Quora insincere question classification using deep neural nets

Le Thanh Tung[a] and Nguyen Trung Hieu[b]

[a]University of Engineering and Technology
[b]Hanoi University of Science and Technology

ABSTRACT

In this report, we present an approach to address the insincere question detection problem using the advances of neural networks architecture with several current state-of-the-art word embeddings.

Keywords: LSTM, NLP, Sentiment Analysis, Elmo, Glove

1. MEMBERS' CONTRIBUTION

Le Thanh Tung:

- preprocessing data
- implementing neural nets with Keras embeddings and Glove embeddings
- making slides

Nguyen Trung Hieu:

- implementing neural nets with GoogleNews embeddings and Elmo embeddings
- writing report

Shared works: debugging, modifying models.

2. PROBLEM STATEMENT

A question is classified as insincere (positive label) if it intended to make statement rather than look for helpful answers * The task is to decide whether a question is insincere or not. Some characteristics that can signify that a question is insincere:

- **Has a non-neutral tone:**
  Has an exaggerated tone to underscore a point about a group of people
  Is rhetorical and meant to imply a statement about a group of people
- **Is disparaging or inflammatory:**
  Suggests a discriminatory idea against a protected class of people, or seeks confirmation of a stereotype
  Makes disparaging attacks/insults against a specific person or group of people// Based on an outlandish premise about a group of people
  Disparages against a characteristic that is not fixable and not measurable
- **Isn't grounded in reality:**
  Based on false information, or contains absurd assumptions
  Uses sexual content (incest, bestiality, pedophilia) for shock value, and not to seek genuine answers

*https://www.kaggle.com/c/quora-insincere-questions-classification/data

3. DATA

The data set from Kaggle contains 1306122 Quora questions. All the questions are labeled (annotate 1 for insincere one and 0 for the other). The data set is highly imbalanced since only 80810 questions is labeled as 1.

4. MODEL ARCHITECTURE

In this section, we propose a simple deep learning model which is able to understand semantic meanings of the texts, so that

We propose a deep neural a deep neural network architecture that integrates word-to-vec embeddings and convolution layers in combination with long short-term memory (LSTM) layer, so that it is capable of capturing the meaning of the question from dense representations of words. The networks consists of embedding block (a pre-trained architecture), then a convolution layer, a max pooling layer, a LSTM layer, followed by an output layer.

4.1 Embedding block

We tried four way to build embedding block : Keras embedding layer, Glove, GoogleNews and Elmo.

4.1.1 Keras embedding layer:

This approach basically is to learn the word embedding of the vocabulary built from the data. The Keras embedding layer maps each index of word in the vocabulary to a dense vector, which is to be learned in the learning process. This embedding requires low time complexity cost.

4.1.2 Glove embedding

In this approach we use a pre-train embedding weights Glove proposed by Jeffrey Pennington, et al. [1]. The Glove model is specially designed to produce linear directions of meaning by making use statistical features based on word-word co-occurence in the corpus. The Glove embeddings archieved remarkable result on several word similarity tasks and on the named-entity recognition bechmark, accourding to the paper.

4.1.3 Google News embedding

This is a set of embedding trained on part of Google News dataset of about 100 billiion word. the embedding contains 300-dimensional vector representations for 3 million words and phrases. [†] The model used for training is followed from the works of Mokolov, et al. [2]. The advangate of this advantage of such embedding is its wide range vocabulary, reducing the rate of out-of-vocab words in the training set.

4.1.4 Elmo

Elmo (embedding from language model) is one of recent embeddings which achieves state-of-the-art result in various NLP tasks [3] The unique characteristics of Elmo is that the word representation is a function of the whole sentence containing the word. In other word, a word in different contexts would have different representations, this innovation also overcomes the problem of mis-represent of words which have the same spelling but distinct meanings. Elmo is a deep representation of word as its combines vectors from not only the last layer but several layers of the models, allowing for very rich word representation.

4.2 Convolution and max-pooling layer

We use 1-dimension convolution layer in order to capture the meaning of several nearby words as a phrase, since words are usually placed in text to form larger components, namely noun phrases, verb phrases, which have more clear, complete meanings than single words. The convolution layer is then followed by a maxpooling layer with the main function is to down-sample the data, reducing the computation cost while still preserving the main information.

[†]https://code.google.com/archive/p/word2vec/

4.3 Long short-term memory layer

LSTM layer is a special variation of recurrent neural networks (RNN), a type of deep learning architectures specialized for capturing information of sequential data. One disadvantage of vanilla RNNs is the instability of gradients (the gradients is explode or vanish, mostly vanish during the training). Long short-term memory (LSTM), an upgraded architecture over conventional RNNs proposed by Sepp Hochreiter and Jrgen Schmidhuber [4], addresses this problem and thus allows the model to eectively capture long-term dependencies. The core idea of LSTM architecture is to balance current and past information by using weights, which are updated during the training process. As a result, LSTM layer is a good choice to encapsulate meaning of the whole question in a more compact form as a vector.

4.4 Output layer

The output of LSTM is linearly combined, then applied sigmoid function to estimate the probability that the question is insincere.

4.5 Model Regularizations

To avoid overfitting we use dropout technique [5], which randomly chooses 1 small portion weights not to update.

5. EMPIRICAL EVALUATION

5.0.1 Model training process

With the given data set, we measure the f1 score on the validation data during the training process in order to avoid overfitting. The networks is trained using Adaptive Moment Estimation (Adam) optimizer [6], which is able to accelerate the convergence and hence reduce the number of training epochs required to reach an optimum.

The number of epochs chosen for each implementations varies due to the complexity of the pre-trained embeddings model and the dimension of the embedding space.

5.1 Results

In order to evaluate the effectiveness of the model we use the f1 score as performance metric, which is harmonic average of the precision (the actual positives in all instances predicted as positive) and recall (the proportion of actual positives that are correctly identified in all positives). F1 score is chosen as the data set is highly imbalance, the common use metrics, such as accuracy would be inappropriate. Based on the given metric, the performance of the model is measured during the training process and also in a independent test set.

The results of the model with four embedings in the test set:

Word representation	F1-score
Keras word embeddings	60.12% (threshold 0.33)
GloVe	66.13% (threshold 0.33)
GoogleNews	64.87% (threshold 0.49)
ELMo	64.26% (threshold 0.25)

6. CONCLUSION AND FUTURE WORKS

6.1 Conclusion

In the project, we have learned the mechanism of vector embeddings for words in NLP tasks, as well as the recent techniques to learn such embeddings. In the field of text sentiment analysis, we have acknowledged the effectiveness of recurrent neural nets in capturing information of sequential data. The performance of the model with four embeddings is quite promising. The result of Elmo is expected to be the best but with the complexity of such embedding the model was not trained with enough iterations.

6.2 Scope for future developments

The first direction is to apply the model with the latest language-understanding model – XLnet – which outperforms previous model (Elmo) on a various NLP tasks, including sentiment analysis [7]. Another possible way to improve the results is to take into account some statistical features of the texts such as punctuation, special characters, as well as emoji, which may be helpful the task. Additionally, it is worth trying ensemble methods to leverage the advantages of all models by linearly combining the softmax layer or the embedding.

ACKNOWLEDGMENTS

We would like to extend our gratitude to VEF for organizing the course and the project in such a professional manner.

We would like to thank instructor Ha Q. Nguyen, coordinator Duong Nguyen and teaching assistants for their expertise and efforts in providing us knowledges and hands-on experience in machine learning.

References

[1] Pennington, J., Socher, R., and Manning, C., "Glove: Global vectors for word representation," in [*Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*], 1532–1543 (2014).

[2] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J., "Distributed representations of words and phrases and their compositionality," in [*Advances in neural information processing systems*], 3111–3119 (2013).

[3] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L., "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365* (2018).

[4] Hochreiter, S. and Schmidhuber, J., "Long short-term memory," *Neural computation* **9**(8), 1735–1780 (1997).

[5] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research* **15**(1), 1929–1958 (2014).

[6] Kingma, D. P. and Ba, J., "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980* (2014).

[7] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V., "Xlnet: Generalized autoregressive pretraining for language understanding," *arXiv preprint arXiv:1906.08237* (2019).