Quora Insincere Question Classification

Detect Toxic content to improve online conversations

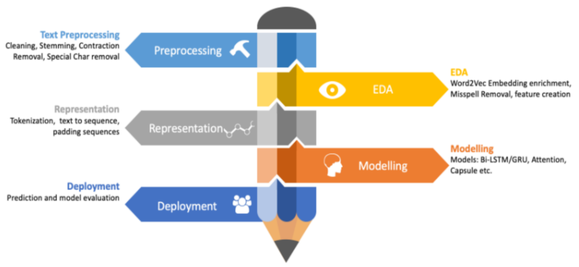
Bharadwaj Aryasomayajula, Akhil Morampudi, Mahanth Mukesh Dadisetty
Department of Computer Science
University of North Carolina at Charlotte

1. Abstract

An existential problem for major social media website today is how to handle toxic, irrelevant and divisive content. There are many content based websites like Quora, Reddit, Twitter, StackOverflow, Facebook etc. So we are using Quora Dataset to tackle this problem to keep their platform a place where users can feel safe sharing their knowledge with the world. Quora is a platform that empowers people to learn from each other. On Quora, people can ask questions and connect with others who contribute unique insights and quality answers. A key challenge is to classify out insincere, toxic and misleading questions those founded upon false premises, or that intend to make a statement rather than look for helpful answers. In this project we will be predicting whether a question asked on Quora is sincere or not. In this undertaking, we will execute this by utilizing Natural Language Processing strategy called Word Embedding. This, undertaking the feature engineering and distinctive methods of word embedding and Exploratory Data Analysis. We begin with two baseline models, logistic regression and naive Bayes, and then create a more advanced recurrent neural network model and use different pre-trained word embeddings to achieve the best result.

2. Dataset and Flow Diagram

We have used Quora Dataset from kaggle in our experiments. Data set contains train.csv, test.csv, sample submission's and different word embedding. The training data includes the question id, question that was asked, and whether it was identified as insincere or sincere. There are 80,810 questions are insincere and other 12,25,312 question are sincere. So about 6% the training data are insincere questions (target=1)and rest of them are sincere. Below diagram depicts the flow diagram of our projects



3. Data Preprocessing

Data Preprocessing is the important phase in Machine Learning, where we analyze, identify and correct the messy, raw data. The real world is full of painful and noisy datasets. Our first step should be cleaning the data in order to obtain better features.

- Transform our text into lower case
- Remove Punctuation (Doesn't add any value)
- Removing Stop words or Commonly occurring words

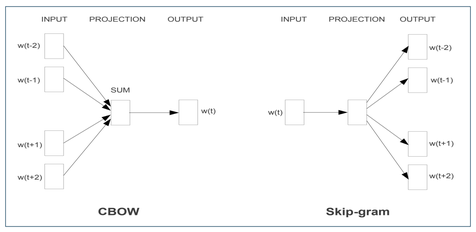
4. Text Preprocessing

Tokenizing:

Tokenization is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. Tokens can be individual words, phrases or even whole sentences. Tokenization is a necessary first step in many natural language processing tasks, such as word counting, parsing, spell checking, corpus generation, and statistical analysis of text. Tokenizer is a compact Python module for tokenizing, which converts Python text strings to streams of token objects, where each token object is a separate word, punctuation sign, etc. It also segments the token stream into sentences, considering corner cases such as abbreviations and dates in the middle of sentences. NLTK provides a number of tokenizers in the tokenize module

Word Embeddings

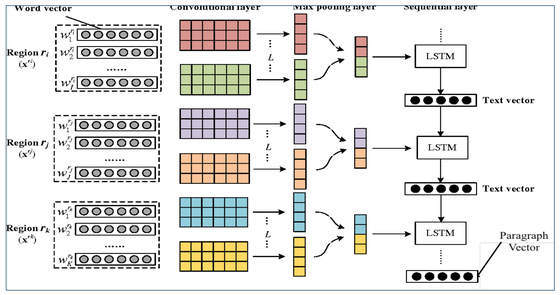
Word embeddings are a type of word representation that allows words with similar meaning to have a similar representation. They are a distributed representation for text that is perhaps one of the key breakthroughs for the impressive performance of deep learning methods on challenging natural language processing problems.



Glove

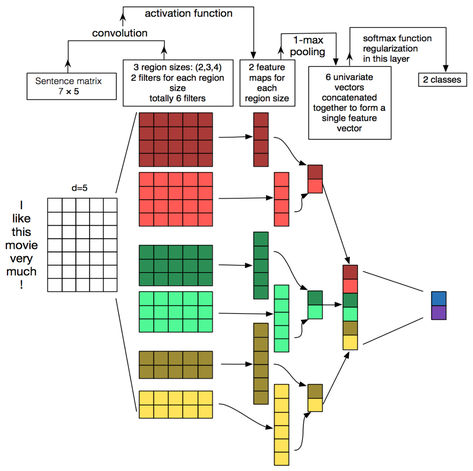
The Global Vectors for Word Representation, or GloVe, algorithm is an extension to the word2vec method for efficiently learning word vectors, developed by Pennington, et al. at Stanford.

Classical vector space model representations of words were developed using matrix factorization techniques such as Latent Semantic Analysis (LSA) that do a good job of using global text statistics but are not as good as the learned methods like word2vec at capturing meaning and demonstrating it on tasks like calculating analogies. GloVe is an approach to marry both the global statistics of matrix factorization techniques like LSA with the local context-based learning in word2vec.



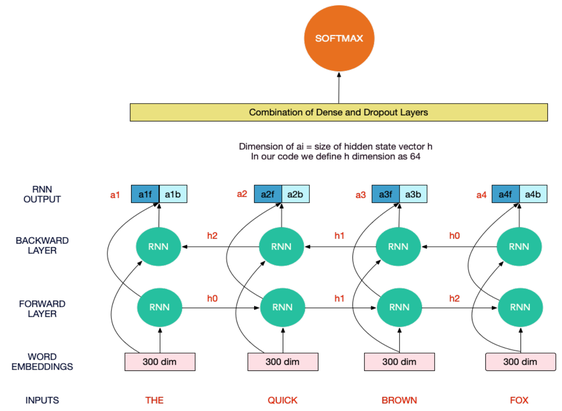
4. Solution

CNN on text data through a diagram. The result of each convolution will fire when a special pattern is detected. By varying the size of the kernels and concatenating their outputs, you're allowing yourself to detect patterns of multiples sizes (2, 3, or 5 adjacent words).Patterns could be expressions (word ngrams?) like "I hate", "very good" and therefore CNNs can identify them in the sentence regardless of their position.

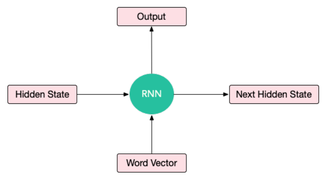


Bi-directional RNN(LSTM/GRU):

Text CNN works well for Text Classification. It takes care of words in close range. It can see "new york" together. However, it still can't take care of all the context provided in a particular text sequence. It still does not learn the sequential structure of the data, where every word is dependent on the previous word. Or a word in the previous sentence. RNN help us with that. They can remember previous information using hidden states and connect it to the current task.



Long Short-Term Memory networks (LSTM) are a subclass of RNN, specialized in remembering information for an extended period. Moreover, the Bidirectional LSTM keeps the contextual information in both directions which is pretty useful in text classification task



6. References

[1] C. Liu, Y. Sheng, Z. Wei, and Y. Yang, "Research of text classification based on improved tf-idf algorithm," in 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE), Aug 2018, pp. 218–222.

[2] Boiy, Erik and Marie – Francine Moens. "A Machine Learning approach to Centiment analysis in multilingual web Texts" Information retrieval 12.5 (2009): 526-558.

7. Acknowledgements

Dr Minwoo Jake Lee
Department of Computer Science
University of North Carolina at Charlotte