BERT Based Topic-Specific Crawler

Yahya Tawil
Computer Engineering Department
Hasan Kalyoncu University
Gaziantep, Turkey
yahya.tawil@std.hku.edu.tr

Saed Alqaraleh*
Computer Engineering Department
Hasan Kalyoncu University
Gaziantep, Turkey
saed.alqaraleh@hku.edu.tr

2021 Innovations in Intelligent Systems and Applications Conference (ASYU) | 978-1-6654-3405-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ASYU52992.2021.9599076

***Abstract*— Nowadays, retrieving certain information using search engines is very popular and one of the main applications of the Internet. To speed up the process of getting the required information(web pages), having a topic-specific crawler is essential to fetch and index only the relevant ones. This paper presents a multi-thread web crawler using a Sentence Bidirectional Encoder Representations from Transformers (S-BERT). The S-BERT is used to calculate the similarity between the predefined classes and the text of the downloaded web pages. This provides a lightweight model compared to using a word embedding with deep learning for text classification.**

***Keywords*— Search engine, Topic-Specific Crawler, Web crawler, Text classification, Text categorization, Document classification.**

I. INTRODUCTION

One branch of Natural Language Processing (NLP) is performing text classification. This topic has made many breakthroughs in the last few years, such as introducing the Bidirectional Encoder Representations from Transformers (BERT) by Google back in 2018. In general, BERT comes in two forms, Basic and Large depending on the number of used encoders as shown in Fig. 1.

Note that having enough training data was and still one of the biggest challenges in NLP. To overcome this gap, general-purpose and pre-trained models like BERT provides the ability to be fine-tuned on smaller task-specific datasets were introduced.

To start off, word embedding's main task is to find a lower-dimensional space using a dense vector representation of words. The word embedding models are now utilized in almost all NLP neural networks. The reason for such wide usage is due to providing the possibility to model the semantic of a word in a unique numeric representation. Also, BERT compared to static embedding techniques like Word2Vec and glove 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 and glove despite the context within which the word appears. For example: "He has a beautiful cat." and "CAT is the world's leading manufacturer of construction and mining equipment" the 'cat' representation in Word2Vec in both sentences is the same. In BERT, the word embedding for "cat" would be different for each sentence.

*Corresponding author

BERT is based on what is called transformers and does contextualized embedding method of pre-training language representations. Starting from a model trained using as much as available corpuses, we get a general-purpose language understanding model and then use that model for narrower NLP tasks such as classification, answering questions, auto-completion, etc. It is worth mentioning that transformers were introduced in [1] to avoid recursion and to allow 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.

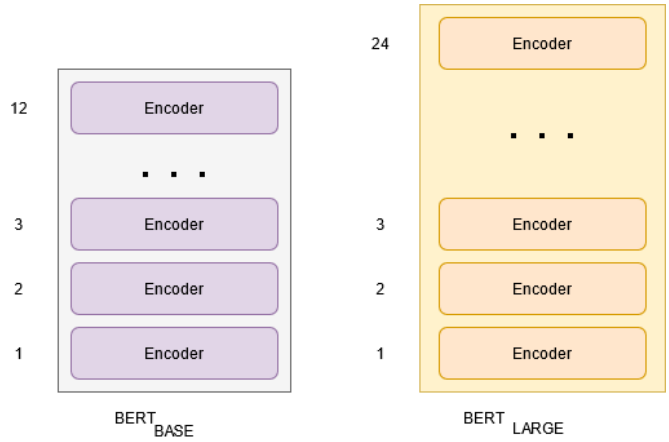

Fig. 1. BERT model, where the base consists of 12 encoders and the Large BERT model consists of 24 encoders.

BERT expects input data in a specific format using masked language modeling (MLM). For example, [CLS] is used at the beginning of the text and [SEP] to mark the end of a sentence or the separator between two sentences. BERT is trained on and expects sentence pairs, where ones and zeros are used to distinguish between the two sentences. So, for each token in the input(tokenized text), we must specify which sentence it belongs to, the first sentence (a series of 0s) or the second sentence (a series of 1s).

