Deep Learning Based Text Classification

ECE504 Data Mining and Knowledge Discovery , Dr. Saed ALQARALEH

Yahya Tawil

July 1, 2021

HASAN KALYONCU
ÜNİVERSİTESİ

1 Abstract

Contextualized embedding techniques like Bidirectional Encoder Representations from Transformers (BERT) made a breakthrough in the application of text classification. In this project we will build a text classifier to predict the topic of web-page text. The model compares the BERT representation between the text extracted from the web-page and the desired labels to find the Cosine similarity between them. This model is built once and can be applied to many text classes.

2 Introduction

One branch of Natural language processing (NLP) is performing text classification. This topic has made many breakthroughs in the last few years and one of them was after publishing publicly Bidirectional Encoder Representations from Transformers (BERT) which was developed and published [1] by Google back in 2018. BERT is based on Transformers in machine learning as a building block. By looking at the results of different approaches used in the General Language Understanding Evaluation (GLUE) [2] benchmark we can see that BERT-based approaches dominate till the time of writing these words.

Yet another breakthrough in terms of producing a generic classifier model is by training the model once and applying it to many text classes either seen or not seen before and that is doable by building BERT based zero-shot classifiers [1]. That can classify text into labels that the system did not see before as shown in figure 1.

[1] `https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html`

[2] `https://gluebenchmark.com/leaderboard`

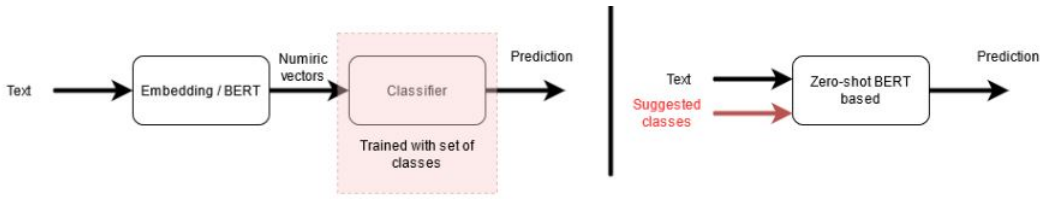

Figure 1: The system can have (left) an BERT embedding with a trained classifer for set of labels or using (right) Zero-shot BERT based can build a general classifier that can take the text classes as an input instead of having a dedicated classifier trained only on few classes.

This will eliminate the need to build or search for a dataset for labeled text with desired classes which is a costly and tedious task.

2.1 What is good about BERT?

Having the enough training data was and still one of the biggest challenges in NLP. To overcome this gap, A general-purpose and pre-trained models like BERT provides the ability to be fine-tuned on smaller task-specific datasets, e.g., when working with problems like questions answering.

To start off, Word embedding main task is to find a lower dimensional space using a dense vector representations of words. The word embedding models are now utilized in almost all NLP neural networks, and the reason for such wide usage is due to providing the possibility to model the semantic of a word in a numeric representation and therefore later perform mathematical operations on it when needed.

BERT is based on what is called Transformers and does contextualized embedding method of pre-training language representations. Starting from a model trained using a large text corpus, we get general-purpose language understanding model and then use that model for narrower NLP task that we care about like classification, answering questions, auto-completion, ...etc.

Transformers were introduced [2] to avoid recursion and this allowed parallel computation thus reducing training time. Sentences are processed in Transformers as a whole rather than word by word. Also, embeddings are

positioned thus encoded information is related to a specific position of a token in a sentence.

On the other hand, traditional RNN and CNN models using bag of words have performance very limited compared to classifiers with BERT based Embedding. For example in ordinary CNN to cover possible combinations of words the number of different kernels required would be huge because of the growing number of combinations when increasing the maximum length size of input sentences [3].

Moreover, BERT compared to static embedding techniques like Word2Vec can provide several contextualized advantages where representations are dynamically informed by the surrounding words. While on the other side, each word has a fixed representation under Word2Vec despite the context within which the word appears. For example: in "He has a beautiful cat." and "CAT is the world's leading manufacturer of construction and mining equipment" the 'Cat' representation in Word2Vec is the same while under BERT the word embedding for "cat" would be different for each sentence.

2.2 What is BERT?

BERT is a pretrained model comes in two forms, Basic and Large depending on the number of used encoders as shown in Figure 2. It expects input data in a specific format using the masked language modeling (MLM) for example: [CLS] at the beginning of our text and [SEP] to mark the end of a sentence, or the separation between two sentences. BERT is trained on and expects sentence pairs, using ones and zeros to distinguish between the two sentences. So for each token in tokenized text we must specify which sentence it belongs to: first sentence (a series of 0s) or second sentence (a series of 1s).

