Sentiment analysis using BERT (pre-training language representations) and Deep Learning on Persian texts.

Soroush Karimi
Department of Computer Engineering
Iran University of Science
and Technology
soroosh_karimi97@comp.iust.ac.ir

Fatemeh Sadat Shahrabadi
Department of Computer Engineering
Iran University of Science
and Technology
f_shahrabadi@comp.iust.ac.ir

Abstract

Nowadays, due to the massive amount of text data, natural language processing has become more prevalence . Sentiment analysis is one of the text classification applications which can be used in some cases to evaluate products, make market decisions, monitor brands and prioritize customer service. sentiment analysis has taken place in English, while in other languages such as Persian, there has not been much attempts. Our goal is to evaluate attention models and BERT -a pre-trainined language representation- in the Persian language. one of the versions of BERT is Un-normalized multilingual model that contains 104 languages and Persian language is one of its languages.

1 Introduction

Sentiment analysis and opinion mining is the field of study that analyzes people's opinions, sentiments, evaluations, attitudes, and emotions from written language. It is one of the most active research areas in natural language processing and is also widely studied in data mining, Web mining, and text mining. The growing importance of sentiment analysis coincides with the growth of social media such as reviews, forum discussions, blogs, micro-blogs, Twitter, and social networks. Sentiment analysis systems are being applied in almost every business and social domain because opinions are central to almost all human activities and are key influencers of our behaviors. Our beliefs and perceptions of reality, and the choices we make, are largely conditioned on how others see and evaluate the world. For this reason, when we need to make a decision we often seek out the opinions of others. This is true not only for individuals but also for organizations. [1]

Sentiment analysis can occur at different levels: document level, sentence level or aspect/feature level. [1] [2]

Document Level Classification

In this process, sentiment is extracted from the entire review, and a whole opinion is classified based on the overall sentiment of the opinion holder. The goal is to classify a review as positive, negative, or neutral.

Sentence Level Classification

In sentence-level, the task is similar to document level, except that instead of documents, for each sentence determines whether it expresses a positive, negative or neutral opinion. Neutral means sentence expresses no opinion.

(LateX template borrowed from NeurIPS 2019)

Aspect/Feature Level Classification
In this process, the goal is to identify and extract object features that have been commented on by the opinion holder and determine whether the opinion is positive, negative, or neutral. Feature synonyms are grouped, and a feature-based summary of multiple reviews is produced.
There are still some challenges to overcome before sentiment analysis can become a more perfect tool. For example, human judgment is still far more accurate as a gauge in sentiment analysis. Automated systems cannot differentiate sarcasm from sincere text, nor can they always correctly analyze the specific contextual meaning of a word. Generally the key challenges for sentiment analysis are:

- Named Entity Recognition - What is the person actually talking about.
- Anaphora Resolution - The problem of resolving what a pronoun, or a noun phrase refers to.
- Parsing - What is the subject and object of the sentence.
- Sarcasm - Interpretation of words, especially constraints, varies from human perspective.
- Weak literature - poor spelling, poor punctuation and poor grammar

2 Related work/Background

There are a variety of methods, proposed for sentiment analysis. Some of them are based on computational linguistic but most of them are based on machine learning methods. Machine learning methods like Naive Bayes (NB), maximum entropy (ME), support vector machines (SVM) and deep learning have achieved great success in sentiment analysis. The deep learning approach, as a new field in machine learning, has attracted many researchers to employ it in different applications. In deep learning methods, representation of the words is too important. An advanced representation, encodes word similarities as a kind of distance, in a continuous highdimensional space. [3]
one of the papers that uses deep learning for sentiment analysis is "Sentiment Analysis using Deep Learning on Persian Texts" which is published in 25th Iranian Conference on Electrical Engineering (lCEE20 17). in this paper two deep neural network architectures are employed to classify Persian reviews which are about electronic products, depend on the sentiment they express. the model that proposed in this paper contains two learning phase. The first phase is learning vector representations of words using a skip-gram model in an unsupervised way and the second phase is learning document sentiments using deep neural networks in a supervised way. Deep learning models are used in this article are Bidirectional Long Short Term Memory (LSTM) and Convolutional Neural Network (CNN). [4]

