Social Computing Project: Intermediate Report 2

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 will describe our text generation-based approaches to data augmentation for hate speech detection. For details on alteration based approaches, kindly refer to Intermediate Report 1.

Broadly speaking, our approaches to data augmentation can be classified into two major types.

1. **Augmentation through perturbation**

   In this space, we create new training data by modifying or perturbing the existing training datapoints. The approaches that we discussed in Intermediate Report 1 such as synonym substitution including MLM based augmentation and WordNet based substitutions and methodologies such as Easy Data Augmentation (EDA) (Wei and Zou, 2019) fall under this category.

2. **Augmentation through generation**

   Within this type of augmentation, we do not modify the existing training data. New training samples are generated from scratch. Approaches including the use of Variational AutoEncoders (VAE) and encoder-decoder models fall under this category. We've made significant progress on this front and would be describing the same through this document.

Augmentation through Generation

After exploring alteration-based approaches to augmentation, we turned our attention to generation-based approaches. These methods create augmentation samples on their entirety using specific Natural Language Generation (NLG) models. Now, a generation model may be trained to generate samples belonging to a particular class without any other constraints, or to generate samples that are constrained to 'mean' the same thing as a given input sentence. We explore both of these paths.

Class-based generation

We present two methods to generate samples constrained only by the class label. The first is a baseline method proposed by Rizos et al. (2019) called `GenAug`, where a language model is trained upon data from a single class and then leveraged to generate samples. The second is an approach not explored in the context of data augmentation for hate-speech detection – the use of Variational AutoEncoders (VAEs).

Paraphrase generation

Paraphrases can be defined as sentences conveying the same meaning but with different surface realizations. In this report, we use "augmentation through paraphrasing" to only refer to the paraphrases generated from scratch as paraphrases generated through substitutions or replacements have been covered earlier under the banner of "augmentation through perturbation". Augmentation through perturbation adds extremely similar data samples to the training pool as the generated samples have very similar syntactic structure and text style. By throwing samples that are generated from scratch into the training pool, we expect the trained classifiers to be more robust, and in that, we expect better performance on unseen data.

[*] Equal contribution. Order determined by roll number.

1 Class-based generation

1.1 GenAug

The main idea behind GenAug is extremely similar to that of using RNNs (LSTMs/GRUs) for Natural Language Generation. Training such a model is synonymous to training a word level language model. Hence, for making inferences using such models, we start with a random word from the vocabulary and attempt to predict each in-sequence next word based on the generation so far. Our implementation, specifically takes N words as input and converts each into a 100-dimensional word embedding vector. This sequence of vectors is then passed through a bidirectional-LSTM layer with 128 hidden units each. Finally, the output of the bi-LSTM is fed into a final FCNN layer. The final output represents a probability distribution over each token in the vocabulary the argmax of which produces the output token.

1.2 VAEs

We also explore the use of VAEs for generating samples within each class. We adopt the approach presented by Bowman et al. (2016). The model is trained as an auto-encoder with an encoder function $f_{enc}(x)$ that maps an input x to a learned code z, and a decoder function f_{dec} that recovers x from the code z. The model is trained to recover the input (with the difference forming one loss term) while keeping the posterior distribution $q(z|x)$ modelled by f_{enc} close to a prior p (with the Kullback-Leibler divergence between p and q forming the second loss term). For the application of text representation learning, the functions f_{enc} and f_{dec} are parameterised as recurrent neural networks.

We train a single VAE model for each class, and then generate samples for each class by sampling a random point z' in the latent space of the model's learned code, and using the decoder to generate a sequence conditioned on z'. The intuition being that a model corresponding to a particular class learns a latent space of codes corresponding to the distribution of samples in that particular class. Then, any point in that latent space is the representation of some sample belonging to that class, and we can decode that sample by conditioning on the point in the latent space.

We adapt an implementation of Bowman et al. (2016)[1] for our application. Examples of samples generated via class based generation can be found in Table 1.2.

