# Pretained SPLADE V2 Sparse Vector using BERTimbau Model on the mMARCO-pt dataset

Thiago Coelho Vieira

FEEC, UNICAMP, Brazil

June 2023

Abstract

This study analyzes the information retrieval tasks performed by the SPLADE V2 Sparse Vector [1] pretrained using the BERTimbau [2] model on the mMARCO-pt [3] Dataset. The MRobust04 [4] dataset was used for the evaluation, with the main focus being on evaluating the nDCG@20 statistic. It is crucial to remember that the acquired results were unsatisfactory, possibly as a result of an issue with the creation of the training validation set that is still under investigation. The pipeline was tested and examined despite running into a number of infrastructure issues on Google Colab. This assignment provided an excellent opportunity to learn and get significant experience by demonstrating how to understand a codebase [5] from an article. The source code of this work is available at https://github.com/tcvieira/IA368-DD-012023/tree/main/final-project/notebooks and slides at https://bit.ly/3puS1Ri

## 1 Introduction

There aren't many effective sparse representation models for non-English languages, like SPLADE. Although significant progress has been made in the development of information retrieval models for English, other languages have not seen the same level of advancement, with the exception of some multi language models.

Dense representations and sparse representations are two approaches commonly used in transformer-based models for information retrieval, each offering its own set of advantages and disadvantages.

Dense representations, exemplified by models like COLBERT [6] (Contextualized Late Interaction over BERT), offer the advantage of capturing intricate semantic relationships and subtle contextual nuances. COLBERT leverages dense embeddings to encode information in a continuous vector space, enabling it to effectively model complex language patterns for information retrieval tasks. This approach enhances the retrieval performance by capturing fine-grained relationships and contextual information, leading to more accurate and relevant search results. Dense representations like COLBERT also support efficient computation, transfer learning, and can handle large-scale datasets effectively. However, one limitation is the potential lack of interpretability compared to sparse representations, as the continuous nature of dense embeddings makes it challenging to directly analyze the importance of individual components within the vector space.

Sparse representations, such as SPLADE [1], provide interpretability and enhance the understanding of learned features. They exhibit sparsity, leading to memory-efficient storage and faster computations. Sparse embeddings are particularly useful for analyzing the importance of different components. However, they face challenges with high dimensionality when dealing with large vocabularies. Sparse representations may require more extensive training and careful tuning to achieve optimal performance, and they may struggle to capture fine-grained relationships and contextual information as effectively as dense representations.

The SPLADE [1], a a bi-encoder BERT-based model was developed as a novel method for learning sparse representations for documents and queries that might inherit from the advantageous characteristics of bag-of-words models, including the effectiveness of inverted indexes and the accurate matching of terms. When compared to cutting-edge dense and sparse methods, the model offers highly sparse representations and competitive outcomes.

SPLADE [1] is a transformer-based retrieval model that uses token pooling over the tokens and the Masked Language Modeling (MLM) head to represent documents and queries. Since only a few dimensions are involved by the SPLADE model for a given document (or query), retrieval can be done sparsely as a result of the model being trained by jointly maximizing ranking and regularization losses.

SPLADE is trained by optimizing a contrastive loss (InfoNCE) with hard negatives (from BM25) and in-batch negatives under the restrictions of regularization, which seeks to minimize the expected number of floating-point operations during retrieval. All queries, positive_passages, and negative_passages are embedded into the vector space. The matching (query_i, positive_passage_i) should be close, while there should be a large distance between a query and all other (positive/negative) passages from all other triplets in a batch.

Some information can be found in the slides created for assignment 07 of the course in [8]. In Figure 1 shows a simple overview of SPLADE one stage retrieval pipeline.

**TL;DR** "SPLADE is a neural retrieval model which learns query/document sparse expansion via the BERT MLM head and sparse regularization. Sparse representations benefit from several advantages compared to dense approaches: efficient use of inverted index, explicit lexical match, interpretability".

