Meiying Chen, Xusheng Ji
CSC 440: Data Mining
November 05, 2019
Course Project Proposal

# Toxic Comments Challenge using Attention-based Bi-LSTM

## Background and Rationale

Wikipedia is an open-access discussion website where editors can assist to compose much better article by discussion and the website was made available to the public for the first time on Jan 2001. Because people are free to comment and discuss article in the wikipedia, the quality of these responses is mixed with some are identified as “toxic”. A toxic comment is one that intended to make an obscene, threat or insult statements rather than a really helpful comment.

As Wikipedia grows, it becomes increasingly important to accurately identify and remove toxic answers and to prevent toxic comments from spreading. Reducing the negative impact is also an important part of social network public opinion research. In this project, several different approaches will be applied and compared to predict if a comment is toxic or not. The result will help online communities quickly remove negative content and deter potential posters.

The classical method on this topic is applying machine learning algorisms. Machine learning is a study of statistical models that computer systems use to make predictions or other tasks without using predefined equations, using learning and inference instead. Traditional machine learning methods like SVM, has achieved good accuracy on this problem. However, because traditional methods rely on the feeding of handcraft attributes, there are great limitations as they are neither able to generate multi-level features nor take context information into account.

Neural networks have emerged the recent years and had groundbreaking improvements in natural language processing domain. Numerous new architectures have been proposed and specially adapted to NLP like CNNs and LSTMs. LSTM is a recurring deep learning architecture that uses memory cell vectors and a set of element multiplication gates to control the way information is stored, forgotten, and utilized in the network[1]. So, it can handle context information, which is considered to be useful in our project.

LSTM employs the previous information of each word, or furthermore, Bidirectional-LSTM handles former and later locations. Nevertheless, this kind of memory information exists in the form of hidden layer and neuron weights, which is not sufficient for predicting, as what is the most relevant information that passed between the past and future remains unknown.

Attention models fixed this problem and avoid emphasis too much on the data point being close to one another by allowing the model to automatically search for parts of the source that are

relevant to predicting the target, without fixing the length of the vectors used[2][3][4]. Therefore, we finally propose a kind attention-based LSTM hoping to achieve greater performance.

## Research Questions

In this project, we will develop models that identify and flag insincere questions. Research contents include:

1. Word embedding method, e.g. GloVe[6], Bert[5], Word2Vec
2. Multi-class text classification algorithms
3. Imbalance data set handling [7].
4. Multi-labeled data processing.
5. Out of vocabulary problem(OOV).

## Research Methodology

- **Data Analysis:**

  We need first to analyze the whole dataset before data cleaning. There are couple of perspective we need to analyze:

  - **Data Distribution:** We need to confirm whether dataset is an imbalanced distribution, if it is an imbalanced dataset then we need to take some measures to solve it such as SMOTE, under-sampling or over-sampling.
  - **Multi-labeled data:** If one comment has been labeled for more than one classification, then we need to analyze the correlation between the different labels. Then the strategy can be divided into three different types:
    - First-order Strategy: Considering that the labels are independent of each other, the multi-label problem can be transformed into a normal classification problem. We can use Binary relevance to solve this problem [8].
    - Second-order Strategy: In this category, pairwise correlation between labels is taken into consideration, which results in a significant increase in computational complexity. For this category, we can use Calibrated Label Ranking to solve the problem [9].
    - This is to consider the correlation between multiple labels, the computational complexity will be much higher than second order. Random K-labelsets would be useful for this strategy [10].
  - **Unknown Words:** The most challenge problem in NLP is the out of vocabulary problem(OOV). For example, some non-English words would happen in our comments content such as Japanese or Chinese and all these words cannot be recognized by the word embedding matrix. We can use genism to train fasttext[16] to solve this problem. Furthermore, we may need to consider a new method which can replace the unknown word with the most similarity word to solve OOV problem.

- **Data Clean:**

  In terms of the data preprocessing, we only need to process the comment_text attribute. The processing methodology includes the following procedures:

  1. Change all the Numbers 0-9 into English zero-nine.
  2. For lower case and upper case, then we need to transform all the words into lower case.
  3. Get rid of the stop words such as the, an, that and so on. Because these kinds of stop words have litter effect on the classification results.
  4. Find some regular expression code to replace the actual date data with the String "date", the phone number with the string "phone number", and some website links with the word "url".
  5. Extract the word stem such as went and go. Essentially, these two different words have the same meaning so it is important for us to get the word stem.
  6. Use textblob to correct some of wrong words in order to make them useful for pre-training word vectors.

- **Feature Engineering:**

  This part should be divided into two different parts which including tokenization and word embedding.

  1. **Tokenizaition:** Text and sequence preprocessing libraries from keras library can be used here. Tokenizer is a class for vectorized text that converts text into sequences and pad_sequences are used for sequence filling.
  2. **Word Embedding:** In this part, we will try three different kinds of word vectors which is glove, Bert and word2vec.

- **Modeling :**

  In this part, we are going to use five different kinds of model to finish the classification problem which is NB-SVM, Elman-RNN, Jordan-RNN, LSTM and attention-based LSTM. Before we

  - **NB-SVM:** The NB-SVM is a combination algorithm of Naïve bayes and Support Vector Machine [11]. This algorithm works well in most of the sentiment analysis problem and so it should be an excellent baseline.
  - **Elman-RNN:** Elman-RNN is known as a simple RNN and it use the hidden layer to be the input for the next time step [12]. RNN will be able to search forward through the long memory for the key information in the input then we also need to use Elman-RNN to be one of our models.
  - **LSTM:** LSTM is a kind of transformation for simple RNN and it can help solve the gradient vanish problem in RNN [13]. Then it can provide some improvement for the model accuracy in our problem.
  - **Bi-LSTM +Attention:** Regardless of length, input sequences are encoded into a fixed-length vector representation, whereas decoding is limited to that fixed-length vector representation. This problem limits the performance of the model, especially when the input sequence is too long[14].The basic idea of Attention mechanism is that it

breaks the limitation of traditional encoder - decoder structure which relies on a fixed length vector. Furthermore, attention-based LSTM work pretty well on aspect-level sentimental analysis [15] then we need to use this model to be our best one. Figure 1.1 shows the architecture for attention-based LSTM.

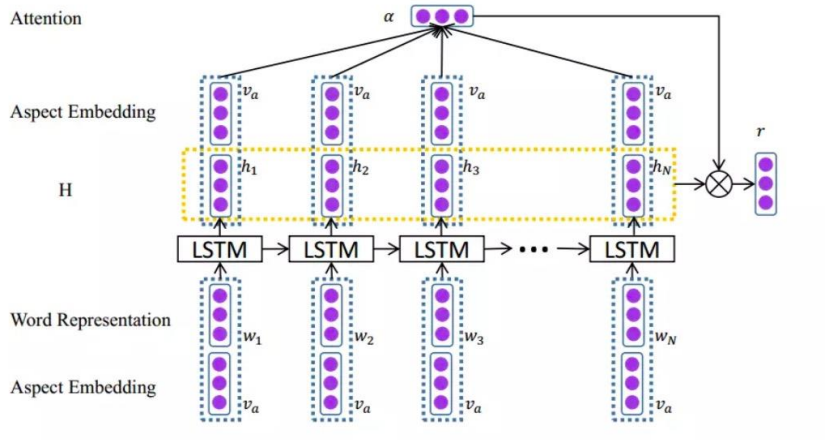


Figure 1.1 The attention-based LSTM used in Sentiment Analysis

## Dataset Acquirement

Kaggle completion for toxic comments question classification provide us archives files we need. The materials include training set and test set in csv format. The training set has been labeled so there is no need for us to label the data manually.

The training set has 159586 records and each of them has 8 different attributes which is ID, comment_text, toxic, severe_toxic, obscene, threat, insult and identity_hate. Test set only have two attributes which is ID and comment_text.

Furthermore, we can also get some embedding vector files from the google research or any other website such as glove vector, google news vector, paragram vector and wiki news vector with 300 dimensions.

Finally, we are trying to use these three different kinds of pre-training word vectors for each of our model to evaluate the performance.

## Model Evaluation

As this is a classical classification problem, so we need to evaluate the performance from the following different aspects:

- Accuracy: For the accuracy, we plan to use confusion matrix, F1 score and ROC-AUC to evaluate each of the model. Remember that we need to apply three different kinds of word vectors to each of the model, so we need to compare 15 different cases. Furthermore, machine learning always need the cross validation to solve over fitting.

However, in most of cases of deep learning, we are trying to avoid this because the data amount is too huge. What we are trying to do is to do hold-out a portion of dataset for testing.

- Speed: In this part, we are trying to consider two different aspects which is the time to construct the model and the time to use the model.
- Consumption: In most of cases, the algorithm always need to consider a trade-off between performance and consumption. In our paper, we also need take the GPU consumption into account in our evaluation procedure.

## Plan of Work and Timeline Schedule

We are trying to finish the project within four weeks and our timeline should be showed in weeks.

- First Week (2019-11-11 to 2019-11-17): Finish the data preprocessing including data analysis and data cleaning.
- Second Week (2019-11-18 to 2019-11-24): Implement NB-SVM, Elman-RNN,LSTM and attention-based LSTM. Then we need to construct these models on the training set and use these models on the test set to evaluate the model.
- Third Week (2019-11-25 to 2019-12-01): Parameter tuning and evaluation results analysis.
- Fourth Week (2019-12-01 to 2019-12-08): Presentation and report writing.

## Reference


[1] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. arXiv preprint arXiv:1601.06733, 2016.

[2] Ashish Vaswani, Noam Shazeer, etc. Attention is all you need. arXiv preprint arXiv:1706.03762, 2017.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, etc. BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[4] Yinhan Liu, Myle Ott, Naman Goyal, etc. RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692, 2019.

[5] John Wieting and Mohit Bansal and Kevin Gimpel and Karen Livescu and Dan Roth, From Paraphrase Database to Compositional Paraphrase Model and Back, TACL, 2015

[6] Y. Sharma, G. Agrawal, P. Jain and T. Kumar, "Vector representation of words for sentiment analysis using GloVe," 2017 International Conference on Intelligent Communication and Computational Techniques (ICCT), Jaipur, 2017, pp. 279-284.

[7] G. Y. Wong, F. H. F. Leung and Sai-Ho Ling, "A novel evolutionary preprocessing method based on over-sampling and under-sampling for imbalanced datasets," IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society, Vienna, 2013, pp. 2354-2359.

[8] Tsoumakas, G., Dimou, A., Spyromitros, E., Mezaris, V., Kompatsiaris, I. and Vlahavas, I., 2009, September. Correlation-based pruning of stacked binary relevance models for multi-label learning. In Proceedings of the 1st international workshop on learning from multi-label data (pp. 101-116).

[9] Fürnkranz, J., Hüllermeier, E., Mencía, E.L. and Brinker, K., 2008. Multilabel classification via calibrated label ranking. Machine learning, 73(2), pp.133-153.

[10] Tsoumakas, G. and Vlahavas, I., 2007, September. Random k-labelsets: An ensemble method for multilabel classification. In European conference on machine learning (pp. 406-417). Springer, Berlin, Heidelberg.

[11] Wang, S. and Manning, C.D., 2012, July. Baselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2 (pp. 90-94). Association for Computational Linguistics.

[12] Chandra, R. and Zhang, M., 2012. Cooperative coevolution of Elman recurrent neural networks for chaotic time series prediction. Neurocomputing, 86, pp.116-123.

[13] Sundermeyer, M., Schlüter, R. and Ney, H., 2012. LSTM neural networks for language modeling. In Thirteenth annual conference of the international speech communication association.

[14] Bahdanau, D., Cho, K. and Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

[15] Wang, Y., Huang, M. and Zhao, L., 2016, November. Attention-based LSTM for aspect-level sentiment classification. In Proceedings of the 2016 conference on empirical methods in natural language processing (pp. 606-615).

[16] Khokhlov, Y., Tomashenko, N., Medennikov, I. and Romanenko, A., 2017. Fast and accurate OOV decoder on high-level features. arXiv preprint arXiv:1707.06195.