Related to the techniques for classifying the webpage content, most of the traditional works were based on keywords evaluation or weighting mathematically the appearance of user query inside the webpage's text. In our work, we employed a BERT-based classifier in the application of

978-1-6654-3405-8/21/$31.00 ©2021 IEEE

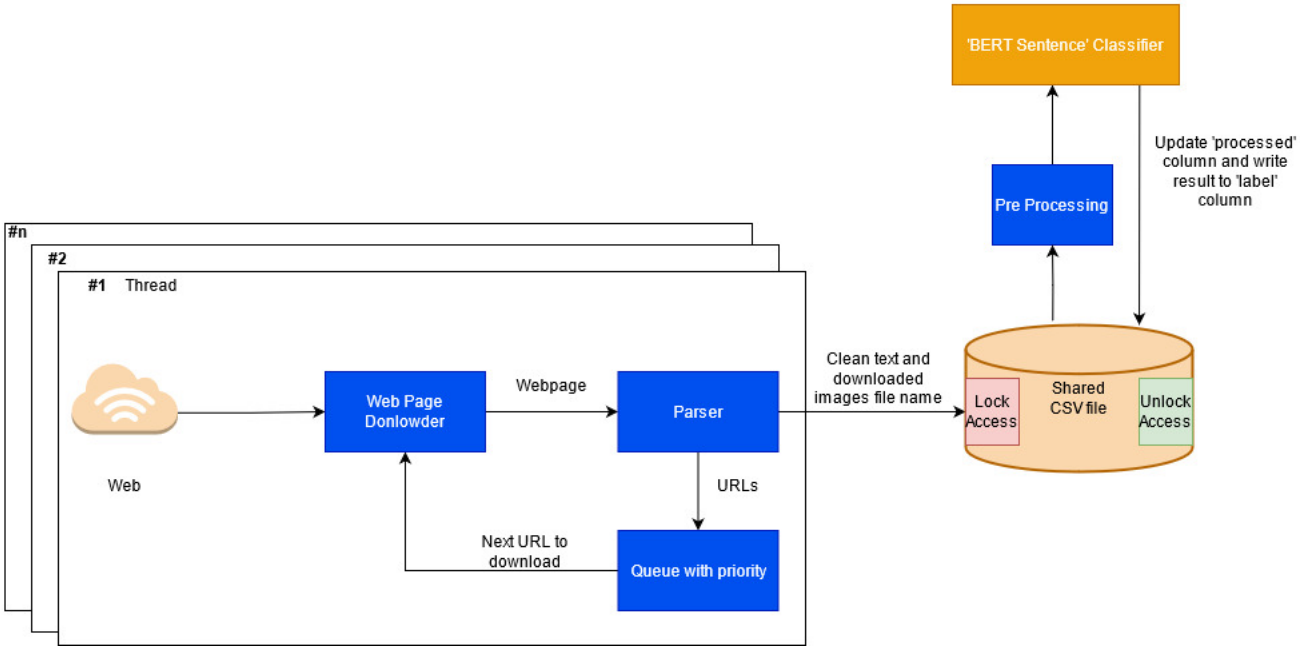

Fig. 2. The system architecture: multiple downloaders with text preprocessing included. The resultant cleaned text is stored into a CSV file to be later processed by the text classifier and write back the class prediction to the same CSV file.

webpage text classification due to the advantages and state-of-the-art performance of BERT based text classifier. We built a multi-thread crawler consist of downloaders (each one on a thread) that include text preprocessing and storing the cleaned text in a *CSV* file shared with the classification sub-system. We added the ability to download and index the webpage's images, however, a detailed study can be done as future work.

II. RELATED WORKS

In the last decade, many researches were performed to implement and improve the topic-specific crawlers with different aspects. One of these aspects is to use keywords-based evaluation, such as the work of [2], where a focused crawler based on query keywords was developed. Here, a score for each URL is calculated in order to have candidates as seeds to their retrieval system. Another focused crawler was built by Aggarwal [3] that aim to reduce the bandwidth consumed by the crawlers by trying to download only the relevant pages. This results in reducing the network load. In more detail, Aggarwal's method checks the appearance of user's query keywords in the page while considering the place of presence, i.e., a keyword present in the title has higher weight than being mentioned somewhere else. The same procedure will be applied for child pages (out links/hyperlinks). Finally, if the total relevance score crosses a predefined threshold, then the page will be downloaded, otherwise it won't. In [4], Shuguang et al., have designed and built a topic specific crawler to fetch offers from commercial web stores to answer user's queries. Their approach uses deep learning to predict price change of commercial offers. For more detail about topic-specific crawler, readers can check [5], [6] and [7].

Recently, the quality of embedding models were increasing by having some models on top of BERT like Sentence-BERT [8] and Zero-shot text classification (or 0SHOT-TC) [9], [10]. Where 0SHOT-TC generalizes the classifier trained on a known label set to an unseen ones. Zero-shot learning requires providing a descriptor for the unseen class (or simply the class name). This will eliminate the need to build or search for an annotated dataset(s), which is a costly and tedious task.

III. THE DEVELOPED WEB-PAGE SBERT- BASED CLASSIFIER

As shown in Fig. 2, the system has 2 main parts: The downloader threads that are responsible for downloading webpages, extracting URLs, and cleaning the content(text and images). Next, the cleaned text will be stored in a shared CSV file and the extracted URLs will be added to the URL's queue. Note that the classifier has the access to the CSV file to process(classify) the newly added webpages and then store back the class of each page to the same file.

A. The Downloader

A multi-thread downloader implemented to increase the speed of the system. The downloader, as shown in Fig. 3, used a dedicated URL list for each crawler instance. Hence, this way solves the bottleneck of having one shared queue between the threads. Our system keeps running the same number of threads by monitoring their state. A periodic check that detects any dead thread to be created again, as any thread may stop working due to some run-time exceptions. Lastly, we introduced a black-list of websites to prevent our system from downloading its content. These websites are blocked for two reasons. Firstly, it may require login in order to get

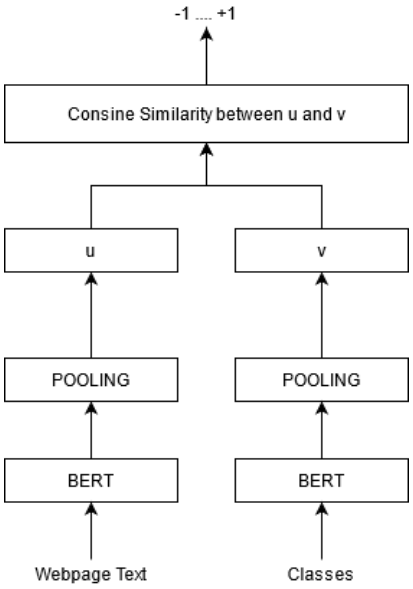

Fig. 4. Find the BERT representation of both the labels and the text of the webpage, and then find the cosine similarity. -1 represents the worst and +1 the best match.

the content, i.e., Facebook. Secondly, some websites have non-text content, i.e., YouTube.

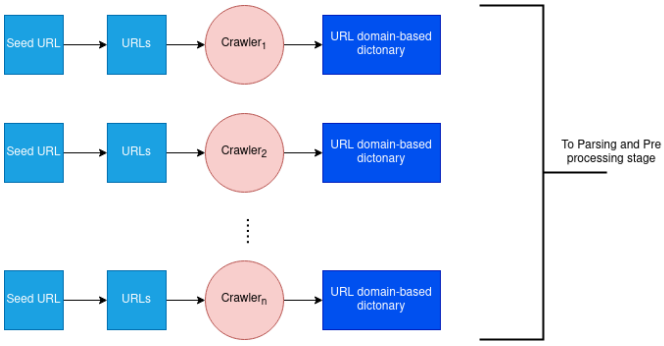

Fig. 3. An illustration for multi-threaded downloader showing the dedicated memory resource for each thread.

B. Text Classifier

Our classifier uses Sentence-BERT (SBERT) [11] for embedding. We calculate the cosine similarity of S-BERT output between the class name and the text to make the prediction. 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. Fig. 4 shows an illustration of using the SBERT to classify the webpage text into one of the predefined classes. Mathematically, this process can be expressed as follows.

$$\hat{c} = \underset{c \in C}{\arg\max} \cos\left(\Phi_{\text{sent}}(x), \Phi_{\text{sent}}(c)\right) \qquad (1)$$

Where: Φ is the embedding model. C is a set of the predefined classes. x is the input text.

IV. EXPERIMENTS

In this section, the proposed classifier was evaluated after using the crawler to download some webpages that

TABLE I. Experiments settings.

Base URLs	# of threads	Labels	Blacklisted
History: worldhistory.org newworldencyclopedia.org ushistory.org www.historic-uk.com Business: hbr.org Politics: newpol.org Technology: pc.net computerhope.com computerlanguage.com	9	technology business politics history	Facebook Instagram YouTube

was stored in a CSV file. In general, the CSV file has the following columns: "timestamp": to store the timestamp(downloading time). "Hash": A 8-digits hash to represent the URL numerically to speed up the search for duplicated URLs in the shared CSV file. "Link": The full URL of the webpage. "post-text": The extracted clean webpage's text, with no HTML tags or any other meta-data. "Downloaded imgs": A list of the downloaded images. "Processed": A field contains 'yes' if this webpage text was classified, or 'no' if not yet. "Label": The predicted labels (class). "True label": The reference label.

To evaluate the output of the main system, we developed a small script that takes the CSV file and perform the following: 1- Export a list of domain names of fetched webpages. 2- Update the "True label" field based on a predefined ground truth for each domain. 3- Calculate the matches and mismatches between the true label and the predicted one. It is worth mentioning that experiments were performed using the settings, shown in Table I.

A. Experiment #1: The Performance of the Developed System

In this experiment, our multi-thread crawler with SBERT classifier was tested by crawling and downloading 4000 webpages and the number of both correctly and incorrect predicted samples are presented in Table II. Overall, the percentage of matches is 90% for business, 70% for politics, and around 50% for technology and history.

TABLE II. Number of matching and mismatching samples.

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

Hence, It is clear that Sentence-BERT is more efficient to be used with a sentence level, not with single- or multi-word representations like the label names.In other words, SBERT performance seems to be downgraded when the similarity calculated between a sentence (webpage text content) and single word like labels (topic). Yet, this can be addressed

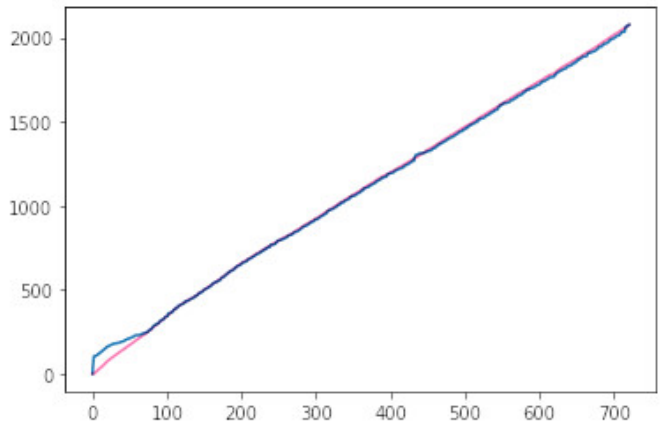

Fig. 5. X-axis: time samples scaled by 10 (total 7200 Sec). Y-axis: number of processed webpages. The line in red is the classifier speed and the line in blue is the downloader speed.

with the help of word2vec model representation for the labels name [9]. This is done by introducing an additional transformation to S-BERT embedding for both sequences and labels. We add the least-squares linear projection matrix Z with L2 regularization to equation 1.

$$\hat{c} = \arg\max_{c \in C} \cos\left(\Phi_{\text{sent}}\left(x\right)Z, \Phi_{\text{sent}}\left(c\right)Z\right) \quad (2)$$

As mentioned before: Φ is the embedding model. C is a set of the predefined classes. x is the input text.

B. Experiment #2: The Speed of the Developed Downloader and Classifier.

In this experiment, The speed of the developed downloader and classifier, i.e., how many webpages can be downloaded. The results are shown in Fig. 5. 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.