2 Paraphrase generation

In order to have a scalable method of generating new labelled samples for hate speech detection, we need to ensure that we can assign a gold standard label to a newly produced sample based on certain heuristics that take the immediate environment of the generation process into account. For example, if we perturb a sentence replacing certain nouns with other synonymous nouns having the same word sense, we do not expect the meaning of the sentence to change and in turn, we can assign the same 'is hate-speech' label to the new perturbed sentence. Similarly, a paraphrased version of a given sentence, by definition, will carry the same semantics and have the same level of hateful content as the original sentence.

In this section, we will describe our two proposed data augmentation techniques that aim to generate a paraphrase of the input sequence. The idea is simple: use an encoder that 'understands' and produces a latent representation for the input sequence. And then generate a new token sequence relying upon the created latent representation such that the core semantics of the newly produced text remains invariant as compared to the source. In general, this idea differs sharply from that of class-based generation where a specific trained language model is used to generate a sequence conditioned upon a small input prompt.

As a baseline method of generating paraphrases, we consider text-to-text Variational AutoEncoders. Note that VAEs have never been used for creating synthetic data meant for augmenting hate-speech datasets. We present further details on our VAE-based experiments in Section 2 and put it forward as our baseline paraphrase generation technique.

Moving from the unsupervised techniques to our proposed supervised model for paraphrase generation, we leverage publicly available supervised datasets (refer to Section 2.2.1) for generating paraphrases of sequences. We utilize state-of-the-art Transformer based encoder-decoder models which are then fine-tuned on a set of these supervised datasets in order to serve as our paraphraser.

2.1 Text-to-Text VAEs

A trained VAE plus a decoder can be intuitively thought of as an unsupervised paraphraser because

[1] https://github.com/timbmg/Sentence-VAE

Label	GenAug	VAE
0	depression quest which is why you need to put in a long time. i don't see the biggest mistake of all the time. it 's a huge issue that they can think of the law. However, that's not a case that is just a fucking idiot.	i don't know how to be a hypocrite to be a hypocrite
1	m afraid but i try not to let it affect my relationships and then the victim has a woman, she was doing the same thing with her pussy pass. you 're a cunt . you 're a fucking idiot . i think that's a good thing to be fair, it has a lot of people	the study will be a liberal utopia.

Table 1: Samples obtained using class-based generation methods for augmentation. Label 0 indicates 'not hate speech' and label 1 indicates 'is hate speech'.

of its functionality — go though an input sequence, create a non-textual representation of its distributional semantics (an embedding) and use the same to produce a new sequence (of approximately similar 'meaning').

We utilize the same training set up described in Section 1.2 and instead of decoding from a random point in the latent code space, we embed an input sequence in the code space, and decode a sequence of the same (expected) meaning from the embedding.

2.2 Transformer-based Paraphraser

To summarize, we utilize a fully trained Transformer-based sequence to sequence architecture for paraphrase generation. Instead of randomly initializing the Transformer weights prior to fine-tuning, we choose a pre-trained Transformer model that has already 'witnessed' and learnt from large chunks of natural language text. Such a pre-trained text-to-text Transformer is then fine-tuned auto-regressively on supervised input-source to paraphrased-target mappings. Like our VAE-decoder models, during inference, we consider the first 128 tokens for each source text and generate sequences having the same figure as maximum length.

2.2.1 Text to Text Transformers

Experiences with T5

T5 (Raffel et al., 2020) was one of the first highly acclaimed encoder-decoder Transformer architectures. Its novelty lies in the ability to perform any sequence to sequence natural language processing task using just one trained model by the use of prompts. Thus, in order to translate a sentence from English to French, a prompt[2] such as 'Translate to french' would work. Similarly, one could have prompts like 'Summarize', 'Regression', 'Classify', 'Get parse', and so on. Although intuitive and ambitious, the actual resultant outputs from the pre-trained T5 model do not seem very human-like. Moreover, upon experimenting with T5 models trained on standard paraphrasing datasets, we experienced the issue of polarity reversal. Essentially, a sentence such as 'He did not do so' would erratically change to 'He did do so' thereby defeating the purpose of writing a paraphrase.