Fig

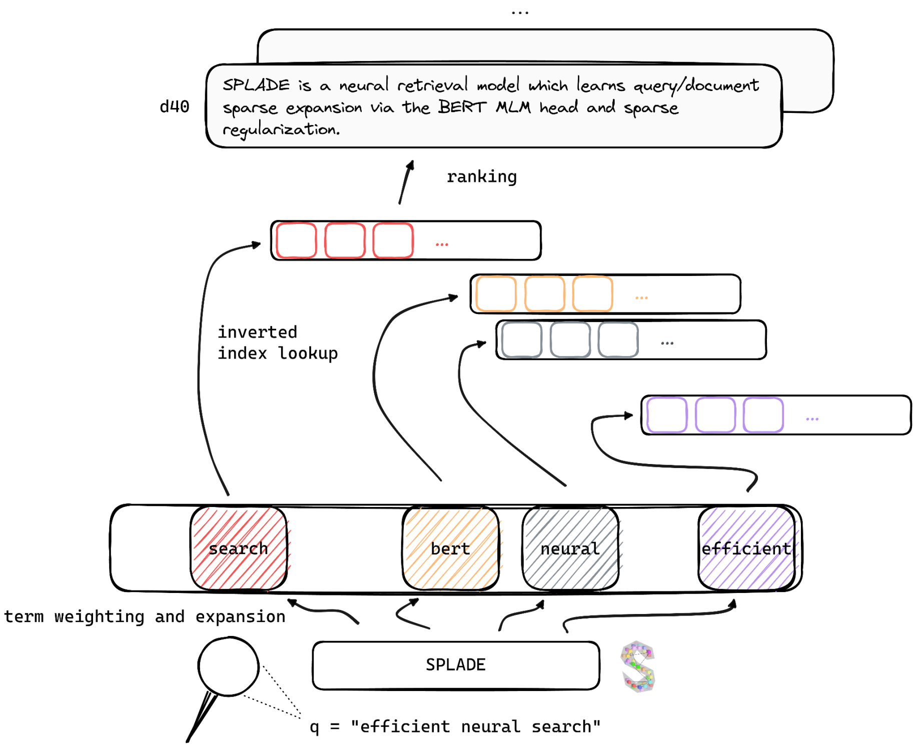
...
d40
SPLADE is a neural retrieval model which learns query/document sparse expansion via the BERT MLM head and sparse regularization.
ranking
...
...
...
inverted
index lookup
...
search
bert
neural
efficient
term weighting and expansion
SPLADE
q = "efficient neural search"

Figure 1 - SPLADE first stage retrieval pipeline example.

## 2 Methodology

To conduct this study, the first step was to prepare the training dataset by utilizing the mMARCO [3] dataset. Additionally, the evaluation dataset, MRobust [4], was obtained and prepared for performance assessment.

Once the training dataset was ready, the SPLADE V2 training pipeline from the Naver SPLADE codebase available on GitHub was employed. This pipeline encompasses various components, including data loading, preprocessing, training, and evaluation. By leveraging this pipeline, we were able to train the SPLADE V2 model on the prepared dataset. Parameters and hyperparameters were appropriately set, considering the specific requirements and objectives of the experiment.

The performance of the trained SPLADE V2 model was evaluated after the training process. The MRobust-PT [4] dataset, which ran as the evaluation set, was used to test the model for this purpose on the following metrics: nDCG@10, nDCG@20, MRR@10 and R@1000.

## 3 Data set

In this section we present the mMarco-pt dataset and the mRobust dataset, which were used for pretrain SPLADE and evaluation, respectively, are presented in this section.

**mMARCO** [3] is a multilingual version of the MS MARCO passage ranking dataset. For this experiments we worked with the collections, queries and qrels for the portuguese language.

**mRobust** [4] is a multilingual version of the TREC 2004 Robust passage ranking dataset [9].

## 4 Experiments

We ran 3 experiments. Two were for testing propurse with 1K triplets and 10K triples with reduced validation queries and collection. Other one with 4M triplets.

Table 1 shows the configuration for the training, indexing and evaluation on the 4M triplets training.

```
config:
  lr: 2.0e-05
  seed: 123
  gradient_accumulation_steps: 1
  weight_decay: 0.01
  validation_metrics:
  - MRR@10
  - recall@100
  - recall@200
  - recall@500
  pretrained_no_yamlconfig: false
  nb_iterations: 150000
  train_batch_size: 64
  eval_batch_size: 64
  index_retrieve_batch_size: 64
  record_frequency: 10000
  train_monitoring_freq: 500
  warmup_steps: 6000
  max_length: 256
  fp16: false
  matching_type: splade
  monitoring_ckpt: MRR@10
  loss: InBatchPairwiseNLL
  augment_pair: in_batch_negatives
  regularizer:
    FLOPS:
      lambda_q: 0.0005
      lambda_d: 0.0003
      T: 3
      targeted_rep: rep
      reg: FLOPS
  tokenizer_type: neuralmind/bert-base-portuguese-cased
  top_k: 1000
  threshold: 0
  eval_metric:
  - - mrr_10
    - recall
    - ndcg_cut_20
    - recall@1000
  checkpoint_dir: /content/experiments/splade_mmarco_pt_4M/checkpoint
  index_dir: /content/experiments/splade_mmarco_pt_4M/index
  out_dir: /content/experiments/splade_mmarco_pt_4M/out
data:
  type: triplets
  TRAIN_DATA_DIR: data/mmarco-pt-4M/triplets
  VALIDATION_SIZE_FOR_LOSS: 60000
  VALIDATION_FULL_RANKING:
    D_COLLECTION_PATH: data/mmarco-pt/val_collection
    Q_COLLECTION_PATH: data/mmarco-pt/val_queries
    QREL_PATH: data/mmarco-pt/qrel/qrel.json
    TOP_K: 500
  COLLECTION_PATH: data/mmarco-pt/full_collection
  Q_COLLECTION_PATH:
  - data/mmarco-pt/dev_queries
  EVAL_QREL_PATH:
  - data/mmarco-pt/dev_qrel.json
  flops_queries: data/mmarco-pt/dev_queries/
init_dict:
  model_type_or_dir: neuralmind/bert-base-portuguese-cased
  model_type_or_dir_q: null
  freeze_d_model: 0
  agg: max
  fp16: false
```

