Generating a Portuguese-European BERT based model using content from Arquivo.pt archive

Nuno Miquelina[0000−0002−3202−2242]
Paulo Quaresma[0000−0002−5086−059X] Vítor Beires Nogueira[0000−0002−0793−0003]

Universidade de Évora, Évora, Portugal
d37384@alunos.uevora.pt
{pq,vbn}@uevora.pt

Abstract. Building a language model from free available internet information takes several steps and challenges. This new model aims to be a BERT-based language model for Portuguese-European, with no specific context. The corpus was built using a web page archive infrastructure provided by Arquivo.pt and restricted to .pt domains. This paper will describe the overall process of building the corpus and training a BERT model.

Keywords: BERT · Vocabulary · Arquivo.pt · Portuguese European

1 Introduction and Motivation

The available text sources on the internet allow the gathering of vast amounts of content for training linguistics models [11]. These massive sources of information still need additional processing techniques like web scraping [6] to guarantee that only text with quality is retrieved, by eliminating pieces of code (HTML, JS, and others) from the actual text. Sentences and words are transformed into vectors (embeddings) and processed in an unsupervised way by the Deep Learning networks, generating language models that can be used for some Natural Language Processing (NLP) tasks like automatic translation among others [20].

Recurrent Neural Networks (RNN) were the main processing method for NLP tasks, but the true nature of this kind of neural networks fail or have less performance in processing long sequences because the first processed tokens get forgotten or lose importance. A novel approach introduced the concept of transformers [22]. This new architecture takes into account the weight of other tokens in the context. Based on this work, investigators presented BERT (Bidirectional Encoder Representations from Transformers) [5].

Recent benchmarks for evaluation of various tasks of natural language understanding (GLUE, MultiNLI, SQuAD v1.1 and SQuAD v2.0 benchmarks) showed that the BERT language representation model improved the state-of-the-art results. This new technique, for creating a language model representation, was designed to have a bidirectional context in all layers. Unlike other models like the OpenAI GPT (Generative Pre-Trained Transformer) that are based only in a left-to-right only context. The OpenAI GPT has evolved and originated the new GPT-2 and GPT-3 (bigger training dataset and number of parameters) improving the overall benchmarks [17,4]. Another approach gives attention to the morphology of words, like fastText [3], and allows training models on large corpora and compute word representations for first seen words. ELMo [10] is also an example of a pre-training model with context-sensitive word representations.

BERT good results inspired other investigators to follow their work and to propose new models, like RoBERTa (Robustly Optimized BERT Pretraining Approach) [13], trying to improve the processing robustness changing: training the model longer over more data, removing next sentence prediction objective, training on longer sequences and dynamically changing the masking pattern applied to the training data. BERTimbau [18] and CamemBERT [14] are other projects in Brazilian Portuguese and French respectively, that aim to train monolingual models and evaluate their performance in different NLP tasks.

The remainder of this paper is organized as follows: section 2 introduces the archive Arquivo.pt; in the section 3 we provided a description of the process to create the Portuguese-European corpus and how to use this corpus to train a BERT language model. Finally, section 4 presents our conclusions together with some pointers for future work.

2 Arquivo.pt

Arquivo.pt [7] is an investigation infrastructure that allows to search and access web pages archived since 1996. Arquivo.pt started in January 2008, but the original idea to create a Portuguese website archive started in 2001 with the scientific project “tumba!” from Faculty of Sciences of the University of Lisbon. This archive makes site content available to researchers, content that could get lost by disappearing from the original sites [8]. The crawling process underneath only considers sites related to Portugal, i.e., sites under the *.pt* domain or embedded on a page hosted in *.pt* or even redirected from a *.pt* domain.

Table 1 describes the volume of data archived and the infrastructure that supports the archive process.

Preserved data volume	Infrastructure
• 13 158 million web files	• 73 servers
• 28 million websites	• 17 TB of RAM memory
• 852 TB compressed content	• 1.816 vCPU
	• 1.186 Hard drives (4,5 PB)

Table 1. Arquivo.pt information (January 2022)

2.1 Arquivo.pt interfaces

Arquivo.pt gives provides API interfaces to access the repository, mainly by two ways:

– Text search: passing a query parameter with the target text, the response is a list of files that have that content. This type of query is not our goal, because we don't want to restrict the response to a given text.

– URL search: passing a given URL the server API (Wayback CDX server API) will return a list of captures of that URL. This is the API used to build the set of URLs to retrieve the content. There wasn't an applied any filter to limit the time when the URL was captured.

