Comparing BERT against traditional machine learning text classification

Santiago González-Carvajal
Universidad Politécnica de Madrid
Madrid, Spain
santiago.gonzalez-carvajal@alumnos.upm.es
Eduardo C. Garrido-Merchán
Universidad Autónoma de Madrid
Madrid, Spain
eduardo.garrido@uam.es

***Abstract*—The BERT model has arisen as a popular state-of-the-art model in recent years. It is able to cope with NLP tasks such as supervised text classification without human supervision. Its flexibility to cope with any corpus delivering great results has make this approach very popular in academia and industry. Although, other approaches have been used before successfully. We first present BERT and a review on classical NLP approaches. Then, we empirically test with a suite of different scenarios the behaviour of BERT against traditional TF-IDF vocabulary fed to ML algorithms. The purpose of this work is adding empirical evidence to support the use of BERT as a default on NLP tasks. Experiments show the superiority of BERT and its independence of features of the NLP problem such as the language of the text adding empirical evidence to use BERT as a default technique in NLP problems.**

I. Introduction

Natural Language Processing (NLP) methodologies such as text classification or summarization [9], have flourished. We can differentiate between two types of approaches to NLP problems: Firstly, linguistic approaches [2] that use different features of the text that the experts on the domain consider that are relevant. Those features could be combinations of words, or n-grams, grammatical categories, unambiguous meanings of words and much more. These features could be built manually or can be retrieved by using linguistic resources.

On the other hand, Machine Learning (ML) have analyzed annotated corpora of texts inferring which features of the text, typically in a bag of words fashion or by n-grams, are relevant for the classification automatically. Both approaches have their pros and cons, concretely, linguistic approaches have great precision but their recall is low as the context where the features are useful is not as big as the one processed by ML. However, the precision of classical NLP systems was, until recently, generally better than the one delivered by ML [4]. Nevertheless, recently, thanks to the rise of computation, ML text classification dominates.

The advantage of linguistic approaches over ML is that they do not need large amounts of data. We can find many examples of ML approaches: BERT [3], Transformers [13], etc. An issue with traditional NLP approaches is multilingualism. We can design rules for a given language, but sentence structure, and even the alphabet, may change from one language to another, resulting in the need to design new rules. Bidirectional Encoder Representations from Transformers (BERT) is a NLP model that was designed to pretrain deep bidirectional representations from unlabeled text and, after that, be fine-tuned using labeled text for different NLP tasks [3].

In this work we compare BERT model [3] with a traditional ML NLP approach that trains ML algorithms in features retrieved by the Term Frequency - Inverse Document Frequency (TF-IDF) algorithm as a representative of these traditional approaches [11]. We have carried out four different experiments about text classification. We start by presenting some related work, then, we describe the models, after that, we describe the experiments we have carried out and, finally, we present the conclusions drawn from the work.

II. Related Work

In this section, we summarize the main comparisons against advanced models such as the BERT transformer and classical natural language processing. Recently, BERT has achieved state-of-the-art results in a broad range of NLP tasks [3]. Hence, it is interesting to study how does the BERT model represent the steps of the traditional NLP pipeline in order to make a fair comparison.

An argument that defends classical ML NLP approaches is that the BERT approach need huge amounts of texts to deliver proper results. An interesting work [12] that focus on a pure empirical comparison of BERT and ULMFiT [10] w.r.t traditional NLP approaches in low-shot classification tasks where we only have 100-1000 labelled examples per class shows how BERT, representing the best of deep transfer learning, is the best performing approach, outperforming top classical ML algorithms thanks to the use of transfer learning [3]. A common critique of classical NLP practitioners is that the BERT model and ML methodologies can be fooled easily, commiting errors that may be severe in certain applications and that can be easily solved by symbolic approaches. Following this reasoning, in this work [7] the authors present the TextFooler baseline, that generates adversarial text in order to fool BERT's classification [7].

III. The BERT model and the traditional ML NLP methodology

Having reviewed related work, we will now introduce the traditional NLP approaches that we are comparing with BERT and then, the details of the BERT model.

A classical way to deal with a supervised learning NLP task is to build a bag-of-words model with the most weighted words given by the TF-IDF algorithm. Assuming there are N documents in the collection, and that term t_i occurs in n_i of these documents. Then, inverse document frequency can be computed as: $idf(t_i) = log\frac{N}{n_i}$. Actually, the original measure was an integer approximation to this formula, and the logarithm was base 2. On the other hand, given a term t_i, we denote by tf_i the frequency of the term t_i in the document under consideration. Finally, TF-IDF is defined for a given term t_i in a given document as follows: $tfidf(t_i) = tf_i \cdot idf(t_i)$.

We now explain what we consider to be the state-of-the-art technique on natural language processing. Regarding the BERT model, there are two steps in its framework: *pre-training* and *fine-tuning* [3]. During pre-training, the model is trained on unlabeled large corpus. For fine-tuning, the model is initialized with the pre-trained parameters, and all the parameters are fine-tuned using labeled data for specific tasks. BERT's model architecture is a multi-layer bidirectional Transformer encoder [3]. This kind of encoder is composed of a stack of $N = 6$ identical layers. Each of these layers has two sub-layers. The first one is a multi-head self-attention mechanism, and the second one, is a simple position-wise fully connected feed-forward network. It employs a residual connection [6] around both sub-layers, followed by a layer normalization [1]. That is, the output of each sub-layer is $LayerNorm(x + Sublayer(x))$, where $Sublayer(x)$ is the function implemented by the sub-layer [13]. In relation to, multi-head self-attention, first, we need to define scaled dot-product attention. It is define as follows: $Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$, where Q is the matrix of queries, K is the matrix of keys, V is the matrix of values and d_k is the dimension of the Q and K matrices. Now, we can define multi-head attention as $MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O$, where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$. Multi-head attention consists on projecting the queries, keys and values h times with different, learned linear projections to d_k, d_k and d_v (dimension of the values matrix), respectively. Then, on each of these projected versions of the queries, keys and values, we perform the attention function in parallel, yielding in d_v-dimensional output values. Finally, these are concatenated and projected, resulting in the final values [13]. Self-attention means that all of the keys, values and queries come from the same place.

BERT represents a single sentence or a pair of sentences (for example, the pair $\langle question, answer \rangle$) as a sequence of tokens. For a given token, its input representation is constructed by summing the corresponding token, position, and segment embeddings [3]. Pre-training is divided into: *Masked LM* and *Next Sentence Prediction (NSP)*. The first one, consists in masking some percentage of the input tokens at random, and then, predict those masked tokens. The second one consists in, given two sentences `A` and `B`, 50% of the time `B` is the actual next sentence that follows `A` (labeled as `IsNext`), and 50% of the time `B` is a random sentence from the corpus (labeled as `NotNext`) [3]. Fine-tuning is straightforward since the self-attention mechanism in the Transformer allows BERT to model many downstream tasks. For each task, we simply plug in the specific inputs and outputs into BERT and fine-tune all the parameters [3].

IV. Experiments

In order to compare BERT model with respect to the traditional ML NLP methodology, we have designed four experiments that are described throughout the section.

In these experiments, we will be using `TfidfVectorizer` from `sklearn` Python 3 module. After using TF-IDF to preprocess the text, we will be using `Predictor` from `auto_ml` module (in the third and fourth experiments), and `H2OAutoML` from `h2o` module (in the second experiment), to find the best model to fit the data. In the first experiment, we will, instead, show how much work needs to be done in order to get close to the results obtained, with no effort, using BERT model. For this purpose, we will be using many `sklearn` models and study their results in depth.

Regarding BERT's implementation, we have used the pre-trained BERT model from `ktrain` Python 3 module. This model expects the following directory structure: a directory which must contain two subdirectories: **train** and **test**. Each one of them, in turn, must contain one subdirectory per class (named after the name of the class they represent). And, finally, each class directory, must contain the `'.txt'` files (their name is irrelevant) with the texts that belong to the class they represent.

In the first experiment, we have downloaded the IMDB dataset. It contains 50000 movie reviews (25000 to train the model and 25000 to test it) to perform sentiment analysis, a popular supervised learning text classification task. We have compared the behaviour of a pre-trained default BERT model w.r.t different popular ML models such as SVC or Logistic Regression that use a vocabulary extracted from a TF-IDF model obtaining the following results:

Model	Accuracy
BERT	**0.9387**
Voting Classifier	0.9007
Logistic Regression	0.8949
Linear SVC	0.8989

TABLE I

Accuracy retrieved by the different methodologies in the IMDB experiment over the validation set.

Our second experiment deals with the RealOrNot tweets written in English. The task to solve here is binary text classification. It contains tweets classified into two different classes: Tweets about a real disaster and those that are not. We have just used the *tweet* and *class* columns. After that, we have generated the directory structure that we need to use BERT model (using 75% data to train and 25% data to validate). The obtained results have been summarized in the following table:

Model	Accuracy	Kaggle Score
BERT	**0.8361**	**0.83640**
H2OAutoML	0.7875	0.77607

TABLE II
REALORNOT EXPERIMENT RESULTS.

Having seen that BERT has outperformed an AutoML technique and other classical ML algorithms using a vocabulary built from a traditional NLP technique such as TF-IDF in the English language, we choose to change the language to see if the BERT model also behaves well. We have downloaded the Portuguese news dataset. It contains articles from the news classified into nine different classes. We have just used the *article text* and *class* columns. We have generated the directory structure that we need to use BERT model (using 75% data to train and 25% data to validate obtaining the following results:

Model	Accuracy	Kaggle Score
BERT	**0.9093**	**0.91196**
Auto ML	0.8480	0.85047

TABLE III
PORTUGUESE NEWS EXPERIMENT RESULTS.

Our last experiment involves a completely different language, Peninsular Chinese simplified characters zh-CN, where we hypothesize that, given that the way of expressing this Language is through different symbols that are not separated by spaces BERT may not output a good result. The experiment is a sentiment analysis problem involving Chinese hotel reviews. It contains hotel reviews classified into two different classes: Positive hotel reviews and negative hotel reviews. In this experiment, we have used 85% of the data to train the model and 15% of the data to validate it. Results are given in the following table:

Model	Accuracy
BERT	**0.9381**
Predictor (auto_ml)	0.7399

TABLE IV
CHINESE HOTEL REVIEWS RESULTS.

We can observe how, independently of the language and its characteristics, BERT behaviour outperforms classical NLP approach. In this experiment, the importance of transfer learning has become apparent, since the dataset was pretty small.

V. CONCLUSIONS AND FURTHER WORK

In this work we have introduced the BERT model and the classical NLP strategy where a ML model is trained using the features retrieved with TF-IDF and hypothesize about the behaviour of BERT w.r.t to tackle NLP tasks. We have introduced four different NLP scenarios where we have shown how BERT has outperformed the traditional NLP approach. Critically, implementing BERT has turned out to be far less complicated than implementing traditional methods. It is also noteworthy the importance of transfer learning. We have been able to obtain this results thanks to pre-training, as it become apparent in the last experiment. We are nevertheless aware of the limitations of the BERT model. Although it seems a good default for NLP tasks, it can be improved. Hence, we would like to apply Bayesian optimization for BERT to enable classification of messages for robots [8] [5] showing consciousness correlated behaviours.

ACKNOWLEDGMENTS

Authors gratefully acknowledge the use of the facilities of Centro de Computacion Cientifica (CCC) at Universidad Autónoma de Madrid. The authors also acknowledge financial support from Spanish Plan Nacional I+D+i, grants TIN2016-76406-P and from PID2019-106827GB-I00 /AEI /10.13039/501100011033.

REFERENCES

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Erik Cambria and Bebo White. Jumping nlp curves: A review of natural language processing research. *IEEE Computational intelligence magazine*, 9(2):48–57, 2014.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] Eduardo C Garrido and Jesús Cardenosa Lera. Expansión supervisada de léxicos polarizados adaptable al contexto.

[5] Eduardo C Garrido-Merchán, Martin Molina, and Francisco M Mendoza. An artificial consciousness model and its relations with philosophy of mind. *arXiv preprint arXiv:2011.14475*, 2020.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*, 2019.

[8] Eduardo C Garrido Merchán and Martín Molina. A machine consciousness architecture based on deep learning and gaussian processes. *arXiv preprint arXiv:2002.00509*, 2020.

[9] Cristina Puente, José Angel Olivas, E Garrido, and R Seisdedos. Creating a natural language summary from a compressed causal graph. In *2013 joint ifsa world congress and nafips annual meeting (ifsa/nafips)*, pages 513–518. IEEE, 2013.

[10] Kristian Rother and Achim Rettberg. Ulmfit at germeval-2018: A deep neural language model for the classification of hate speech in german tweets. 2018.

[11] Bruno Trstenjak, Sasa Mikac, and Dzenana Donko. Knn with tf-idf based framework for text categorization. *Procedia Engineering*, 69:1356–1364, 2014.

[12] Peter Usherwood and Steven Smit. Low-shot classification: A comparison of classical and deep transfer machine learning approaches. *arXiv preprint arXiv:1907.07543*, 2019.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.