Briefly reviewing BART and PEGASUS

Recently, Facebook's BART (Lewis et al., 2019) has proven very effective as a pre-trained encoder-decoder Transformer. As opposed to T5, the idea around using BART is to train and store task specific models that generate very high quality data. BART was once the state-of-the-art model for automatic abstractive text summarization and has been applied to domain specific text summarization tasks yielding state-of-the-art results (Ghosh Roy et al., 2020). A more recent venture into the area of text-to-text Transformers was carried out by Google and they released their PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization) (Zhang et al., 2020) model. Not only did it beat BART, but it also introduced a flavour of versatility being able to work with sequences of varying sizes — ranging from very short sequences

[2] A colon separates the main input from the prompt.

Original	VAE-paraphrase	Tf-Paraphraser
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 .	good luck , you can get arrested for her to $< unk >$. she 's a good thing to her , and she 's minding her to the ground . she 's just that she 's a good thing to her . she 's just a $< unk >$, she 's just going to be honest , but you 're not going to be honest .	louis le vau's latest video is disturbing and makes me hope that jimmy dore wakes the left up.
kill yourself you whiny , self-righteous faggot .	'm not sure how much they are n't like it. i 'm not gonna be able to be so much as well as well as well as well as well as well as well as well as well as well as well as well as well as well	You are self-righteous and should kill yourself.
but why do they make that face	the video is the same as a whole thing , the only one is the same as the only one who thinks the other party is the same as the $< unk >$ is a good thing to the character . if you 're a great , then the other engages isn't a liberal , and the other option isn't a good thing to the other side of the $< unk >$	Why do they make that face?

Table 2: Some samples from the training set along with corresponding augmented versions using VAEs for paraphrasing, and Tf-Paraphraser (Transformer-based Paraphraser)

right upto large research papers and patent documents. They introduced the concept of sentence masking and experimented with the selection of specific (important) to-be-masked sentences.

Choice of pre-trained Transformer

After manually studying the qualities of various output sentences from these pre-trained Transformer encoder-decoder models, we found that PEGASUS produces the best looking sequences with the least amount of noticeable grammatical errors (something which is corroborated in their publication as well). Therefore, we chose to utilize PEGASUS as our base pre-trained Transformer model for the paraphrasing task. We used Hugging Face's implementation of PEGASUS[3] with number of beams for decoding set to 10, a maximum sequence length of 128, with all other hyperparameters set to their default values.

It is also to be noted that paraphrasing and abstractive summarization are both sequence-to-sequence tasks where the semantics of the output sequence does not deviate from that of the input, in that, we do not expect the model to add new information or to alter existing information bits. As opposed to summarization, a paraphraser would ideally not prune out any information pieces while expressing the source sequence in 'its own words'. We have shared some examples to illustrate the quality of language generation for our paraphrasing models in Table 2.

2.2.2 Datasets related to paraphrase generation

In this subsection, we will quickly review two publicly available datasets having supervised mappings from source texts to their paraphrases. PAWS-Wiki (Zhang et al., 2019) contains a collection of sen-

[3] huggingface.co/transformers/model_doc/pegasus.html

tence pairs sourced from Wikipedia having supervised labels judging whether the two sentences are good paraphrases of each others or not. Similarly, PAWS-QQP contains contains pairs of Quora questions with similar paraphrase-worthiness labels. PAWS[4] stands for 'Paraphrase Adversaries from Word Scrambling' and Zhang et al. (2019)'s presented dataset contains over 108,463 well-formed paraphrase and non-paraphrase pairs with high lexical overlap. PAWS-Wiki and PAWS-QQP can be regarded as the gold standard for the paraphrase recognition task since all of their supervised samples are based on human judgements. For training an encoder-decoder model for paraphrasing, only the true-labelled sentence pairs from each of these datasets are considered.

3 Conclusion

In Intermediate Reports 1 and 2, we have explained our classification test-bench and the collection of data augmentation techniques that we have proposed and implemented. Our codebase is publicly available at github.com/sayarghoshroy/Augment4Gains.

In our final deliverable, we will present all of our experimental results illustrating the performance of each data augmentation technique on the downstream task of hate-speech detection.

References

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Sayar Ghosh Roy, Nikhil Pinnaparaju, Risubh Jain, Manish Gupta, and Vasudeva Varma. 2020. Summaformers @ laysumm 20, longsumm 20. *Proceedings of the First Workshop on Scholarly Document Processing*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

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.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase Adversaries from Word Scrambling. In *Proc. of NAACL*.

[4] github.com/google-research-datasets/paws