Table 1 - config.yaml for training experiment

The training metrics results are shown on Table 2.

```
iter,batch_ranking_loss
10500,0.6054369211196899
11000,0.6265198588371277
20000,0.4764997959136963
30000,0.5922725200653076
40000,0.40848782658576965
50000,0.39034831523895264
70000,0.45360419154167175
80000,0.48550644516944885
90000,0.42392784357070923
100000,0.4602883756160736
110000,0.5774438381195068
120000,0.4405728876590729
150000,0.4526127278804779
```

Table 2 - training metrics

The training metrics result on the last iteration is shown on Table 3.

| **iteration** | **loss** | **MRR@10** | **R@100** | **R@100** | **R@500** |
|---|---|---|---|---|---|
| 150000 | 0.4584 | 0.0942 | 0.3131 | 0.3680 | 0.4530 |

Table 3 - training metrics last iteration

The evaluation metrics result is shown on Table 4.

| **nDCG@10** | **nDCG@20** | **MRR@10** | **R@1000** |
|---|---|---|---|
| 0.0538 | 0.0483 | 0.1371 | 0.1168 |

Table 4 - final evaluation metrics

No additional experiments were conducted and the issue with the validation dataset during training was not addressed due to time and cost limitations.

## 5 Conclusion

The performance of the SPLADE V2 Sparse Vector pretrained using the BERTimbau model on the mMARCO-pt Dataset for information retrieval tasks was examined in this research. The MRobust04 dataset was used to evaluate the nDCG@20 metric. The pipeline was successfully tested and analyzed on Google Colab despite running into a number of infrastructure issues.

It is important to note, nevertheless, that the results were not satisfactory. This may be a result of an issue with the way the validation set was created for training. We'll run more tests to confirm and resolve this issue.

Yet, this project gave us the opportunity to explore a codebase that was described in an article, which improved our comprehension of the implementation aspects. It was a useful way to learn about and develop expertise in the subject of multilingual information retrieval.

Overall, despite the setbacks encountered, this project has provided valuable insights and a solid foundation for future research and development in the field of multilingual information retrieval. By learning from the limitations and refining the methodologies, we can work towards developing more effective models for information retrieval tasks in diverse languages.

## 6 Future Work

As future work, aiming to enhance the performance of the model going forward, more analysis and improvement are required. It will be essential to address the problems with the validation set building if we want to get better outcomes. To overcome the obstacles encountered during the testing process, additional infrastructure resources and tools should also be taken into consideration.

## References

[1] Formal, T., Lassance, C., Piwowarski, B., & Clinchant, S. (2021). SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval. ArXiv, abs/2109.10086.

[2] Souza, F., Nogueira, R., & Lotufo, R.D. (2020). BERTimbau: Pretrained BERT Models for Brazilian Portuguese. Brazilian Conference on Intelligent Systems.

[3] Bonifacio, L.H., Campiotti, I., Lotufo, R.D., & Nogueira, R. (2021).

mMARCO: A Multilingual Version of MS MARCO Passage Ranking Dataset. ArXiv, abs/2108.13897.

[4] Jeronymo, V., Nascimento, M., Lotufo, R.D., & Nogueira, R. (2022). mRobust04: A Multilingual Version of the TREC Robust 2004 Benchmark. ArXiv, abs/2209.13738.

[5] SPLADE codebase https://github.com/naver/splade

[6] Khattab, O., & Zaharia, M.A. (2020). ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.

[7] https://archive.pinecone.io/learn/splade/

[8] https://github.com/tcvieira/IA368-DD-012023/blob/main/assingments/07-Sparse-Vectors-SPLADE/article-slides.pdf

[9] TREC 2004 Robust Track Guidelines https://trec.nist.gov/data/robust/04.guidelines.html