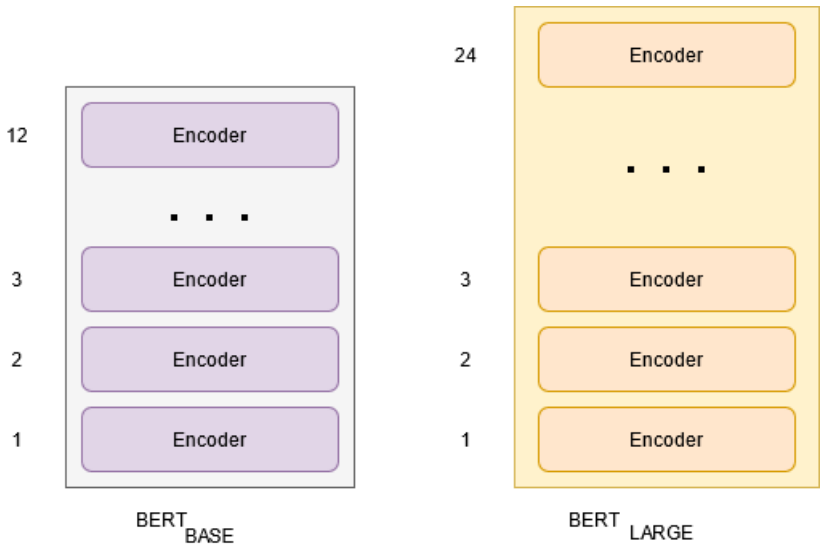

Figure 2: Base BERT model consists of 12 encoders and Large BERT model consist of 24 encoders

We built a demo using BERT to put all together by building a classifier for news data source shown in Figure 3.

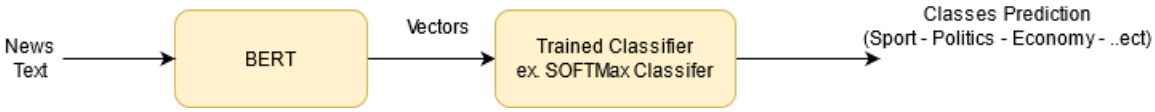

Figure 3: The block diagram of a demo was built to investigate BERT-based news topic classifier

To build the model shown in figure 4, a code is built and hosted on Google Colab[3] demonstrates how to build BERT layer by using a ready model hosted in Tensorflow hub[4] [5]. The Basic BERT consists of 12 transformer blocks, 768 hidden layer size. Finally, it has 12 attention heads and "Uncased" means that it is just for lowercase sequences [5].

[3] https://colab.research.google.com/drive/17vqDyynQAJuDIVMe1b6PJz9q5PsfCOJC?usp=sharing

