zhitomirsky

Imperial College London

`az620@ic.ac.uk`

`gitlab.doc.ic.ac.uk/az620/nlp-cw-2024`

1 Introduction

Most social media websites contain portals through which even unregistered users can view their content. Focusing primarily on textual posts, this leaves many vulnerable users to unfiltered content, which may be harmful if not regulated. Ng (2007) concludes that regardless of intention, if this content goes unfiltered it may "justify", "encode", "enact" and "routinize" discrimination amongst targeted groups.

This motivates the creation and popularization of The Don't Patronize Me! dataset to provide a source for engineers to develop categorization algorithms to progress the protection of targeted groups. The dataset, authored by Perez Almendros et al. (2020), is a labelled dataset containing more than 10,000 paragraphs extracted from news stories, country of origin, and keyword which is then labelled indicating its level of Patronizing and Condescending Language (PCL).

I propose an extension to the base-line categorization model 'RoBERTa' which beats the initial benchmarks of 0.48 F1-score on the dev-set and 0.49 on the test-set. The repository link is available at the top of this report.

Model	F1 dev-set	F1 val-set
Baseline	0.48	0.49
Mine	0.82	0.57

Table 1: F1 score of PCL language for baseline model compared to final model

2 Data Analysis

2.1 Feature Distribution

Labels arrive annotated by two main annotators. Their values span from $[0, 4]$ indicating the level of PCL within the text snippet. Perez Almendros et al. (2020) explains how both annotators individually classify each sentence with a score in the range $[0, 2]$ to indicate the strength of PCL from: no PCL, borderline PCL and blatant PCL. These scores were then summed to produce a 5-point scale of PCL, where the range $[0, 1]$ indicates a negative example, and $[2, 4]$ indicating a positive example.

Data also arrives with labeled country information indicating the source media outlet, and a keyword field. The keyword is provided as input to the model as the search term used to retrieve texts about a target community. A distribution of these fields is shown in Figure 1. Here, the supplied dev dataset contains a disproportionate number of negative (9475) and positive (993) examples of PCL.
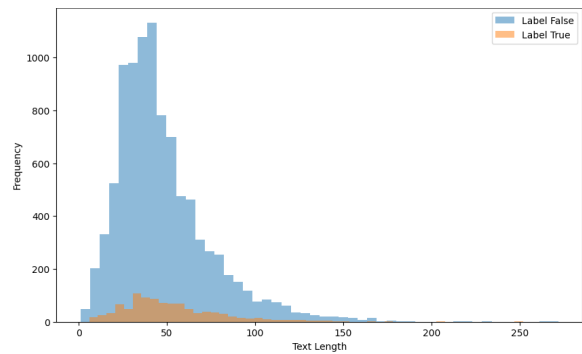

Figure 2: Histogram of word count by PCL (true) and non PCL (false)[1]

Furthermore, the distribution of word counts between the two (See Figure 2) offers no obvious underlying pattern between the two classes. This was further supported by the unequal variances t-test which returned a p-value of 1.331×10^{-7} which is not enough to reject the hypothesis that the means are the same. Therefore, there is no correlation between the length of the input and the PCL label.

2.2 Subjective Difficulty

Indeed, Perez Almendros et al. (2020) mention the subjective nature of the task as it is complicated. The word 'lovely' appears uniquely in PCL text examples. This word isn't negative, but in the context of the sentence (par_id: 2404) "We think it's

---

[1]Without loss of generality, text with more than 250 words have been clipped
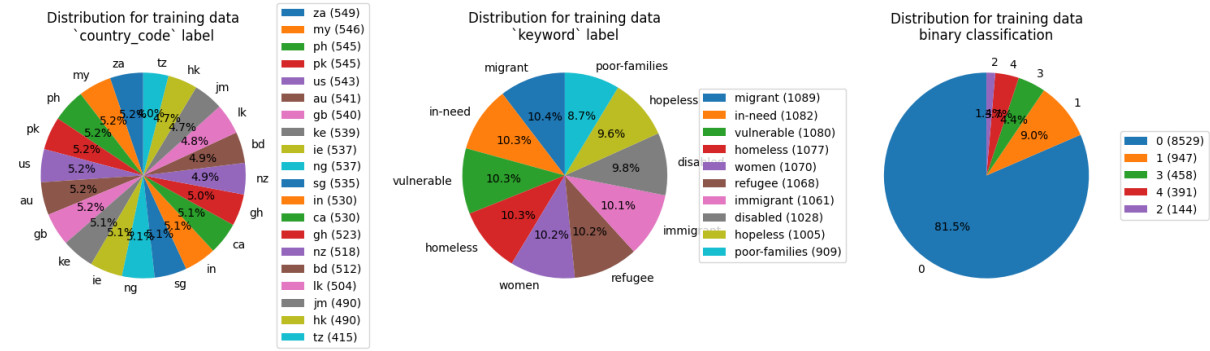
Figure 1: Pie charts of different label distributions in the supplied dev-set

lovely that so many have come forward to help out a family clearly in need!" the sentence has been labeled as blatantly PCL with a score of 3. In fact, tweaking the sentence from "a family clearly in need" to "families in need" removes entitlement and converts it to compassion. However, this is once again subjective, depending on intonation and cadence.

2.3 Artifacts

The flexible environment of news articles allows for authors to post resources and further reading, like in the case of par_id: 2838: *"Read more about the site's history here: [LINK]"*. Therefore, with a primitive language processing model, we cannot determine the degree of PCL from a link. Some links and references are also clipped in the source with token "TOOLONG" which also aught to be pre-processed to avoid confusion from the model. Additionally, writers are free to reference other users (par_id: 5598 "@toekunbore [...]"), the handle of which would likely be ignored by language models.

Lastly, adding to the complexity of processing language, we must also filter and process out-of-vocabulary words. This may appear in the form of a typo like in par_id: 8221 *"Do they think it is ""jsutice "" ti imprison [...]"*, hyperboles like par_id: 5333 *"But nooooo [...]"*, or slang like in par_id: 1674 *"Kyle really your a pig, lol [...]"*.

However, we cannot place too much focus on filtering typos, as this is a common technique employed by social media 'trolls' to avoid automatic PCL detection; this is a cat-and-mouse game which even the most sophisticated language models employed by the largest companies cannot win. We will continue with the assumption that the data is mostly clean.

3 Preprocessing: Minority Upsampling

The data is very imbalanced; Section 2.1 discusses the sparsity of positive PCL examples in the training data. Therefore, it is possible that the model may experience great performance in the majority negative class, and fail to learn details about the minority positive class (Fernández et al. (2018)).

We can synthetically produce more data points for the minority positive class by changing the content of the sample without changing the sentiment of the text.

3.1 Failed Considerations

Even the most popular augmentation techniques are task specific. These were considered, but ultimately not implemented.

3.1.1 Stop Word Removal

Stop words contain little value, so they can be useful to decrease the length of sentence. However, it may change the sentiment of a sentnece: "I do not like you because I'm better than you" changed into "I like I'm better" – removing all PCL sentiment from the sentence – and rendering an ineffective augmentation.

3.1.2 Easy Data Augmentation

Wei and Zou (2019) propose a pipeline to tweak sentences by replacing, deleting, swapping and inserting words. The authors comment that by fine-tuning replacement parameters and the up-sampling factor, it can "can boost performance on text classification tasks". However, we found that the model quickly overfit within one epoch.

3.1.3 Rephrasing Models

Further effots were dedicated to rephrasing the text with the help of a pre-trained GPT-2 model as suggested by Dai et al. (2023). However, significant

inconsistencies were discovered depending on the sentence length, so ultimately this effort was abandoned.