3 Proposed method

One of the biggest challenges in natural language processing (NLP) is the shortage of training data. Because NLP is a diversified field with many distinct tasks, most task-specific datasets contain only a few thousand or a few hundred thousand human-labeled training examples. However, modern deep learning-based NLP models see benefits from much larger amounts of data, improving when trained on millions, or billions, of annotated training examples. To help close this gap in data, researchers have developed a variety of techniques for training general purpose language representation models using the enormous amount of unannotated text on the web (known as pre-training). The pre-trained model can then be fine-tuned on small-data NLP tasks like question answering and sentiment analysis, resulting in substantial accuracy improvements compared to training on these datasets from scratch. To do sentiment analysis, we used a pre-trained model called BERT (Bidirectional Encoder Representations from Transformers). BERT is the first deeply bidirectional, unsupervised language representation, pre-trained using only a plain text corpus (in this case, Wikipedia). [5]
BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task. Since BERT's goal is to generate a language model, only the encoder mechanism is necessary.
In data preprocessing part, we transformed our data into a format BERT understands, this format contains three parts:

1. textA is the text we want to classify, which in this case, is the Request field in our Dataframe

2. textB is used if we're training a model to understand the relationship between sentences like question answering tasks. This doesn't apply to our task, so we can leave textB blank

3. label

we did this by using the constructor provided in the BERT library. second we preprocessed our data so that it matches the data BERT was trained on and we did following steps [6]:

1. Lowercasing our text

2. Tokenizing it

3. Breaking words into WordPieces (i.e. "calling" -> ["call", "ing"])

4. Maping our words to indexes using a vocab file that BERT provides

5. Adding special "CLS" and "SEP" tokens

6. Appending "index" and "segment" tokens to each input

The collected data is unbalanced i.e. the number of positive and negative samples are unequal. We balanced train data for fine-tuning by sampling and then evaluated the test data so we get the better results. We also balance data by increasing negative documents instead of sampling positive documents.
after preparing our data, we built out model. we loaded the BERT tf hub module to extract the computation graph and fine-tuned BERT by creating a single new layer for our sentiment task. we also added a dropout layer to helps prevent overfitting and finally we converted labels into one-hot encoding. so we have a model is trained on ducoments with label negative or possitive and we can predict the new document's label as negative or possitive.
Figure 1 explaines BERT architecture and figure 2 shows our model architecture for sentiment task using BERT.

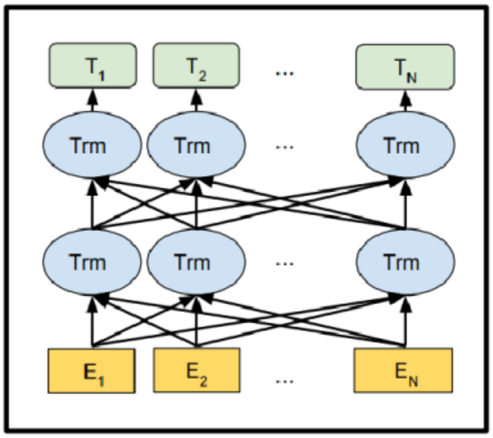

Figure 1: BERT architecture

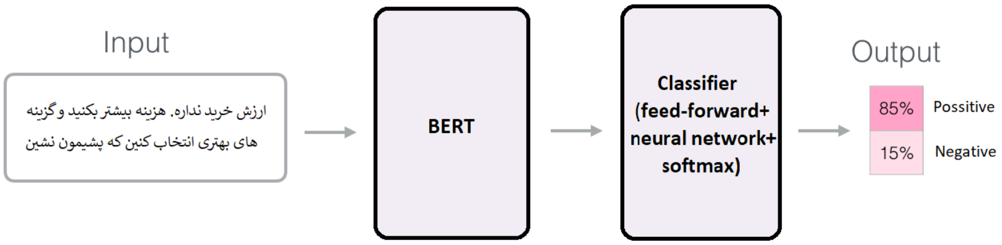

Figure 2: Our Model Architecture (BERT + our new layer for sentiment classification)

We tested different values for different parameters and finally we choosed the best. some of values for fine-tuning are:

- batch size = 32
- learning rate = 2e-5
- max seq length = 128
- num train epochs = from 3.0 to 8.0
- warmup proportion = 0.1

4 Results

We used a dataset containing a total of 200761 customer reviews about electronic products in Persian language. These reviews are gathered from the website www.digikala.com. this data set has 45839 possitive, 4196 negative and 150726 unlabeled examples. The dataset collected is unbalanced i.e. the number of positive and negative samples are unequal. Therefore we used precision, recall and F -score to evaluate the models. Because the positive samples were about 10 times more than negative samples, we calculate precision, recall and F-score base on negative samples. We also reported a Confusion Matrix for each model we test on our dataset. we tested our methods with various parameters and finally we choosed some of them that worked better due to hardware and time. The overall form of the reported confusion matrix is shown in Table 1. Table 2 reports the experimental results of each individual model. The confusion matrixes of the models are also reported in Tables 3, 4, 5 and 6.

Table 1: *overall form of the confusion matrix*

	Selected as Negative	Selected as Positive
Negative	TN	FP
Positive	FN	TP

Table 2: *presicion, recall and f-score measures of the models*

Approach	Precision	Recall	F-score
BERT (unbalanced data for fine-tuning and testing)	0.44	0.63	0.51
BERT (balanced data for fine-tuning and unbalanced data for testing)	0.32	0.98	0.48
BERT (positive data twice the negative data for fine-tuning and unbalanced data for testing)	0.49	0.89	0.63
BERT (balanced data for fine-tuning by increasing negative documents and unbalanced data for testing)	0.35	0.56	0.43

Table 3: *Confusion matrix for BERT (unbalanced data for fine-tuning and testing)*

	predicted as negative	Predicted as positive
negative	188	235
positive	109	4472

Table 4: *Confusion matrix for BERT (balanced data for fine-tuning and unbalanced data for testing)*

	predicted as negative	Predicted as positive
negative	415	8
positive	849	3732

Table 5: *Confusion matrix for BERT (positive data twice the negative data for fine-tuning and unbalanced data for testing)*

	predicted as negative	Predicted as positive
negative	378	45
positive	390	4191

Table 6: *Confusion matrix for BERT (balanced data for fine-tuning by increasing negative data and unbalanced data for testing)*

	predicted as negative	Predicted as positive
negative	267	205
positive	480	5032

5 Discussion

We do not have enough data to achive fine word representation vectors in this task, that's why we used a pre-trained network. Pre-trained representations can also either be context-free or contextual, and contextual representations can further be unidirectional or bidirectional. Context-free models such as word2vec or GloVe generate a single "word embedding" representation for each word in the vocabulary, contextual models instead generate a representation of each word that is based on the other words in the sentence.

BERT is a method of pre-training language representations that outperforms previous methods because it is the first unsupervised, contextual, deeply bidirectional system for pre-training NLP. It uses a simple approach for being contextual: it mask out 15 percent of the words in the input, run the entire sequence through a deep bidirectional Transformer encoder, and then predict only the masked words. Due to the structure we described, BERT can learn term dependencies, therefore Polarity of words and sentiment of sentences are well recognized.

As the results show we improved the reference article [3] and showed that BERT works better that skip-gram and LSTM or CNN.

References

[1] B. Liu, "Sentiment Analysis and Opinion Mining," *Synthesis lectures on human language technologies*, vol. 5(1), pp. 1-167, May 2012.

[2] B. Pang and 1. Lee, "Opinion Mining and Sentiment Analysis," *Foundations and trends in information retrieval*. vol. 2(1-2), pp. 1-135, January 2008.

[3] A. 1. Maas, R. E. Daly, P. T. Ph am, D. Huang, A. Y. Ng, and C. Potts, "Learning Word Vectors for Sentiment Analysis," *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. I, pp. 142-150, June 2011, Association for Computational Linguistics.

[4] Roshanfekr, Behnam, Shahram Khadivi, and Mohammad Rahmati. "Sentiment analysis using deep learning on Persian texts." 2017 *Iranian Conference on Electrical Engineering (ICEE). IEEE,* 2017.

[5] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[6] Google-Research. (2019, July 16). Google-research/bert. Retrieved from https://github.com/google-research/bert