XCS224U Final Project

Literature Review

Blake Norwick, Dongyao Wang, Xiuyu Yan

Table of Contents

* We've organized this section by the title of the papers we reviewed. Under the title of any given paper, we include a brief summary, any relevant connections to other works we reviewed and any ideas we might have to extend the ideas presented in this paper.

Problem Statement:

The general task we want to study is the use of Natural Language Processing techniques to detect abusive language online. We use abusive language as a bit of an umbrella term, entailing both overtly vitriolic language like Hate Speech and more subtle offensive language like condescension. In general, this is a text classification task like sentiment analysis, where the goal is to parse some text and label it as abusive or not.

Paper Reviews:

TalkDown: A Corpus for Condescension Detection in Context

Condescending language differs from other harmful languages as it is generally used unconsciously with good intentions. Given the nature of condescension, the author presented TalkDown, which is a new labeled dataset derived from Reddit of condescending linguistic acts in context.

The data comes from Reddit data dump 2006-2018. After initial pattern-based extraction methods, the author conducted an annotation project on Amazon Mechanical Turk to get some insights and further processed the data. The final dataset consists of annotated positive and negative instances, with randomly sampled negative instances. The dataset was then partitioned into 80% train, 10 % development, and 10% test splits. The model used is BERT, and BERT Large produces the best macro-F1 Score. The author found that considering context (in this experiment, it is combining the quoted part of the text and the context) can provide a 3-4% boost in performance. A balanced dataset yields the best result, and for the imbalanced dataset, an oversampling ratio of 2 to 4 is ideal.

Don’t Patronize Me! An Annotated Dataset with Patronizing and Condescending Language towards Vulnerable Communities

Patronizing and condescending language (PCL) is an interesting technical challenge for the NLP. This paper introduces the Don’t Patronize Me! Dataset, which contains more than 10,000 annotated paragraphs extracted from news stories at the text span level. The categories of PCL is provided: the savior (with unbalanced power relations and shallow solution as sub-categories), the expert (with presupposition and authority voice as subcategories) and the poet (with metaphor, compassion and the poorer, the merrier as subcategories).

Annotation task consists of paragraph-level identification and a further span-level PCL labeling process with BRAT rapid annotation tool.

Several experiments are conducted using different models such as SVM, random, BERT, roBERTa, DistilBERT. For task 1, BERT-based methods achieve the best results, with RoBERTa performing slightly better than DistilBERT and BERT-base. For task 2, RoBERTa outperforms the rest models in all the categories except for Authority Voice, where BERT-large gets the best result. However, the challenge remains from the fact that accurately modeling such subtle nuances requires knowledge of the world and common sense. Overall, the BERT based model outperforms simpler methods.

Predicting Different Types of Subtle Toxicity in Unhealthy Online Conversations

The author analyzed a dataset comprised of 44000 labeled comments of unhealthy conversations. NLTK VADOR is used for sentiment analysis, they found that all types of unhealthy comments except sarcastic and condescending resulted in slight negative results. Condescending and healthy produced the most neutral sentiment scores while sarcastic provides the most positive score. Hostile comments are the most negative. Among the models used, CNN- LSTM model achieves the best result to detect unhealthy conversation. But detection of sarcasm was still the most difficult for the CNN-LSTM model.

If our goal is to detect the subtleness of toxic speech such as condescending language detection, then sentiment analysis wouldn't be very helpful since the score would more likely be neutral from the third paper. The third paper uses deep learning architecture(CNN-LSTM), whereas the first two papers demonstrated that BERT model outperforms other models if the sentence contains obvious hostile or insulting words, current models generally perform well. However, all three papers suggested that detecting condescending and sarcasm is still challenging for current models and there is considerable room for improvement. As for now, deep learning model and BERT-based model generally provide better results. Rich contextual representations would be very useful for improving performance. In the future we might increase the variety of sources (news, social media, etc.), include the context while training the model and be sure to use a large and balanced dataset.

Multi-Task Learning with Sentiment, Emotion, and Target Detection to Recognize Hate Speech and Offensive Language

They experiment on existing data sets for each of these concepts (sentiment, emotion, target of HOF). Firstly, make sentiment analysis(positive and negative). Second, emotion analysis is concerned with the classifier. Finally, the target is, by

definition, a crucial element of hate speech, whether mentioned explicitly or not. The model of this experiment is transformer-based,the model that achieves the best performance in the final evaluation considers the concepts of emotion, sentiment, and target together, A clear downside of this model is its high resource requirement: it needs annotated corpora for all the phenomena involved, and as this model selection experiments showed, the quality of these resources is very important

Towards Offensive Language Identification for Tamil Code-Mixed YouTube Comments and Posts