3.2 Successful Considerations

3.2.1 Translation

This upsampling technique is inspired by Nallamuthu (2023) who suggests translating the text into a non-English language and then back into English. The hope is that translating back introduces new wording and structure without augmenting the sentiment of the sentence. We take this a step further by introducing a parameter to control the depth of translation.

3.2.2 Swapping Sentences

Within the same paragraph, we can swap the ordering of sentences. This augmentation was less researched but helped add variance to the upsampling the variance of the minority class.

4 Data Augmentation

4.1 Tokenization

By tokenizing the input, we can efficiently assign `[UNKNOWN]` tokens to out-of-vocabulary words. This is important for a pre-trained model which is trained on well-formed English text, and may not understand slang or typos.

4.2 Masking inputs

Further benefit of tokenization includes the choice of randomly masking a subset of words in the input. This allows models to capture rich contextual information and dependencies in language. This means that our model generalizes well to various text classification tasks, even when faced with different writing styles, vocabulary, and sentence structures, which is vital for detecting unseen ways to patronize.

5 Modelling

5.1 Final model predictions

5.2 Hyper-parameters

For the final model architecture, we use a linear layer on top of a RoBERTa base model. We experimented with various other architecture, such as BERT, RoBERT, DistilBERT, along with the number of output neurons. In the end, our model has 2 output neurons, corresponding to non-PCL and PCL – the higher neuron is the prediction. We trained the model for 5 epochs with no early stopping and an initial learning rate of 2×10^{-5}. Even through the different BERT flavours have similar performance – which we hypothesize is due to the data augmentation step – we chose the best performance on a validation set.

The most significant model improvement was the introduction of a weighted loss criterion, which scales loss depending on the rarity of the class in the dataset. In this case, the loss is scaled significantly for PCL examples, of which there are few. This means that the model is incentivized to learn information from the minority class.

6 Results

6.1 Comparison

We implemented 2 simple models based on a Bag of Words approach. After segmenting all input sentences by spaces and finding unique 'words' (some of which are strings of punctuation), we encode the number of occurences of each word in the vocabulary per document in the corpus. On this matrix, we then fit a Gaussian Naive Bayes model and a Logistic Regression model. The models got positive class F1 scores of 0.14 and 0.28 on test data, respectively.

The highest weighted words in the Linear Regression model, that is, the words whose appearance most strongly correlates with PCL are: `'burlesque'`, `'duty'`, `'Dreamers'`, `'hearts'`, `'Christmas'`. An example sentence in which one of these words appears is: *"Dreamers are immigrants who were brought into the United States illegally as children . Under the program President Obama created , " Dreamers " have been allowed to stay legally"*. It is very likely that the model overfits, and because of the model, it is unable to learn the true context of the word.

7 Analysis

Effect of level of PCL on model performance

Effect of input sequence length on model performance

Effect of PCL category on model performance

8 Conclusion

In conclusion, we have successfully implemented a model which outperforms the baseline model. We have also shown that the model is better at

Model	F1 dev-set	F1 val-set	s	s
Model	F1 dev-set	F1 val-set	s	s

Table 2: F1 score of PCL language for baseline model compared to final model

predicting examples with a higher level of patronising content, and that the length of the input sequence impacts the model performance. We have also shown that the model performance depends on the data categories.

References

Haixing Dai, Zheng Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, W. Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2023. Auggpt: Leveraging chatgpt for text data augmentation.

Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C. Prati, Bartosz Krawczyk, and Francisco Herrera. 2018. *Learning from Imbalanced Data Sets*. Springer Cham.

Nandhini Nallamuthu. 2023. Handling imbalanced data – machine learning, computer vision and nlp.

Sik Hung Ng. 2007. Language-based discrimination: Blatant and subtle forms. *Journal of Language and Social Psychology*, 26(2):106–122.

Carla Perez Almendros, Luis Espinosa Anke, and Steven Schockaert. 2020. Don't patronize me! an annotated dataset with patronizing and condescending language towards vulnerable communities. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5891–5902, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389, Hong Kong, China. Association for Computational Linguistics.