[4] https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/2

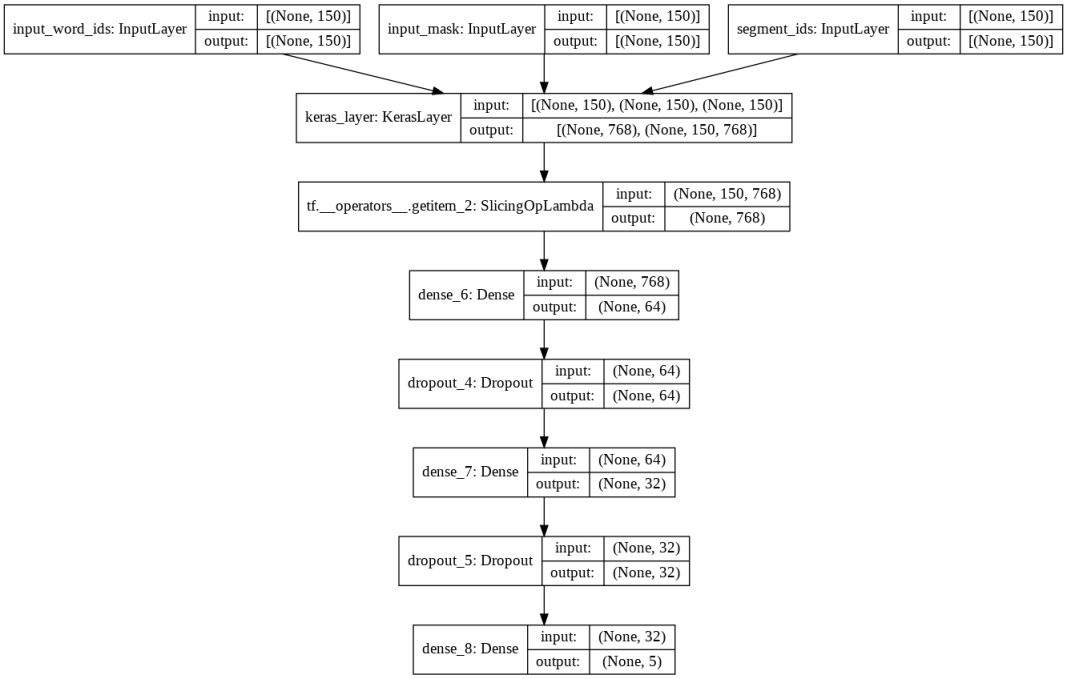

Figure 4: The detailed network structure of a demo was built to investigate BERT-based news topic classifier. First we applied BERT to get numerical representation, and then we used a simple full-connected layers as a classifier

The classifier added after the BERT layer, need to be trained. Thus, AG's news topic classification dataset [5] was used. In figure 5 an example of this dataset content. It has 3 columns: Label (4 labels), Title and a description. A 30,000 training samples and 1,900 testing samples for each class, and with a total number of training samples is 120,000 and testing 7,600.

[5] `http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html`

	label	Title	Description
0	3	Wall St. Bears Claw Back Into the Black (Reuters)	Reuters - Short-sellers, Wall Street's dwindli...
1	3	Carlyle Looks Toward Commercial Aerospace (Reu...	Reuters - Private investment firm Carlyle Grou...
2	3	Oil and Economy Cloud Stocks' Outlook (Reuters)	Reuters - Soaring crude prices plus worries\ab...
3	3	Iraq Halts Oil Exports from Main Southern Pipe...	Reuters - Authorities have halted oil export\f...
4	3	Oil prices soar to all-time record, posing new...	AFP - Tearaway world oil prices, toppling reco...

Figure 5: Example of Dataset content

3 Related Work

The quality of sentence embedding models are increasing recently by having new models on top of BERT like Sentence-BERT [7]. Sentence-BERT is a recent technique which fine-tunes the pooled BERT sequence representations for increased semantic richness, as a method for obtaining sequence and label embedding Therefore, some experiments in Table 1.

Table 1: Example output from Sentence-BERT

Example Text	Input Classes	Result (Cos Similarity)
Hasan Kalyoncu University is Turkish private university located in Gaziantep.	'technology','history','business', 'art and culture', 'politics', 'education'	label: education similarity: 0.1903897374868393 label: business similarity: 0.1696242094039917 label: history similarity: 0.1575978547334671 label: technology similarity: 0.11214858293533325 label: politics similarity: 0.08053351193666458 label: art and culture similarity: 0.04724632203578949
'Gaziantep, previously and still informally called Antep, is the capital of Gaziantep Province, in the western part of Turkey Southeastern Anatolia Region, some 185 kilometres east of Adana and 97 kilometres north of Aleppo, Syria.'	'technology','business', 'art and culture', 'politics', 'history'	label: history similarity: 0.11404849588871002 label: politics similarity: 0.06558618694543839 label: art and culture similarity: 0.05085340142250061 label: business similarity: 0.02563968487083912 label: technology similarity: -0.03650127723813057
'AirPods are wireless Bluetooth earbuds created by Apple. They were first released on September 7, 2016 , with a 2nd generation released in March 2019.'	'technology','business', 'art and culture', 'politics', 'history'	label: technology similarity: 0.11916965246200562 label: business similarity: -0.03503386676311493 label: art and culture similarity: -0.05030977725982666 label: politics similarity: -0.06877294182777405 label: history similarity: -0.12665867805480957

Zero-shot text classification (or 0SHOT-TC) [6] generalizes the classifier trained on a known label set to an unseen without using any new labeled data. Zero-shot learning requires providing a descriptor for the unseen class (or simply the class name).