C. Experiment #3: SBERT vs. 0SHOT-TC

In this experiment, we explore the performance of Sentence-BERT compared to 0SHOT-TC. As mentioned before, Sentence-BERT is a recent technique which fine-tune the pooled BERT sequence representations in order to increase the semantic richness. Table III shows some examples of the processed samples.

Table IV shows the examples used to evaluate the 0SHOT-TC. It is worth mentioning that in this paper, we investigated the 0SHOT-TC using a pre-trained Multi-genre Natural Language Inference (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 considered in Natural Language Inference (NLI): a "premise" and a "hypothesis". The model is used from Hugging Face Transformers, which is a Python-based library that exposes an API to use many well-known transformer architectures.

TABLE III. The Cos Similarity of Some Examples using Sentence-BERT.

Input	Investigated Classes	Result (Cosine Similarity)
Hasan Kalyoncu University is Turkish private university located in Gaziantep.	technology, history, business, art and culture, politics, education	education:0.19 business: 0.16 history:0.15 technology:0.11 politics:0.08 art & culture:0.04
Gaziantep, previously and still informally called Antep, is the capital of Gaziantep Province, in the western part of Turkey Southeastern Anatolia Region, some 185 kilometers east of Adana and 97 kilometers north of Aleppo, Syria.	technology, business, art and culture, politics, history	history:0.11 politics:0.06 art & culture:0.05 business:0.02 technology:-0.03
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	technology: 0.11 business: -0.03 art & culture:-0.05 politics:-0.06 history:-0.12

TABLE IV. The Cos Similarity of Some Examples using 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%

Overall, as mentioned before, based on the results of the above experiments, we can say that Sentence-BERT is more efficient to be used for sentence-level, not single- or multi-word representations like the label names. In other words, SBERT performance seems to downgrade when the similarity is calculated between a sentence (webpage's text) and single words like labels (topics).

V. CONCLUSION AND DISCUSSION

The proposed system can effectively classify fetched URLs and assign a label based on a set of predefined classes. However, the crawler's threading design needs more enhancements to increase the system throughput. On the

classifier side, the limited results' of S-BERT confirm the need to enhance the S-BERT embedding as mentioned in [10] especially for the single-word labels. Finally, our system is capable to download and index images in order to be used in future work to classify the web pages using both their text and images.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[2] M. Kumar, A. Bindal, R. Gautam, and R. Bhatia, "Keyword query based focused web crawler," *Procedia Computer Science*, vol. 125, pp. 584–590, 2018.

[3] K. Aggarwal, "An efficient focused web crawling approach," in *Software Engineering*. Springer, 2019, pp. 131–138.

[4] S. Han, B. Brodowsky, P. Gajda, S. Novikov, M. Bendersky, M. Najork, R. Dua, and A. Popescul, "Predictive crawling for commercial web content," in *The World Wide Web Conference*, 2019, pp. 627–637.

[5] Y.-B. Yu, S.-L. Huang, N. Tashi, H. Zhang, F. Lei, and L.-Y. Wu, "A survey about algorithms utilized by focused web crawler," *Journal of Electronic Science and Technology*, vol. 16, no. 2, pp. 129–138, 2018.

[6] S. Kumar and M. Gupta, "A review of focused crawling schemes for search engine," *Smart Trends in Computing and Communications: Proceedings of SmartCom 2020*, pp. 311–317, 2021.

[7] S. M. Mirtaheri, M. E. Dinçktürk, S. Hooshmand, G. V. Bochmann, G.-V. Jourdan, and I. V. Onut, "A brief history of web crawlers," *arXiv preprint arXiv:1405.0749*, 2014.

[8] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[9] W. Yin, J. Hay, and D. Roth, "Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach," *arXiv preprint arXiv:1909.00161*, 2019.

[10] J. Davison, "Zero-Shot Learning in Modern NLP," https://joeddav.github.io/blog/2020/05/29/ZSL.html, [Online; accessed 14 May 2021].

[11] C. Mccormick, "BERT Word Embeddings Tutorial," https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/, [Online; accessed 23 Apr 2021].