One example of the API calling is the following:

https://arquivo.pt/wayback/cdx?url=publico.pt that returns

the list of captures for "publico.pt".

3 Content Retrieve Process

The content retrieve process and model train from Arquivo.pt was developed in Python programming language, using several separated applications, one per process action. Fig 1 shows the overall process actions and information saved. Each step of this process is detailed in the following sections.

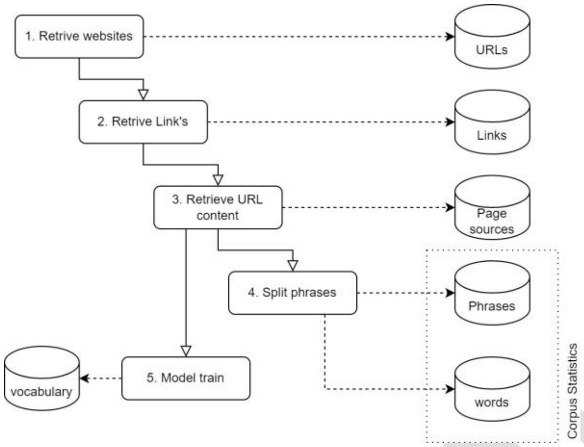

Fig.1. Content retrieve and model train process

.

3.1 Step 1: Retrieve Websites

From the millions of available websites in Arquivo.pt we consider a subset of identified Portuguese periodicals. These websites were denoted by dados.gov [1], a Portuguese Government agency. The list has 1.703 websites, of which we consider 1.535 that have correct links for Arquivo.pt archive. The periodicals are Nationwide or Regional, such as “Público” and “Jornal do Fundão”.

Our main criteria to select these Portuguese periodical websites is to access well-written and structured Portuguese news/text and with no special or restricted subject. Moreover, to consider that some of these periodic have forums been written by followers, that could be written with grammatical errors.

3.2 Step 2: Retrieve Links

When we access the Arquivo.pt’s archive API services, using the collected websites, we receive a list of links to the content archived over time. Also, here we had to make choices regarding the content retrieved:

- Response Status 200: Each of the returned links has metadata associated that indicates the response status of the retrieved page when it was collected, and we only care about the status 200, that is the HTTP status of a correct accessed page. We discarded pages with status like 301, which is a redirect and assumed that the content is only the necessary information for the browser to follow the new destination.
- Mime type: In the HTTP protocol, the response of a request indicates the type of the content returned, and normally a website is composed of several elements

(like style sheets, JavaScript, images, ...) that we didn't want to process. Since Arquivo.pt also has this kind of metadata information, the mime type, we filtered the following types of content: text/plain, text/html, application/pdf, application/rtf, text/rtf and application/msword.

Statistic	Count
Websites processed	1.535
Links	3.487.429
Links Per Mime type	• application/msword: 4.880 • application/pdf: 275.705 • application/rtf: 191 • text/html: 3.197.706 • text/plain: 8.766 • text/rtf: 181

Table 2. Retrieved links statistics

In Table 2 it is possible to see that the main content type retrieved is the expected one, i.e., text/html. There is also some binary content to be processed (like PDFs). Not all of this content will be saved because the corpus doesn't benefit from repeated text content. It is expected to be repeated because if the site/page is archived for two consecutive days (or near) the content is the same.

The list of links that are extracted through the APIs provided by Arquivo.pt, contains more metadata that allows obtaining a certain page in the various moments in which it was captured. Therefore, we can have links where the address is the same, but the moment they were archived is different. All links are stored in a PostgreSQL database for processing.

3.3 Step 3: Retrieve URL Content

Due to the fact that were applied filters to receive URLs with text and that succeeded at the time of the crawling process, it is now necessary to obtain the content. Depending on the type of content that was collected (indicated in the link list metadata) different processing is done:

- text/html or text/plain: the text of the request made to the Arquivo.pt repository (Wayback) is extracted, using a Python Trafilatura framework that can interpret the structure of the obtained text (html) and retrieve only the text that is rendered on the page. At this step, the process is also instructed to look only for the text in Portuguese;

- Other mime types: for these contents, the Apache Tika (Python port) is used. This framework allows extracting text from different formats, including binaries like PDFs.

After extracting the text of the available links, the content is saved in a PostgreSQL database for further processing. Arquivo.pt can retrieve the same page over time (with the same content) but, for our process, there is no advantage in keeping equal content, so a hash of the collected text is generated, which has to be unique in the database, thus avoiding content duplication. A SHA256 hash is calculated and used as a unique constraint in the database. To keep the same encoding for the next steps, is guaranteed to save the extracted content as utf-8.