During recent year, increasing media user from different countries present their idea by using multiple language, which is know as code-mixed form. This form of language make offensive speech identification by the algorithm more challenging. The author proposed a methods to classify offensive language from a Tamil code-mixed dataset of comments and posts collected from YouTube. While Transfomer-based models have some good attributes for this multiple language task, those models are trained on languages in their native script, not in the romanized script in which users prefer to write on- line. The author proposed a method to selective translate and transliterate, experiment in code-mixed and romanized scripts. The dataset used in this paper was taken from a Offensive Language Identification in Dravidian Language (ODL) competition. Which is a multi-class classifica-tion problem with multiple offensive categories (Not- Offensive, Offensive-Untargeted, Offensive-Targeted-Insult-Individual, Offensive-Targeted-Insult-Group, Offe- nsive-Targeted-Insult-Other, Not-Tamil), then preprocessing the text and selective translation. Experiment with fine-tuning of state-of-the-art Transformer architectures. Comparing the result of some models used by this paper, ULMFiT models attained a better performance in predicting the minority classes as well. The major reasons for the better performance of ULMFiT over other models are due to its superior fine-tuning methods and learning rate scheduler.

Unlike *Multi-Task Learning with Sentiment, Emotion, and Target Detection to Recognize Hate Speech and Offensive Language,* which places emphasis on detecting toxic speech by focusing on sentiment and emotional analysis and target of the text, this paper focuses on social media forms of text. It’s detecting toxic words by preprocessing text with selective translation and transliteration, and transforming them to a social media-form.

Agreeing to Disagree: Annotating Offensive Language Datasets with Annotators’ Disagreement

Offensive language detection relies on supervised learning, so the performance of the classifier depends deeply on the accuracy of the annotation of the dataset. To

reduce the need for annotated data, some ideas have been proposed to deal with the problem from an algorithmic perspective. Some argued that they should discard some ambiguous data in order to improve the ability of the classifier. Compared to existing works, the contribution of this paper is different in that they are interested mainly in the process of dataset creation rather in evaluation metrics or classification strategies. To pre-evaluate the tweets data-set, they use a heuristic approach by creating an ensemble of 5 different classifiers, all based on the same BERT configuration and fine-tuned starting from the same abusive language dataset (Founta et al., 2018). They compared the result of the algorithm and annotators.

Finally, they contribute a data annotation process and a thorough set of experiments for assessing the effect of (dis)agreement in training and test data for offensive language detection. They showed that an ensemble of classifiers can be employed to preliminarily select potentially unambiguous or challenging tweets. By analysing these tweets we found that they represent real cases of difficult decisions, deriving from interesting phenomena, and are usually not due to low-quality annotations.

In the future, we might try to experiment with methods to reduce the reliance on annotation work.

Fine-Tuning of Pre-Trained Transformer Networks for Offensive Language Detection

In *Fine-Tuning of Pre-Trained Transformer Networks for Offensive Language Detection*, Wiedemann et. al present a state-of-the-art approach to the *Offensive Language Detection Shared Task* in the International Workshop on Semantic Evaluation. The *Offensive Language Detection Shared Task* actually consists of a series of sub-tasks. Task A asks whether a Tweet is offensive, Task B distinguishes between generally offensive language and language aimed at a specific party, and Task C asks whether the affected party is a group, individual or other. Wiedemann et. al. show the benefit of fine-tuning transformer networks, like BERT, on the offensive language detection task, which is perhaps unsurprising, given the success of such approaches in other text classification tasks. Following the suggestion of Sun et al. in *How to fine-tune BERT for text classification,* Wiedemann et. al. further pre-train the RoBERTa model on the Masked-Language Model (MLM) task on in-domain data, which in this case was a collection of Tweets, some of which have offensive content. Finally, Wiedemann et. al. use an ensemble of RoBERTa models pre-trained on the MLM task with a majority voting scheme to make a prediction.

Detection of Abusive Language: The Problem of Biased DatasetsDetection of Abusive Language: The Problem of Biased Datasets

While the success of Wiedemann et. al. is encouraging, Wiegand et. al. in *Detection of Abusive Language: The Problem of Biased Datasets* show how models trained to detect abuse language suffer to generalize, often due to biases in the underlying data on which they were trained. To study the extent to which datasets might suffer from biases, Wiegand et. al train a single model architecture (FastText) on the standard binary classification task (i.e. Is this text abusive or not?) and use the F1 score of the model as a measure of its success. In their study of the biases that affect popular datasets curated for the task of offensive language detection, Wiegand et. al make the distinction between explicitly and implicitly offensive language. Naively, explicitly offensive language might seem easier for NLP systems to detect, as it by definition contains at least one abusive word, according to a lexicon of abusive words cataloged by Wiegand et. al in a previous paper. While this naive hypothesis was largely confirmed by their work, Wiegand et. al. identify one popular dataset, *Waseem*, which featured a higher incidence of implicitly abusive language without the (expected) corresponding decay in the F1 score of the FastText classifier. However, as part of their work to understand why this was the case, Wiegand et. al learned that the *Waseem* dataset suffers from *Topic Bias*. Specifically, much of the abusive language in this dataset is drawn from a similar topic/domain so the classifier learns spurious correlations between words that are commonly observed in conversations on this topic, but are not inherently abusive, and the abusive language label. They note that such a phenomenon is not unique to the *Waseem* dataset. Wiegand et. al identify another class of biases that is common to offensive language datasets which they call the *Author Bias*. Again in the *Waseem* dataset, Wiegand et. al. note that a majority of the sexist Tweets originate from a couple Twitter users; thus, a classifier that simply uses the identity of the author of the Tweet to label a Tweet as sexist or not is quite successful on this dataset. While this is obviously troublesome, because it suggests that the models might not actually be encoding an understanding of abusive language in their networks, it also points to the power of using contextual information in identifying abusive language, which Schmidt et. al note in *A Survey on Hate Speech Detection using Natural Language Processing.*

This paper is similar to *Agreeing to Disagree: Annotating Offensive Language Datasets with Annotators' Disagreement***,** in that they both acknowledge the outsized role that the composition of the underlying dataset has on the quality of the model trained on it.

Neural Word Decomposition Models for Abusive Language Detection

In *Neural Word Decomposition Models for Abusive Language Detection*, Bodapati et. al. experiment with several techniques to cope with the reality that abusive language often consists of obfuscated words (like *w0m3n*) specifically written to evade automatic abusive speech detection. Specifically, Bodapati et. al note that “adding character based embedding to aid word embedding based models, and subword models enhance performance over their pure word based modeling baselines”, adding that “end to end character models aren’t as effective”, which perhaps isn’t too surprising. Interestingly, Bodapati et. al learned that the use BERT’s BPE tokenizer with a FastText classifier was superior to the use of a FastText classifier with a BPE tokenizer (with the same vocabulary size) trained on the abusive language dataset, which provides more evidence that the abusive language detection task, like many text classification tasks, benefits from knowledge encoded in pre-trained language models. Interestingly, a FastText or TextCNN classifier with a BERT Tokenizer is superior (albeit, slightly) to a fine-tuned BERT model on the task of identify Hate Speech in Tweets, whereas a fine-tuned BERT model is superior (again, albeit slightly) to both the FastText and TextCNN in identifying abusive language in the Personal Attack and Toxicity datasets drawn from Wikipedia; perhaps this isn’t entirely surprising as those datasets are more lexically similar to the datasets on which BERT was trained.

A Survey on Hate Speech Detection using Natural Language Processing

In *A Survey on Hate Speech Detection using Natural Language Processing*, Schmidt et. al argue that broader contextual information is often necessary to classify speech as abusive or not. They take the following sentence as an example: “Put on a wig and lipstick and be who you really are”. In isolation, this isn’t an inherently abusive statement. However, in certain contexts, such a statement might be meant to malign the sexuality or gender identity of the individual whom the speaker is addressing (Dinakar et. al, 2012). Given that such context was provided to a model, it’s not entirely clear how to encode such context to feed to a model. However, Schmidt et. al. conclude the paper with a call to include more “knowledge-based” features, the context in the above example, into models, so perhaps this is an idea to consider as we finalize what form exactly our final project will take.

References:

Anna Schmidt, Michael Wiegand (2017). *A Survey on Hate Speech Detection using Natural Language Processing*. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*. https://aclanthology.org/W17-1101.pdf

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. arXiv preprint arXiv:2011.08320.

Charangan Vasantharajan · Uthayasanker Thayasivam 2021.Towards Offensive Language Identification for Tamil Code-Mixed YouTube Comments and Posts arXiv:2109.13563

Elisa Leonardelli, Stefano Menini, Alessio Palmero Aprosio, Marco Guerini, Sara Tonelli Agreeing to Disagree: Annotating Offensive Language Datasets with Annotators' Disagreement arXiv:2109.13563

Flor Miriam Plaza-del-Arco, Sercan Halat, Sebastian Padó, Roman Klinger Multi-Task Learning with Sentiment, Emotion, and Target Detection to Recognize Hate Speech and Offensive Language arXiv:2109.10255

Gilda, Shlok & Silva, Mirela & Giovanini, Luiz & Oliveira, Daniela. (2021). Predicting Different Types of Subtle Toxicity in Unhealthy Online Conversations.

Gregor Wiedemann, Seid Muhie, Chris Biemann (2020). *Fine-Tuning of Pre-Trained Transformer Networks for Offensive Language Detection.* https://arxiv.org/pdf/2004.11493.pdf

Karthik Dinakar, Birago Jones, Catherine Havasi, Henry Lieberman, and Rosalind Picard. 2012. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. ACM Trans. Interact. Intell. Syst., 2(3):18:1–18:30, September.

Michael Wiegand, Josef Ruppenhofer, Thomas Kleinbauer (2019). *Detection of Abusive Language: the Problem of Biased Datasets*. In *The 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. https://d-nb.info/1190084805/34.

Sravan Babu Bodapati, Spandana Gella, Yaser Al-Onaizan (2019). *Neural Word Decomposition Models for Abusive Language Detection*. In *Proceedings of the Third Workshop on Abusive Language Online.*

https://assets.amazon.science/23/d7/6843513743cf9fbd3cce1532d20e/neural-models-for-abusive-language-detection.pdf

Zijian Wang and Christopher Potts. 2019. Talkdown: A corpus for condescension detection in context. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing.