Implementing RL Based Representation Learning on Hate Speech and Abusive Detection Datasets: Comparison and/or Extension

Wei Li 300113733, Libo Long 300151908

1 INTRODUCTION

A foundation problem in many natural language process tasks is learning good representation to texts (i.e. word embedding, sentence embedding, document embeddings). In recent representation learning researches, a series of neural network methods have been proposed. The mainstream architecture can be classified roughly into 4 classes: 1) bag-of-words method which does not consider the order of words, 2) convolution network that learns representation through convolution and pooling layers, 3) recurrent network that considers the word orders and 4) structured RNN like tree-LSTM[1](Tai et al., 2015) which also considers the structure of the context from pre-defined parsing. Some tricks such as attention and transformer have also been proposed to enhance the neural model.

Tianyang Zhang and Minlie Huang proposed a novel method[2] (Zhang et al. 2018) that uses Reinforcement Learning method to explore and identify task-related structures automatically. In this project, we consider the downstream classification task in Hate Speech and Abusive Language datasets, and implement the two models proposed in the paper (a.k.a. ID-LSTM and HS-LSTM), comparing them with mainstream baseline models, and hopefully exploring the properties of the model to extend the model and increase its performance.

2 A BRIEF SUMMARY OF METHODS

2.1 Classification Task

In natural language processing, classification task is to assign tags to texts given information extracted from its content, either semantic or not. Text classification is one of the fundamental tasks in NLP and is widely used in many applications such as sentimental analysis, topic classification, spam detection, author classification, etc.

Chronologically, 3 groups of methods have been used in text classification: rule-based approaches, machine learning approaches, and deep learning approaches. Rule-based approaches use a set of rules summarized from linguistic experiences and researches. These rules are handcrafted for each specific task and dataset and required a lot of domain knowledge, which render the approach less economical comparing to latter approaches.

A more recent approach, Machine learning systems trying to extract features from the unstructured text data and apply machine learning to learn a distribute that map the feature-represent to tags. One of the most commonly used feature extraction methods is bag-of-words, which represent the text by counting the frequency of words appearing in a corpus.

Recent years, deep learning architectures achieved state of art results in text classification. In deep learning, representation of text can be learned automatically and effectively from corpora, and the result can be generalized well into different tasks.

2.2 Neural representation learning

Text classification using neural networks relies on learning vector representation of text. We give a brief summary to some of the widest used neural representation architectures.

a. CNN[3][4](Kim 2014; Kalchbrenner, Grefenstette, and Blunsom 2014;): CNN models use the same architectures as image processing tasks over texts, a potential problem of CNN for text is big 'channel' size.

b. RNN[5][6](Hochreiter and Schmidhuber 1997; Chung et al. 2014): RNN can treat inputs as a sequence, and assign weights to the previous state in a sequence, which make it a powerful method for text, string and sequential data classification. Vanilla RNN suffers from vanishing gradient problem, which is fixed to some extent by gated models like LSTM or GRU.

c. RCNN[7](R. Girshick et al., 2014): The idea of RCNN is combine the advantage of RNN and CNN by capturing contextual information with the recurrent structure and constructing the text representation with CNN.

d. Hierarchical Attention Networks[8](Yang et al. 2016): this method has a hierarchical structure that mirrors the hierarchical structure of documents; The member of the hierarchy can be words, sentences and documents. Moreover, each hierarchy has its attention.

2.3 ID-LSTM and HS-LSTM

ID-LSTM and HS-LSTM are two reinforcement learning based representation learning models. Unlike the architecture mentioned above, the two models can use the information of text structure (POS, for example). While other structured representation models like tree-RNN used pre-specified parsing trees, ID-LSTM and HS-LSTM use reinforcement learning to learning the structure from data.

3 PROJECT DESCRIPTION

3.1 Hate Speech Detection

Hate speech detection is an application of text classification and is deployed on many social media platforms. The objective of hate speech detection is identifying abusive languages that target specific individuals or groups. The kind of hate speech can be either about race, gender, sexuality cyberbully. One observation is made in this area that there exist some bias to certain groups of user, and thus models tend to generate false-positive results[9](Davidson et al., 2019). For example, it is innocuous in the context of the homosexual community to assert "I am a gay man", but when the statement is evaluated in a big data set together with comments from other groups, it gets high toxicity score.

3.2 Data Set

In the project, we focus on Twitter data. Twitter comments are short and are rich in hate speeches. Now, we identified 5 different hate speech datasets, all are labeled by humans. Some of the data only distinguish whether a comment contains hate or offensive content, while others specify the kind of offense like racism or sexism. The data sets are:

- Waseem and Hovy (2016): 130k tweets, 3 classes (racism, sexism, neither)
- Waseem (2016): 7k tweets, 4 classes (racism, sexism, both, neither)
- Davidson et al. (2017): 24k tweets, 3 classes (hate, offensive, neither)
- Golbeck et al. (2017): 20k tweets, 2 classes (harass, non)
- Founta et al. (2018): 92k tweets, 4 classes (hate, abusive, spam, neither)

Currently, we have not decided which data is to be used in our project. Given the limit of time, we want to focus on one data set and train models that obtain fine-grained results instead of comparing models that trained roughly in different data sets.

3.3 Methodology

1) Preprocessing

- Word embedding: Using pre-trained word embedding in twitter dataset(GloVe, Word2Vec, etc). Ideally, good

pre-trained word embedding in Twitter corpus can take good care of slangs, abbreviations and emojis that is rarely occurred in formal context. We want to fix the choice of word embedding to one.

- Sentence preprocessing: We tokenize sentences and eliminate unnecessary tokens. Then we padding all sentence to a fix appropriate length while eliminating data that exceed that length.

2) Models: In this phase, we consider the following model architectures: CNN, LSTM, Bi-LSTM, RCNN, Hierarchical Attention Networks, RL-based(ID-LSTM, HS-LSTM)

3) Evaluation: In this text classification problem, we consider three evaluation metrics: Accuracy: the performance of each model in general sense Precision: taking the false positive problem in hate speech detection task into consideration F1 score: a balanced evaluation that takes both bias and performance into account The number of classes in datasets listed above range from 2 classes to 4 classes. When the number of classes exceeds 2, we calculate scores for each of the classes.

3.4 Extent the State of Art

We hope to understand and explore the underlining structure of reinforcement learning through the experiment and extend its performance. We also want to find out whether RL-based models' ability to explore sentence structure unsupervisedly can attend to the bias problem in hate speech detection problem.

REFERENCES

1. Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks, 2015.
2. Tianyang Zhang, Minlie Huang, and Li Zhao. Learning structured representation for text classification via reinforcement learning, 2018.
3. Yoon Kim. Convolutional neural networks for sentence classification. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL, 2014.
4. Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 655–665. The Association for Computer Linguistics, 2014.
5. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
6. Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.
7. Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 580–587. IEEE Computer Society, 2014.
8. Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. Hierarchical attention networks for document classification. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1480–1489. The Association for Computational Linguistics, 2016.
9. Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. Racial bias in hate speech and abusive language detection datasets. *CoRR*, abs/1905.12516, 2019.