Paper 2: Supervised System Evaluation

Gareth Harcombe
University of Canterbury
33186156
grh102@uclive.ac.nz

Abstract

This document contains a system evaluation for the shared task SemEval-2019. It outlines the basic task and training data available. The system evaluation compares two separate approaches to the task using Bayesian and BERT based classifiers, covering their methodology such as the tokenisation and models used, and how the results of each system compare. Furthermore, it analyses whether the approaches used generalise to Pacific languages, and identifies contexts and populations for which the performance may vary.

1 Introduction

Offensive language in social media is becoming increasingly prevalent, and it is desirable for the companies managing these sites to remove offensive language so as not to offend other users on the platform. Manual identification of offensive language is time consuming, and so it is desirable to automate the process. Zampieri et al. (2019b) SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval) is a shared task that seeks to identify offensive language within a short piece of text from social media such as a tweet. The shared task has 3 sub-tasks: identifying offensive language; breaking down what category of offensive language is used; and who the offensive language is directed at. This system evaluation will only consider sub-task A, offensive language identification, in order to simplify model comparison. Sub-task A describes a one-label problem: to label each document as either offensive or not offensive. This system evaluation will compare a Bayesian Classifier and a fine-tuned BERT model against each other and several common baseline performances in sub-task A, and consider the challenges adapting these systems to non-English languages and populations where the performance may vary.

2 Data

This shared task uses English tweets from the Offensive Language Identification Dataset (OLID) dataset, presented by Zampieri et al. (2019a). The OLID dataset includes 14,100 annotated tweets, with 13,240 tweets used for training and 860 tweets used for testing. Two examples of tweets with labels from the dataset are found in Table 1.

2.1 Annotation

OLID's annotation scheme is a hierarchical scheme with three levels to include the target of the offensive language and the type of offensive language (Zampieri et al., 2019a). The first level is to classify offensive (OFF) and non-offensive (NOT) tweets. Offensive tweets can include profanity, and any kind of insult or threat, regardless of whether it is direct or veiled.

If the tweet is labelled as offensive, then the second level of annotation is whether the offensive language is targeted or not. This is a binary classification of whether the post includes targeted insults or threats to a group or individual, or whether the post contains profanity.

If the tweet contains targeted insults or threats, then the third level of annotation focuses on the target of the offensive language. There are three labels to this sub-task: individual, such as a famous person, named or unamed individual; group, such as ethnicity, gender or religion; and other, such as an organisation, event or issue.

This evaluation focuses on labels from the first level, which annotates each document as either offensive (OFF) or not offensive (NOT).

Tweets were annotated using the crowd sourcing platform Figure Eight[1]. In order to used reliable and accurate annotators, only annotators who were experienced and passed a threshold of test queries

[1] https://figure-eight.com

Tweet	Sub-task A Label
@USER He is so generous with his offers.	NOT
IM FREEEEE!!!! WORST EXPERIENCE OF MY FUCKING LIFE	OFF

Table 1: Two tweets from the OLID dataset with labels for sub-task A

were used. All tweets were annotated by two annotators, with a third annotation and a majority vote being used in the event of a disagreement.

2.2 Class Imbalance

There is a significant class imbalance in the OLID data set, as the number of non offensive documents outnumbers the number of non offensive documents. 66.7% and 72.1% of the training and testing data respectively is labelled as not offensive. This is somewhat representative of all tweets, as the majority of tweets are not going to be offensive. However, this does also mean that a system can achieve reasonable level of accuracy by simply choosing the most common label, not offensive. In doing so, the system will not detect any offensive language, avoiding the overall purpose of the shared task. To avoid this issue, the standard metric to report for this task is the macro-averaged F1 score to measure precision and recall as well as accuracy. This presents a more accurate representation of how well the system performs at detecting offensive language.

3 Systems

Two text classifiers were chosen to compare results on this shared task. All code for the systems and the data used for the task is found in Appendix A.

3.1 Bayesian Classifier

A Bayesian Classifier is a classical machine learning technique that uses a prior and a likelihood function to approximate the probability of classifying the document in each of the classes. The predicted class is the class with the highest probability. In this implementation, outlined by Dunn (2023), the prior function $P(C)$ is the probability of the class occurring. This is multiplied by $P(W|C)$, the probability of each word in the document occurring, given the document is of class C. In order to prevent issues with small probabilities, log probabilities are used for $P(W|C)$ and $P(C)$.

