**AMAZON REVIEWS CLASSIFICATION**

**Amazon Review Classification**

CS584 Homework 1

Edvin Beqari and Doug Williamson

GMU  September 21, 2020

Natural Language Processing (NLP) is an application of statistics and machine learning techniques to construct systems that interpret human language. The goal of NLP is to perform functions such as optical character recognition, machine translation, sentiment analysis, and many more. The field is rich with algorithms, but almost all approaches involve a first important step of representing the input, whether it is a single word, paragraph, book, etc., as n-dimensional numerical vectors called “word embeddings.’ In particular, for sentiment analysis considered in this assignment, the vector representation of word paragraphs can be compared using a loss function such as cosine similarity to determine their semantic similarity. Consequently, a simple algorithm such as Nearest Neighbor Classification can be applied to store the similarity measures and make a classification based on the top scores.

**PV-DBOW with interleaved Skip-Gram word embeddings**

Doc2Vec is a language processing algorithm which represents documents or paragraphs as vectors. The algorithm uses unsupervised techniques namely Skip-Gram and Bag of Words (BOW), and other helper algorithms to achieve a vector representation of a given corpus. **Skip-Gram** is a shallow neural network *to train words* where its input is a *target word* and its output are the words in its window (words to the left and right) formally called the *contexts*. Each training sample is also annotated as contextually positive or negative samples. Hence, the data and positive label pairs are of the form **[(*target*, *context*), 1], and target and negative labels are annotated [(*target*, *random*-*words*), 0]**.Following, due to the size of typical corpus, a *doc2vec* network will have millions of weights which makes the training very challenging. **Negative Sampling**4 is a helper algorithm which significantly reduces the number of weights updated during training. This is achieved by selecting only a subset of negative samples and updating a portion of the network weights while still updating all weights for each positive training sample. Lastly, the doc2vec training algorithm *to train paragraphs* selected for this assignment is **Distributed Bag of Words**5 (PV-DBOW). In theory, a DBOW model works opposite of the Skip-Gram, and instead it attempts to predict a word from a target vector (paragraph in this case). The Gensim6 library allows training of the paragraph and word vectors simultaneously and both can be used to predict words. For instance, in PV-DBOW with interleaved Skip-Gram word embeddings the algorithm attempts to predict a word with the DBOW model from a paragraph vector, then words within the specified context window are used to predict the context paragraph. This specific setup was found to work best with the provided dataset.

**BERT with SentenceTransformers**

To perform the encoding of the product reviews within our model, we utilized the Bidirectional Encoder Representations from Sentence Transformers7 (BERT) tool which was first released by Google in 2018. It is a pre-trained model which utilizes forward and backward analysis (hence, bidirectional) to analyze the context of each word in the input document. BERT comes in two versions: BERTBASE which is pre-trained with 800M words from the BooksCorpus andBERTLARGE which is pre-trained with 2,500M words from the English Wikipedia1. We utilized BERTBASE in our initial run and BERTLARGE in our subsequent run to evaluate the difference in performance. As expected, BERTLARGE produced better results.

BERT is based on the Transformer concept of Natural Language Processing wherein the system processes entire sentences as opposed to single words. This allows the model to understand the context of the words within the sentence, providing a much higher degree of understanding. During the pre-training process, BERT conducts two exercises simultaneously: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). In MLM processing, random words are replaced by tokens (“masks”) and the model must predict the correct missing word. For NSP, the model is given two sentences and the system must determine if the second sentence logically follows the first sentence. Both of these processes require an understanding of the context of the words and sentences to make accurate predictions.2 Inside the BERT transformer model are multiple encoder layers (12 in Base, 24 in Large), a large feedforward neural network (768 hidden units in Base, 1024 in large) and multiple attention heads (12 in Base, 16 in Large). Training models on large data sets, such as the BooksCorpus used to pre-train the Base model, requires considerable computational time, but as a pre-trained model, processing time is very efficient. We were able to utilize both BERT models on standard laptop computers and the results were produced in minutes.2

**Results and Discussion**

The BERT Transformer models are the successors of Long Short-Term Memory (LSTM) architecture and the state-of-the-art technology in NLP. As mentioned earlier, their outstanding performance is also related to the vast amount of data these models are trained on. Table 1 below captures the results of the 5-fold cross validation on the training data, the accuracy score on the test set, and the board rank as of submission on September 21, 2020 at 07:19 AM with registered team name as “beqarie.”

**Table 1:** Best performing model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **KNN Coef. (k)** | **Best CV Score %** | **Test Score %** | **Board Rank** |
| **bert-large-nli-mean-tokens** | 56 | 86.2 | 87% | 1 |

Sentence embedding is made very easy with the transformer models. They are trained on full sentences including common stop-words; therefore, it is not necessary to preprocess the data. On the contrary, discussions with other teams leads us to believe that cleaning the data may harm the prediction accuracy. Nevertheless, working with this large model presents other challenges such as sentence embedding and prediction time. Our team made use of parallelization for Pandas dataframes with pandarallel8. By utilizing all available threads in the computer, it was possible to run 5-fold cross validation for the full training set and incrementing the KNN coefficient in the range (*5*, *125*, *by=1*). The best coefficient was found to be 56. The Bert model presents other opportunities such as fine tuning with other data. According to the Bert Transformers documents, the API accepts other examples in the form:

[text = [<***Sentence A***>, <***Sentence B***>], label = <***Cosine Score***>]

An experimental attempt was made to fine tune the model for sentences that were wrongly classified in the training set by manually adjusting the cosine score compared to random paragraphs - based on their true and predicted labels. Unfortunately, the results were inconclusive as the model required training time and also due to the lack of documentation on how to properly adjust the cosine scores. More research is necessary but BERT is most definitely the best prospect for transfer learning. The full code for model finetuning is included in the source code for this assignment.

Doc2Vec is also a very interesting model that works quite well with even such a small dataset compared to larger models. As with Bert, this model treats words in their context by converting the paragraphs to vectors utilizing fewer dimensions which are easier to process, and is therefore considered superior to earlier methods such as TF-IDF. The implementing libraries like Genism offers functions to clean and tokenize the data and are rich with features to tune models for different types of datasets. The model hyperparameters that worked best for our trials were the following:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Vector Size** | **Word Model** | **Training Algorithm** | **Context Window** | **Negative Sampling** | **Epochs** |
| **Doc2Vec** | 300 | Skip-Gram | PV-DBOW | 5 | 5 | 50 |

However, there is now strong evidence that the model is overfit. The tuning problem may have been spotted earlier; however, due to chance the test set has similar labels to the training set and the model gave the impression of giving better results than expected.

In conclusion, this assignment was a great hands-on opportunity to learn natural language processing, as well as some important libraries which are heavily used in academia and industry alike. Given more time, our team would have desired to build a Convolutional Neural Network for NLP as this architecture also offers a [bountiful](https://www.thesaurus.com/browse/bountiful) array of tools to deal with complex data.

References:

1. <https://en.wikipedia.org/wiki/BERT_(language_model)>
2. <http://jalammar.github.io/illustrated-bert/>
3. <https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270>
4. <https://shuzhanfan.github.io/2018/08/understanding-word2vec-and-doc2vec/>
5. <https://cs.stanford.edu/~quocle/paragraph_vector.pdf>
6. <https://radimrehurek.com/gensim/models/doc2vec.html>
7. <https://sbert.net/docs/package_reference/SentenceTransformer.html>
8. https://github.com/nalepae/pandarallel