VADER

CNN

pre-trained BERT

In order to analyze text, we first need to convert words in to numerical representations.

VADER:

Tokenization method for VADER:

VADER itself includes 7500 words with polarity (positive/negative) with intensity scale from -4 to +4. Therefore, the word representation for it is unidimensional.

For example, the word “okay” has a positive valence of 0.9, “good” is 1.9, and “great” is 3.1, whereas “horrible” is – 2.5, the frowning emoticon “:(“ is – 2.2, and "sucks" and its social media slang “sux” are both – 1.5.

CNN:

Tokenization method for neural networks:

In order to have a more comprehensive understanding of the words, such as part of speech, meaning other than its polarity, we need a more complicated way to represent words. Word2vec is a popular way to convert words into numbers. Word2vec converts each unique word into 300 dimensional vectors and the vectors contains meaning.

Ideally, people can draw analogy from them. From example, [king] - [man] + [woman] = [queen]

And word2vec is trained based on some given text corpus.

Initially, researchers used one-hot encoding for words. One-hot encoding uses a vector with “1” in one dimension and “0” in the rest of the dimension to unique words. This way the neural network is then able to tell the difference between each word.

BERT: Bidirectional Encoder Representations from Transformers

BPE Tokenization method for BERT:

byte pair encoding: essentially, BPE is a sub-word level tokenization method that enables us to split prefix and suffix of words into unique token and therefore reduce the number of different words in text.

For example: Word2vec gives similar but numerically different vectors for “loves”, “loving”, and “loved”. However, BPE can tokenize these words into [lov], [ed], [ing], [es], which is more intuitive for neural networks to understand.

Contextual embedding: another advantage of tokenization method for BERT is that it have contextual embedding. For example, the word “bank” would have the same context-free representation in “bank account” and “bank of the river.” Contextual models instead generate a representation of each word that is based on the other words in the sentence. For example, in the sentence “I accessed the bank account,” a unidirectional contextual model would represent “bank” based on “I accessed the” but not “account.” However, BERT represents “bank” using both its previous and next context — “I accessed the ... account”.

Transformer architecture:

[insert photos]

Parallelizable

Visualized attention

long range dependency

The advantage of transformer over traditional LSTM:

Data processing procedure:

BERT max input length=> 512 byte pair

Since earnings call for one company have multiple answers, and each answer can vary greatly in length, I concatenated all answers in one single transcript and split them into a paragraph of 400 words. 400 words using BPE can be tokenized into length of 480 to 512 different tokens. Transcript for one company can usually be converted to 5 - 15 paragraphs.

I cla

Architecture:

Training:

|  |  |  |
| --- | --- | --- |
| binary classification task | data | label |
| Un-processed data | plain text | 3 month industry relative return after earnings call grouped into 10 bins |
| processed data | n\* 400 words paragraphs | 10 bins grouped into 2 bins for binary classification |
| trainable data | n\* 512 length BPE indices | n\* binary label (broadcast) |

1. training examples and labels: Binary classification task-> 3 month industry relative return after earnings call grouped into 10 bins and 2

Result:

We train the epoch for 2 epochs based on the previous year’s data and