Defining words for the Bayes Classifier requires text preprocessing to break down each document. This is an important step to ensure that similar words have the same probability. Although there are many ways to complete text tokenisation, this implementation first converts all text to lowercase, before removing punctuation and emojis. The text is then broken into words using NLTK's `word_tokenise`[2].

3.2 BERT

BERT is a transformer-based model. The transformer architecture was developed by Vaswani et al. (2017) and uses only attention heads, removing Recurrent Neural Network and Convolutional Neural Network aspects of model architectures proceeding it. Devlin et al. (2019) use the transformer architecture to build Bidirectional Encoder Representations from Transformers (BERT). The model is pre-trained using a masked language model objective, randomly masking tokens in the input and predicting the original token. BERT then outputs embedding vectors for all of the tokens in the input sequence.

BERT has specialised tokeniser for tasks. There are two special tokens that it adds to the input: the *[CLS]* or classification token, which is the start of every sequence; and the *[SEP]* token to determine which tokens belong to each sentence/context. In this problem, the *[CLS]* token will always be at the start of the sequence, and the *[SEP]* token will always be at the end of the sequence.

To turn this model into a classification model as described by Sun et al. (2020), all of the token embedding vectors are discarded except for the *[CLS]* token. The embedding is passed into a linear layer with a ReLU activation function to output class probabilities.

Cross entropy loss is used as the loss function, with the Adam optimiser. The model was fine tuned for four epochs, with more training cycles resulting in overfitting and decreasing test error.

4 Results

The results of the systems for sub-task A, along with models and baseline tests from Zampieri et al.

[2] `https://www.nltk.org/api/nltk.tokenize.html`

(2019a) and the top performing team in Zampieri et al. (2019b) are shown in Table 2.

The top performing team, NULI, was developed by Liu et al. (2019). This team used BERT-base-uncased with the max sentence length set to 64 and trained for 2 epochs to achieve the F1-score of 82.9%. NULI was 1.4 points better than the next best team in the shared task reported by Zampieri et al. (2019b).

The BERT system achieved an F1-score of 82.3%, and would have placed second, only slightly behind the NULI team. The Bayesian Classifier system achieved an F1-score of 72.9%, achieving better performance than the baseline tests and SVM model, but worse than the deep learning methods such as CNN and BERT.

Model	F1
NULI	0.829
BERT	0.823
CNN	0.800
Bayesian Classifier	0.729
SVM	0.690
All NOT	0.420
All OFF	0.220

Table 2: System Results

5 Discussion

When evaluating these systems, it is important to consider how the systems differ from each other, what challenges might occur when adapting them to other languages such as Pacific languages, and contexts where the performance of the systems may vary.

5.1 System Comparison

The NULI team chose some non-default hyper-parameters when fine-tuning BERT that differ from the system implementation described in this paper. Setting the max sentence length to 64 likely eliminated the noise of having a sequence of 450 zeros at the end of each string of tokens, allowing the model to better learn the differences between classes. Because of the 280 character limit set by Twitter (Twitter), most tweets would not be truncated by the max sentence length of 64, so this decision would not adversely affect the model's performance. Training for only 2 epochs likely prevented overfitting, which is very easy with a model as large as BERT and a test set with only 860 tweets. These factors combined likely lead to the NULI team's implementation performing better than the system implementation described in this paper.

The Bayesian Classifier system performed better than the baselines, which is to be expected as the prior function is simply the All NOT baseline. Therefore, we would expect that the additional information of the likelihood function to result in increased performance.

The Bayesian Classifier performed worse than deep learning models. Neural networks have the capability to capture inherent complexities in tweets beyond simple word presence such as semantic analysis, and therefore would be expected to perform better.

5.2 Pacific Languages

At a basic level, Pacific languages have different words, grammar and structure, and have less data available. This presents issues for the BERT model, as it has been pre-trained on English language texts including over 3 billion words before being fine tuned for this task. In order to be trained on Pacific language, a similarly large corpus of text would be to be obtained, and would take a large amount of resources to train. The number of native speakers of Pacific languages is significantly less than those of English speaking, making it significantly harder - if not impossible - to collate a data set in a Pacific language that is as large.

