Social Computing Project: Intermediate Report 1

Data Augmentation for Hate Speech Detection

Sayar Ghosh Roy* 20171047 | **Souvik Banerjee*** 20171094 | **Saujas Vaduguru*** 20171098 | **Ujwal Narayan*** 20171170

⚠ This document contains samples of online social media hate speech which are present in the publicly available datasets we are currently studying. These are included for illustrative purposes only and in no way reflect the views of the authors. Please note that these samples are highly derogatory. Reader discretion is advised.

In this document, we describe the progress made by 15th March, 2021, and sketch our plans for the rest of the project. Our work so far falls along three axes:

1. Developing a common experimental pipeline to have a uniform mode of training and evaluating models using different augmentation methods
2. Reproducing baselines from prior work
3. Improving over baselines using modified baselines or novel methods

1 Experimental pipeline

In this section, we describe our overall test-bench for evaluating our data augmentation strategies. Our code is publicly available at https://github.com/sayarghoshroy/Augment4Gains/.

1.1 Dataset

The language in use on Twitter is in a different text style as compared to day-to-day speech, formally written articles, and web-pages. The Twitter platform's style of text is full of emojis, smileys, hashtags, acronyms, abbreviated forms of words and phrases, orthographic deviations from standard forms including dropping of vowels from certain words, and instances of code mixing. Most of the publicly available datasets for hate speech detection are composed of Tweets. Since our augmentation approaches rely on both lexical and semantic cues, we require a data-source containing grammatical text that is more in-line with standard English language style.

In such a setting, we consider Qian et al. (2019). Their dataset contains posts which are scraped from the Reddit[1] and Gab[2] platforms and are much more structurally sound as compared to Tweets. The dataset contains human annotated counter narratives specific to particular comment threads. In addition, they provide a list of indices of comments which contains instances of hate. We extract each user response in every post from these datasets separately and lookup the corresponding indices list to assign a binary label to every single user response, namely 1 for hateful, offensive or derogatory and 0 for otherwise. Each response, composed of multiple sentences is thus one datapoint.

As a pre-processing step, we remove information such as hashtags, mentions, URLs, emojis, etc. from every datapoint. We only leverage the cleaned text without any additional features for our augmented sample generation and classification tasks. For each dataset, we randomly divided the data in a 70:10:20 ratio (as done in Qian et al. (2019)) into training, validation and testing splits (following the decision). Our processed Reddit dataset contains 15619 train samples, 2231 validation samples, and 4464 test samples while the processed Gab dataset carries 23643, 3377, and 6756 samples for training, validation, and testing respectively.

For our final report, we will include a third dataset based on Tweets[3]. Although our augmentation approaches are designed keeping standard English language text in mind, we will evaluate how each of these techniques work out on cleaned Tweet texts. Throughout this and further documents, we will refer to our three datasets as simply, Data_{Reddit}, Data_{Gab}, and $\text{Data}_{Twitter}$.

*Equal contribution. Order determined by roll number.

[1] www.reddit.com
[2] gab.com
[3] twitter.com

1.2 Classification Model

Fine-tuning of pre-trained Transformer models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) are becoming the new baseline for various tasks in the Natural Language Processing domain. For evaluating our data augmentation approaches, we utilize a pre-trained Transformer-based model (such as BERT or RoBERTa) as a text encoder paired with a Multi-Layer Perceptron (MLP) classifier head that considers the final-layer output embedding of the `[CLS]` token. We utilize the Adam optimizer to train our classification architecture for a total of 8 epochs and save the model weights corresponding to the point in training that manifested the least validation loss.

One way of dealing with class imbalance in the base as well as the augmented datasets is to weigh each class differently during training.

- $N_p =$ # positive samples in training
- $N_n =$ # negative samples in training
- $N = max(N_p,\ N_n)$

We set the weights of the positive and negative classes as $\frac{N}{N_p}$ and $\frac{N}{N_n}$ respectively. We implement the option of weighing terms in the classifier loss accordingly to observe the effects of class imbalance.

1.3 Classification test bench

We require a uniform scheme of evaluating all of our augmentation approaches. Therefore, for each dataset under consideration (be it Data_{Reddit}, Data_{Gab}, or $\text{Data}_{Twitter}$), we lock the validation and the testing data. All the augmented examples are based solely on the training samples. This is to ensure that there is no seepage of information and as such, the classification models do not indirectly look at some modified version of a test (or validation) sample.

Every augmentation method can be viewed as an algorithm that takes in a sample of text as input conditioned on which one or more revamped samples are produced. In our pipeline, the in-question augmentation procedure is applied to all the training datapoints. The generated samples are appended to the existing training split resulting in the augmented train data-frame. The classification test-bench utilizes the newly created training set, taking care of the downstream steps.

2 Baselines

2.1 Easy Data Augmentation

While there are a variety of techniques involved in data augmentation for text classification tasks, one of the most easiest yet surprisingly performant one is the Easy Data Augmentation (EDA) method (Wei and Zou, 2019). EDA consists of the following four basic operations.

- **Synonym Replacement (SR)**: Here n randomly selected words are replaced with one of it's synonyms chosen at random. These synonyms are generated by querying WordNet (Miller, 1995), however it's important to note here that for synonym replacement all the senses of the word are considered and not just the sense it is currently being used in.

- **Random Swap (RS)**: Here two words are randomly chosen from the sentence and their positions are swapped. This operation is repeated n times

- **Random Deletion (RD)**: Here each word of the sentence is randomly removed with the probability p

- **Random Insertion (RI)**: Here a randomly synonym of a randomly selected word is inserted randomly into the sentence. This operation is then repeated n times

For the synonym specific operations i.e. RI and SR only content words are chosen and the stop words are ignored. While the methods are simplistic and do not lead always lead to generation of high quality grammatically correct data, they produce significant performance boosts. Wei and Zou (2019) also observe the newly generated sentences occupy positions closely around the original sentences in the latent semantic space and thus the meaning and therefore the labels do not significantly change. For operations such as SR, RI and RS, in order to prevent the new sentences from being too noisy we formulate n to be dependent on the length of the sentence l, given by $n = \alpha l$ where α indicates the percentage of the sentence that will be modified. Other than these hyper-parameters, we also cap the number of generated sentences per source sentence with a parameter n_{aug} so as to avoid pollution of the source data.

Original	EDA	ThreshAug
you should watch louis le vau 's latest video . steven oh of tyt is disturbing as hell and makes me hope that jimmy dore wakes the left up .	you should watch louis le vau s latest video steven oh of tyt is disturbing take go out as hell and makes me hope that jimmy dore wakes the left up	you should watch liam le vau's latest youtube . steven ooh of tyt is disturbing as shit and makes me hope that bruce dore wakes the left up.
kill yourself you whiny , self-righteous faggot .	kill yourself you whiny ego righteous faggot	let yourself you whiny, self-absorbed fag.
but why do they make that face	but why do they shuffle that face	but why do they make that look

Table 1: Some samples from the training set, along with corresponding augmented versions with EDA methodology and *ThreshAug* (Glove-50 + POS + cosine similarity).

2.2 Synonym substitution

In order to make relevant substitutions of words with synonyms without using tools external to neural networks (such as a thesauraus, etc.), we make use of pre-trained word embeddings (GLoVe in our actual implementation). These embeddings allow us to determine the relative similarity between each word in the vocabulary space of a text corpus. Now the question remains about which words to substitute. A substitution is determined by two factors. Firstly, any potential replacement word must exceed the cosine distance threshold t, where $t \in [0, 1]$ and it must match the POS-tag assigned to the word. The intuition behind the inclusion of both the above requirements is that two words must have been seen in sufficiently equal contexts such that one can be replaced with the other without changing the sentence semantics. We follow Rizos et al. (2019) and use POS tags to choose words of only very specific tags like common nouns, adjectives and verbs. This method, termed *ThreshAug* forms a baseline for the synonym replacement approaches we explore.

3 Improvements

3.1 Contextual synonym substitution

Rizos et al. (2019) propose a way to use word embeddings to identify suitable synonyms to substitute in augmented samples. However, words can have multiple senses, and word embeddings do not provide a way to distinguish which sense is being used based on words in the context (Arora et al., 2018). We have explored two directions in incorporating context information into the choice of synonyms for replacement. The first relies on contextual word representations, and the second relies on incorporating contextual knowledge into choosing replacements from a lexical database.

3.1.1 Contextual word representations

With the advent of contextualized word representations (Peters et al., 2018; Devlin et al., 2019), we have access to methods that can compute word embeddings that can allow for sense disambiguation based on context.

Kobayashi (2018) propose using contextual embeddings in the data augmentation pipeline for text classification tasks. Similar to Rizos et al. (2019), who select words for substitution and determine the alternatives using word embedding similarity, Kobayashi (2018) select words to substitute, and use a bidirectional language model to choose the word to be substituted. The intuition behind their method is that a language model is likely to choose synonyms as alternatives, and these choices are made with the context in consideration, resulting in a more informed augmentation technique.

Using the language model, they obtain the distribution $p(w_i'|S \setminus \{w_i\})$, where S is the sample, w_i is the word chosen for substitution, and w_i' is the alternative word. Then, they sample from the distribution $p(w_i'|S \setminus \{w_i\})^{\frac{1}{\tau}}$, which is the distribution predicted by the model annealed with a temperature. The temperature hyperparameter allows us to control the strength of the augmentation, with smaller values more faithfully choosing the most likely words, and larger values allowing the model to choose lower probability words more often resulting in more diverse augmented samples.

Since the publication of Kobayashi (2018), there have been significant strides in large, pretrained lan-

Original	MLM	WSD+WordNet
you should watch louis le vau 's latest video . steven oh of tyt is disturbing as hell and makes me hope that jimmy dore wakes the left up .	you should enjoy louis le vau's bloody claudius. steven oh of tyt is disturbing as hell and makes me wish that heather dore wakes the wolf up.	you should enjoy louis le vau's bloody claudius. steven oh of tyt is disturbing as hell and makes me wish that heather dore provokes the left up.
kill yourself you whiny , self-righteous faggot .	save yourself you whiny cold self - righteous faggot.	kill yourself you complaining , self-righteous faggot .
but why do they make that face	but why do they make their face	but why do they make that visage

Table 2: Some samples from the training set, along with corresponding augmented versions with MLM-based augmentation and WSD+WordNet-based augmentation.

guage models. In this project, we explore the use of these large pretrained models to perform augmentation in a similar way. The formulation put forth by Kobayashi (2018) naturally lends itself to the masked language modelling (MLM) task, which is used as the pretraining task for Transformer-based language models (TLMs) (Devlin et al., 2019).

We use the MLM task to generate synonyms for augmentation. We choose whole words (as opposed to tokens in the TLM vocabulary, which tend to be subwords) with a fixed probability, and replace these words in the data with the `[MASK]` token. We then pass these to a TLM with the MLM head, and obtain the distribution over tokens that can fill the `[MASK]` token. This distribution, which is equivalent to $p(w_i'|S \setminus \{w_i\})$, is then used to sample replacements and obtain augmented samples. We will choose the temperature value for sampling based on downstream performance on the classification task.

3.1.2 WordNet-guided synonym substitution with word sense disambiguation

Words can often have multiple meanings. Thus when augmenting synonyms it becomes important to identify the right sense of the word so that the right synonyms of that word can be found. This problem of finding the correct sense of a word in text is termed as "Word Sense Disambiguation". While the problem is often termed as an AI complete problem, recent work utilising Deep Neural Networks have made significant headway into it. Transformers such as BERT (Devlin et al., 2019) have given state of the art results on many NLP tasks and this task is no different. We follow the approach of Yap et al. (2020), where both BERT and WordNet (Miller, 1995) are leveraged to find the right sense of the word.

We first need to identify the right phrases to substitute or replace and thus we chunk the sentence. Once the sentence is split into multiple chunks, we check if that particular chunk is present in WordNet . If it is present, we then leverage BERT to rank all the senses of the word in the context of the sentence. We pick the sense with the highest rank as the correct sense of the word. If the chunk is not present in WordNet , we back-off and search through all the words of the chunks individually and repeat the process we mentioned earlier to identify the sense of the word. Once we found the right sense of the word, we query WordNet again to retrieve the set of synonyms and hypernyms associated with the particular word or phrase. Once we have the synonyms, we substitute these synonyms in-place of the original word or phrase to generate new samples for training.

4 Next steps

As discussed with our mentor, we used this time to perform small exploratory studies and we will run full experiments and present results in future deliverables.

So far, we have focused on alteration-based approaches to data augmentation. Another dimension we hope to explore for the upcoming deliverables is using generation-based approaches to augmentation. We also plan to further improve and vary the alteration-based approaches we have experimented with so far.

4.1 Contextual synonym substitution

One important challenge in the contextual synonym substitution method is the incorporation of class information in the augmentation process. This is a step towards ensuring that the model does not choose words that change the label as alternatives to words in the original data.

Kobayashi (2018) propose a method for class conditional replacement prediction with bidirectional LSTM language models. They fine-tune their model on the original training data for the task to predict $p(w_i'|S \setminus \{w_i\}, y)$, where y is a label embedding that allows the model to learn the correlation between labels and word choice.

We are in the process of exploring ways of modifying the MLM head for the TLMs that we use to incorporate class information, and hope to have a data augmentation model that can incorporate class information for augmentation.

Another avenue we hope to explore is determining what types of words we choose to mask based on linguistic criterion like part-of-speech information. We are currently working on this, and hope to present results in the next deliverable.

References

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2018. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang. 2019. A benchmark dataset for learning to intervene in online hate speech.

Georgios Rizos, Konstantin Hemker, and Björn Schuller. 2019. Augment to prevent: Short-text data augmentation in deep learning for hate-speech classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 991–1000, New York, NY, USA. Association for Computing Machinery.

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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Boon Peng Yap, Andrew Koh, and Eng Siong Chng. 2020. Adapting BERT for word sense disambiguation with gloss selection objective and example sentences. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 41–46, Online. Association for Computational Linguistics.