We investigated the 0SHOT-TC using a pre-trained MNLI sequence-pair classifier as an out-of-the-box zero-shot text classifier. It translates each candidate label into a "hypothesis" by taking the sequence we're interested in labeling as the "premise." We accept the label as true if the NLI model predicts that the premise "entails" the hypothesis. Two statements are considered in natural language inference (NLI): a "premise" and a "hypothesis." The code to test this approach is provided in Colab [6]. The model is used from Hugging Face Transformers which is a Python-based library [7] that exposes an API to use many well-known transformer architectures. The examples used to evaluate this model is presented in Table 2

[6] https://colab.research.google.com/drive/1ng1iCpgFzVGlVKkRJny-WzSBuQZNX0fB?authuser=3#scrollTo=Czx6dcBI0NOf

[7] https://blog.tensorflow.org/2019/11/hugging-face-state-of-art-natural.html

Table 2: Example output from 0SHOT-TC

Premise	Hypothesis	Probability
Hasan Kalyoncu University is Turkish private university located in Gaziantep.	'This text is about education.'	96.85%
Hasan Kalyoncu University is Turkish private university located in Gaziantep.	'This text is about politics.'	5.59%
'Baklava is a layered pastry dessert made of filo pastry, filled with chopped nuts, and sweetened with syrup or honey.'	'This text is about food.'	98.49%
'Baklava is a layered pastry dessert made of filo pastry, filled with chopped nuts, and sweetened with syrup or honey.'	'This text is about dishes.'	83.93%

4 Web-page Text Classification using SBERT

In Figure 6 a diagram showing a general structure of the system that has been tested and evaluated. The system has 2 main parts: The downloader threads that are responsible for downloading webpages and extracting URLs, clean text and the included images. Later, the clean text will be stored in a shared CSV file and the URLs will be added to a queue.

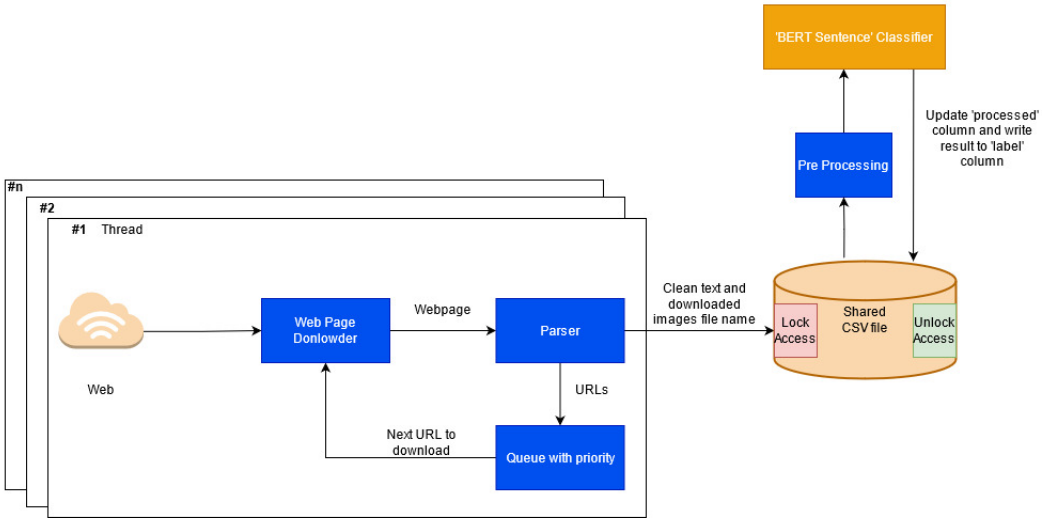

Figure 6: The system architecture

4.1 The Downloader

To increase the speed of the system, a multi-thread downloader was implemented. Instead of having one shared queue between the threads, the system used a dedicated URL indexed-list, called in python dictionary data type, to each crawler instance using the domain name for indexing. This will make the size of the URL list increase for each domain in each crawler instance separately as shown in Figure 7. During experiments, the threads may die due to any run-time exception and to guarantee that our system keeps running the same number of threads, a periodic check is done to detect any dead thread to be later inserted again to the system. Lastly, we introduced a list of website to prevent our system from downloading its content due to either the need to log-in in order to get the content (.i.e: Instagram) or the website content may not be interested in the current design design (.i.e YouTube).

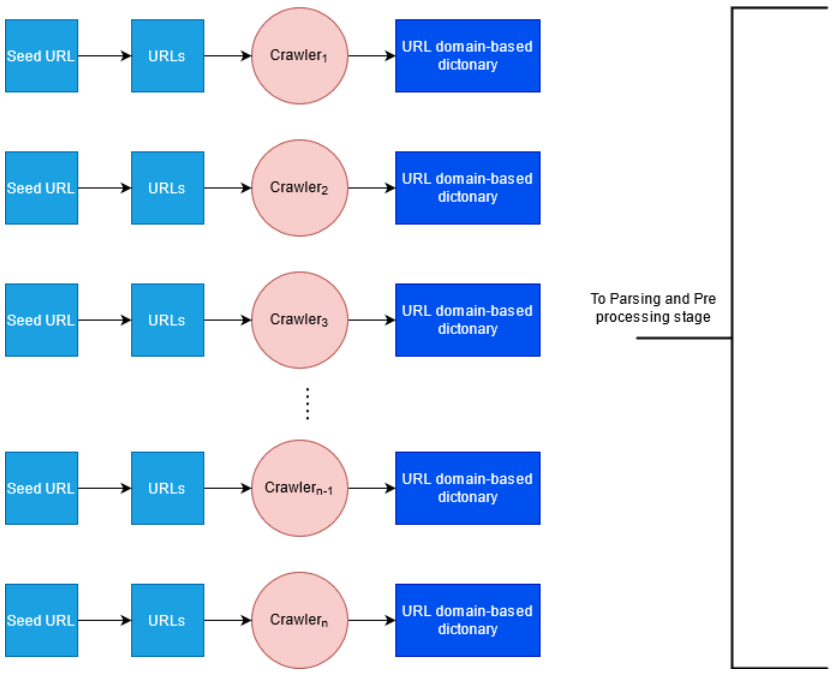

Figure 7: The Multi-Threaded downloader.

Each thread writes the results to a shared CSV file and as it is shared, a simple blocking lock available in threading python module is used to manage access to this shared resource between the downloader thread and classifier thread.

4.2 The Classifier

A Sentence-BERT (SBERT) [7] is used as a classifier which is a pretrained BERT network. The classifier can take whatever labels the user specifies in the system settings and try to predict how far the text is close to each label. Let's have an embedding model Φ, set of classes name C and input text x , then the classifier output $\hat{C}$ is according to the following equation:

$$\hat{c} = \arg\max_{c \in C} \cos\left(\Phi_{\text{sent}}\left(x\right), \Phi_{\text{sent}}\left(c\right)\right)$$

In Figure 8 an explanation of the usage of SBERT to classify the webpage text into the user specified classes.

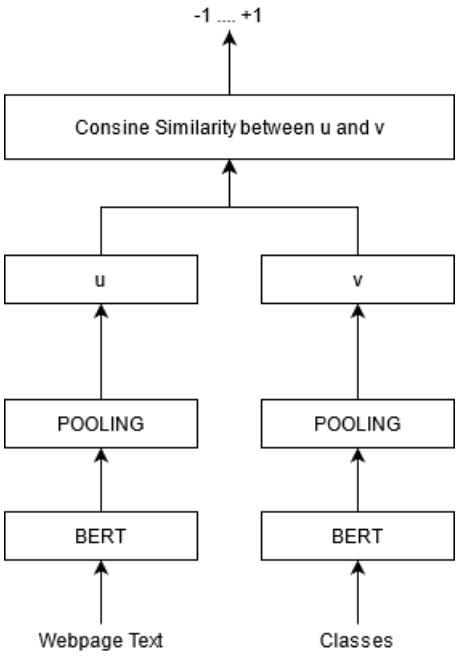

Figure 8: How SBERT model is used in the system: We find the BERT embedding of the labels and the webpage text and then find the best cosine similarity. -1 represents the worst and +1 the best matching.

5 Experiments

By running the system using the following settings:

Table 3: Experiment Settings

Base URLs	Num of threads	Labels	Black-listed websites
History: https://www.worldhistory.org https://www.newworldencyclopedia.org/ https://www.ushistory.org https://www.historic-uk.com/ **Business**: https://hbr.org/ **Politics**: https://newpol.org/ **Technology**: https://pc.net/ https://www.computerhope.com https://www.computerlanguage.com/	9	technology business politics history	facebook instagram youtube

The system will be evaluated based on the output CSV file. The CSV file has the following columns:
"timestamp": store the timestamp of when this webpage is downloaded. "Hash": 8-digits hash to represent the URL numerically to speedup search for duplicates in the shared CSV file. "Link": Explicit and full URL of the webpage. "post text": The extracted clean text from the webpage with no HTML tags or any other meta-data. "Downloaded imgs": a list of downloaded images file name. "Processed": A field contains 'yes' if this webpage text was classified or 'no' if not. "Label": the predicted labels from text classifier. "True label": The reference label of the URL.

To evaluate the output of the main system, we developed a small system that takes the CSV file to do the following: 1- Export a list of domain names of fetched webpages. 2- Update the "True label" field with the true value based on a predefined table. 3- Calculate the matching and mismatch between the true label and the predicted one.

In the Table 4 a summary of evaluating a CSV file produced by the system. We can see that the percentage of matches are 90% for business, 70% for politics, and around 50% for technology and history.

Table 4: Matching and Mismatching results of web-pages among the user defined labels

Total prediction: 4460	Tech	History	Business	Politics
Match	1230	334	200	133
Mismatch	654	516	21	38

The down loader and classifier speed of how many webpage downloaded per second is shown in the diagram bellow. This is after running the system for 2 hours. It is obvious the linearity of the downloader and classifier speed over time. This means that downloader thread and classifier work on parallel on almost the same speed.

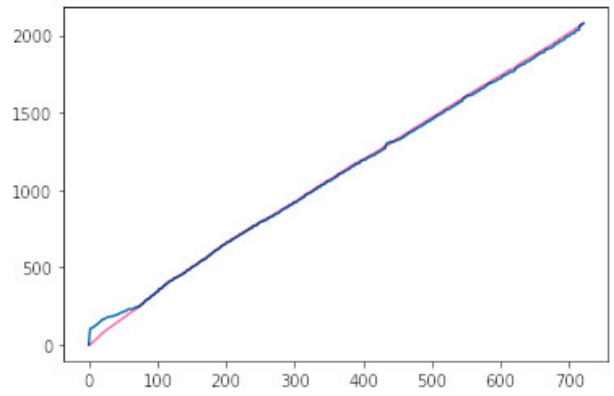

Figure 9: X-axis: time samples scaled by 10 (7200 Sec). Y-axis: The line in red is the classifier speed and line in blue the downloader speed.

Based on the experiments results we can say that Sentence-BERT is more efficient to be used sentence-level, not single- or multi-word representations like the label names. Yet this can be addressed with the help of word2vec

model representation for the labels name [8]. This is done by introducing Z. An additional transformation to S-BERT embedding for both sequences and labels. We add Z to equation (4.2). Z is a least-squares linear projection matrix Z with L2 regularization.

$$\hat{c} = \arg\max_{c \in C} \cos\left(\Phi_{\text{sent}}\left(x\right)Z, \Phi_{\text{sent}}\left(c\right)Z\right)$$

6 Conclusion

The proposed system can effectively classify fetched URLs and assign a predicted label based on a set of classes the user enters in the system settings. However, the threading design needs more enhancements to increase the system throughput. On the classifier side, the limited results precision of S-BERT confirms the need to enhance the S-BERT embedding mentioned in [8] especially for the labels that are not sentences rather a single-word.

7 References

1. Gupta, A., Thadani, K. and O'Hare, N., 2020, December. Effective Few-Shot Classification with Transfer Learning. In Proceedings of the 28th International Conference on Computational Linguistics (pp. 1061-1066).

2. Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." arXiv preprint arXiv:1706.03762 (2017).

3. "Why Does The Transformer Do Better Than RNN And LSTM In Long-Range Context Dependencies?". Artificial Intelligence Stack Exchange, 2021, `https://ai.stackexchange.com/questions/20075/why-does-the-transfo` . Accessed 23 Apr 2021.

4. "BERT Word Embeddings Tutorial · Chris Mccormick". Mccormickml.Com, 2021, `https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/` . Accessed 23 Apr 2021.

5. Chandrashekhar, Tandoori. "Simple Text Multi Classification Task Using Keras BERT!". Analytics Vidhya, 2021, `https://www.analyticsvidhya.com/blog/2020/10/simple-text-multi-classification-task-using-keras-bert/` . Accessed 23 Apr 2021

6. Yin, W.,Hay, J.,and Roth, D.2019. Benchmarking zero-shot text classification: Datasets, evalu-ation and entailment approach. InProceedings of the 2019 Conference on Empirical Methods inNatural Language Processing and the 9th International Joint Conference on Natural LanguageProcessing (EMNLP-IJCNLP), pp. 3905–3914

7. Reimers, N. and Gurevych, I., 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

8. Joe Davison, "Zero-Shot Learning in Modern NLP " `https://joeddav.github.io/blog/2020/05/29/ZSL.html` . Accessed 14 May 2021.