A similar challenge is presented with the vocabulary. BERT uses WordPiece embeddings (Wu et al., 2016) that are from the English language. In order to adapt BERT for Pacific languages, a different word embedding would need to be used. Creating this word embedding would require a large data set which may not be easily available.

Adapting Bayesian and BERT classifiers for Pacific languages may also present challenges due to the range of dialects that can exist within a language. This is less of an issue as the above challenges, as the English language already has dialects such as British and American English which do not pose issues. However, the presence of dialects in Pacific languages will still likely require a large training sizes in order to accurately learn and classify tweets regardless of dialects.

A deeper issue with adapting systems for Pacific language is that Pacific languages may have unique beliefs, customs, and values that may be reflected in their language use. Issues with different words, grammar, and dialects could theoretically be overcome by translating the Pacific tweets into English, and then processing them the same as an ordinary English tweet. However, this may not prove to be effective due to the different way that the language is used. For example, what is classified as an insult in a Pacific language may not be classified as an insult in English language due to what is fundamentally considered to be offensive in the two languages.

5.3 Performance in Different Context and Populations

Offensive language depends heavily on context. A comment with vulgar language directed at friends can be considered not offensive, and instead be a form of trusting joking around. However, a comment with vulgar language directed at a stranger can be considered offensive. Without the context associated with each tweet such as who the author is tweeting about or tweeting to, it is inherently hard to classify the tweet as offensive or not. If the context is a general tweet aimed at the public, then the systems will likely perform better, as it is unlikely that the author is making an inside joke or communicating with friends. Conversely, if the context is a tweet between associates, then the systems will likely perform worse, as it becomes more difficult to determine whether any offensive language is joking between friends or actually intended to be offensive.

Offensive language also depends on whether the individual making the comment is inside the group being offended. If an individual is part of the group targeted by the offensive language, then it is usually accepted by society to not be offensive, as otherwise the individual would be inherently offending themselves. However, if an individual is not part of the group targeted by the offensive language, it is more likely to be offensive language. Without the context of whether the tweet is made by an individual inside or outside the group being offended, the systems will perform worse.

Beyond Pacific languages, any population other than English will see a decrease in performance for the BERT classifier. This is because BERT has been pre-trained on an English text corpus with English text embeddings. In order for BERT to perform well in other languages, both the model and the WordPiece embeddings would need to be retrained on a large corpus of text in that language with a potentially different character scheme. It would be challenging to both collect a data set of such size, and to train a model as large as BERT.

6 Conclusion

The paper evaluates two systems on the shared task SemEval-2019 Task 6 on Identifying and Categorizing Offensive Language in Social Media (OffensEval) (Zampieri et al., 2019a). This task uses OLID (Zampieri et al., 2019b), a data set of 14,100 English tweets with a three level hierarchical annotation scheme. This evaluation compares a Bayesian Classifier with a BERT based classifier, amongst common benchmark models and baselines from Zampieri et al. (2019a). The Bayesian Classifier achieves an F1-score of 72.9%, whilst the BERT classifier achieves an F1-score of 82.3%. The different vocabulary and dialects may present challenges when adapting these systems for Pacific languages, as well as the difference in beliefs, values and hence what is considered offensive. More generally, the systems will struggle to generalise to non-English data sets, as well as depending greatly on the context of the tweet such as the relationship between the author and tweet target, and whether the individual is part of the offended group.

Future work includes further text tokenisation and normalisation such as stemming for the Bayesian Classifier, and varying parameter values for the BERT based classifier such as max sentences lengths and number of training epochs.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Technical report. ArXiv:1810.04805 [cs] type: article.

Jonathon Dunn. 2023. Natural language processing - linear text classification. Learn. Accessed April 13, 2023.

Ping Liu, Wen Li, and Liang Zou. 2019. NULI at SemEval-2019 Task 6: Transfer Learning for Offensive Language Detection using Bidirectional Transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to Fine-Tune BERT for Text Classification? Technical report. ArXiv:1905.05583 [cs] type: article.

Twitter. Counting characters. Accessed April 21, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. Technical report. ArXiv:1609.08144 [cs] type: article.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1415–1420, Minneapolis, Minnesota. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

A Code

All of the code to implement the Bayesian Classifier and BERT models, and the data used for testing and training can be found in the following repo: `https://github.com/GarethHarcombe/cosc442-supervised-systems`