The original html is also stored in the database so, if necessary, other frameworks for extracting text from html can be used in the future. Table 3 shows statistics of the content retrieved.

Statistic	Count
Unique content text	428.719
Average text size (bytes)	5.240

Table 3. Text capture statistics

In the following, we described the Python frameworks used for this process:

- Trafilatura: [2,12] this library was evaluated as one of the best tools for web scraping, with great performance in retrieving text for web pages. It is used when the URL has a mime type of text/html or text/plain. This framework has also the possibility to define a target language (that in our case was pt).
- Apache Tika: [19,16,15] proven capacity of extracting text from binary documents, is used to process the content when it is found a mime type of: application/pdf, application/rtf, text/rtf or application/msword. This will permit the record of the text for the next processing steps. A Python port of the Apache Tika library that makes Tika available using the Tika REST Server.

3.4 Step 4: Split Phrases

For each content collected (text) it is necessary to extract the sentences that make up the text. In the first phase, the blocks are separated (by indicating the line change) and then in each block, the phrases are extracted. For this sentence extraction, is used the Python framework nltk, which allows this separation into sentences. At this stage and for each sentence, it is checked again if it is in Portuguese. The text as a whole may have been classified as being in Portuguese but, we want a second check at the sentence level. Again, when we are going to record the extracted phrases, there is no gain in repeating phrases (regardless of the source text) and we use again the creation of a phrase hash and this hash must be unique in the database. Therefore, in the background, the database insertion can be denied because there is another phrase with the same hash. Words are also extracted individually from the sentence for:

- Having sentence structure information, indicating the word count, whether classified with stop words. This information per sentence will also allow us to choose sentences for training the model according to its dimension.
- Recording the unique words found. – Statistics of the words found.

Statistic	Count
Unique phrases	16.198.437
Average phrase size (bytes)	68
Average words	6
Average stop words	4

Table 4. Phrase capture statistics

In Table 4 it is possible to see from the collected text, how many unique phrases are retrieved. The phrases also create a unique constraint in the database (hash created on the phrase lower case) to guarantee that there are no replicated phrases.

3.5 Step 5: Model Training

After collecting and building the corpus, it is needed to create a vocabulary that will be used in the training process. The vocabulary defines a set of tokens, collected from the corpus or special tokens like “[UNK]”, “[PAD]”, “[CLS]”, “[SEP]” or “[MASK]”. BERT can receive two sentences, so at the beginning always receive the “CLS” token and to separate two sentences is used the “SEP” token. “MASK” is a special token used to hide a token and let the algorithm try to find the token that best suits in the context. Since BERT receives a fixed sentence size, the special token “PAD” is used to fill the remain tokens. When a token don’t appear in the vocabulary, the special token “UNK” is used to represent it.

- Normalization: used BertNormalizer: this pre-tokenizer splits tokens on spaces, and also on punctuation. Each occurrence of a punctuation character will be treated separately;
- Pre-Tokenization: used BertPreTokenizer: that takes care of normalizing raw text before giving it to a Bert model. This includes cleaning the text, handling accents, Chinese chars and lowercasing;
- Model: used the WordPiece Tokenization Model, that is the tokenization algorithm Google developed to pretrain BERT;

- Post Processor: define the rules for processing inputs with one or two sentences (rules for using “CLS” and “SEP” tokens).

The code was inspired by Hugging Face [9] community. We created vocabularies with 20000, 25000, 30000, 35000, 40000, 45000 and 50000 tokens with the proposal of future evaluations on the results and on the compute performance based on different vocabularies sizes. For the BERT training, we used the vocabulary of size 20000.

For the model training, the corpus was random separated to have a train and a validation corpus (10% of the original corpus). For this first approach, the model was trained for a Masked Language Model – predicting masked tokens in new sentences. The config was loaded from a pretrained “bert-base-cased”: 12-layer, 768-hidden, 12-heads, 110M parameters. The training took around 40 hours of computation in a high performance computing infrastructure. This infrastructure belongs to the University of Évora and is a supercomputer (Vision [21]) made by 2 compute nodes and a management node. Each compute node is an NVIDIA DGX A100 system with the following specifications:

- 8x NVIDIA A100 GPU (40 GB each GPU)
- 2x AMD Rome 7742 (64 cores, 128 threads each CPU)
- 1 TB RAM
- 8x Single-Port Mellanox ConnectX-6 VPI 200Gb/s HDR InfiniBand
- 1x Dual-Port Mellanox ConnectX-6 VPI, 10/25/50/100/200Gb/s Ethernet
- 5 petaFLOPS AI / 10 petaOPS INT 8

4 Conclusions and Future Work

The training procedure was defined and tested to generate a first (that we know) Portuguese-European language model based on BERT. The process was run entirely with success and the future work will be focused on the optimization and evaluation of the model quality and compute performance. The final model and code will be shared with the community for investigation proposes. This work was done using a subset of information retrieved from Arquivo.pt and is expected, when using more content, to have better results in the language model quality.

References

1. AMA- Agência para a Modernização Administrativa, I. P.: Dados.gov, https://dados.gov.pt/pt/datasets/publicacoes-periodicas-portuguesasjornais-e-revistas-websites-e-historico-de-versoes-no-arquivo-pt/
2. Barbaresi, A.: Trafilatura: A Web Scraping Library and Command-Line Tool for Text Discovery and Extraction. In: Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations. pp. 122–131.

Association for Computational Linguistics (2021), https://aclanthology.org/2021.acl-demo.15

3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5, 135–146 (2017), https://aclanthology.org/Q17-1010
4. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 1877–1901.

   Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/ file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota

   (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology. org/N19-1423
6. Diouf, R., Sarr, E., Sall, O., Birregah, B., Bousso, M., Mbaye, S.: Web scraping: State-of-the-art and areas of application. pp. 6040–6042 (12 2019). https://doi. org/10.1109/BigData47090.2019.9005594
7. FCT - Fundação para a Ciência e Tecnologia: Arquivo.pt, https://www.arquivo. pt
8. Gomes, D., Nogueira, A., Miranda, J., Costa, M.: Introducing the portuguese web archive initiative. In: 8th international Web archiving workshop. Springer (2009)
9. Hugging Face: Hugging face, https://huggingface.co/
10. Joshi, V., Peters, M., Hopkins, M.: Extending a parser to distant domains using a few dozen partially annotated examples. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1190–1199. Association for Computational Linguistics, Melbourne, Australia (Jul 2018). https://doi.org/10.18653/v1/P18-1110, https: //aclanthology.org/P18-1110
11. Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., Allauzen, A., Crabbé, B., Besacier, L., Schwab, D.: Flaubert: Unsupervised language model pre-training for french. CoRR abs/1912.05372 (2019), http://arxiv.org/abs/ 1912.05372
12. Lejeune, G., Barbaresi, A.: Bien choisir son outil d’extraction de contenu à partir du web. In: 6e conférence conjointe Journées d’Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 4: Démonstrations et résumés d’articles internationaux. pp. 46–49. ATALA; AFCP (2020)
13. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized BERT pretraining

   approach. CoRR abs/1907.11692 (2019), http://arxiv.org/abs/1907.11692
14. Martin, L., Muller, B., Ortiz Suárez, P.J., Dupont, Y., Romary, L., de la Clergerie, É., Seddah, D., Sagot, B.: CamemBERT: a tasty French language model. In: Proceedings of the 58th Annual Meeting of the Association for Computational

Linguistics. pp. 7203–7219. Association for Computational Linguistics, Online (Jul 2020), https://www.aclweb.org/anthology/2020.acl-main.645

15. Mattmann, C.A., Zitting, J.L.: Tika in action (2012)
16. McCandless, M., Hatcher, E., Gospodnetić, O., Gospodnetić, O.: Lucene in action, vol. 2. Manning Greenwich (2010)
17. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language Models are Unsupervised Multitask Learners (2019), https://openai.com/blog/ better-language-models/
18. Souza, F., Nogueira, R., Lotufo, R.: Bertimbau: Pretrained bert models for brazilian portuguese. In: Intelligent Systems: 9th Brazilian Conference, BRACIS 2020, Rio Grande, Brazil, October 20–23, 2020, Proceedings, Part I. p. 403–417. Springer-Verlag, Berlin, Heidelberg (2020). https://doi.org/10.1007/978-3-030-61377-8_28
19. The Apache Software Foundation: Apache tika, https://tika.apache.org/
20. Tripathy, J.K., Sethuraman, S.C., Cruz, M.V., Namburu, A., P., M., R., N.K., S, S.I., Vijayakumar, V.: Comprehensive analysis of embeddings and pre-training in nlp. Comput. Sci. Rev. 42(C) (nov 2021). https://doi.org/10.1016/j.cosrev.2021.100433, https://doi.org/10.1016/j.cosrev.2021.100433
21. Universidade de Évora: Vision lab, https://vision.uevora.pt/
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf