# Training and Evaluating SPLADEv2 for Portuguese

Technical Report

Leonardo Avila and Monique Monteiro

FEEC, UNICAMP, Brazil

June 2023

**Abstract**

In neural Information Retrieval (IR), ongoing research is directed toward improving the first retriever in ranking pipelines. Learning dense embeddings to conduct retrieval using efficient approximate nearest neighbors methods works well. Meanwhile, there has been a growing interest in learning sparse representations for documents and queries that could inherit from the desirable properties of bag-of-words models, such as the exact matching of terms and the efficiency of inverted indexes. The SPLADE model provides highly sparse representations and competitive results concerning state-of-the-art dense and sparse approaches. This technical report builds on SPLADEv2 to train and evaluate a single-stage sparse retriever for the Portuguese language.

## 1 Introduction

According to Formal et al., [3], in neural Information Retrieval (IR), ongoing research is directed toward improving the first retriever in ranking pipelines. Learning dense embeddings to conduct retrieval using efficient approximate nearest neighbors methods works well. Still, according to the same authors, because of strict efficiency requirements, these models have initially been used as re-rankers in a two-stage ranking pipeline, where first-stage retrieval – or candidate generation – is conducted with bag-of-words models (e.g., BM25) that rely on inverted indexes. Further, while BOW models remain strong baselines, they suffer from the long-standing vocabulary mismatch problem, where relevant documents might not contain terms that appear in the query. Thus, there have been attempts to substitute standard BOW approaches with learned (neural) rankers. Designing such models poses several challenges regarding efficiency and scalability. Therefore, there is a need for methods where most of the computation can be done offline, and online inference is fast. Dense retrieval with approximate nearest neighbors search has shown impressive results. However, it can still benefit from BOW models (e.g., by combining both signals) due to the absence of explicit term matching. Hence, a strong interest in learning sparse representations for queries and documents has grown. By doing so, models can inherit the desirable properties of BOW models, like the exact match of (possibly latent) terms, the efficiency of inverted indexes, and interpretability. Additionally, by modeling implicit or explicit (latent, contextualized) expansion mechanisms – similar to standard expansion models in IR – these models can reduce the vocabulary mismatch.

This technical report builds on SPLADEv2 [3] to train and evaluate a single-stage sparse retriever for the Portuguese language. It is organized in the following way: 2 introduces the techniques used to conduct the work, while 3 and 4 present the detailed datasets and experiments, respectively. Source code is available at `https://github.com/monilouise/unicamp-P_IA368DD_2023S1/tree/main/Final_project`.

## 2 Methodology

To adapt SPLADEv2 model to the Portuguese language, our first choice for development methodology was to reuse SPLADE codebase, an open-source software that supports a multitude of configuration options. In particular, SPLADEv2 codebase enables us to choose the text encoder neural network, the training hyperparameters, the validation strategy, loss functions, etc.

However, some choices were not readily available as simple configurations. Some examples include special optimizations in the data loading process and adjustments for different encoder networks. In these situations, we had to create different forks for the software, including the needed modifications[1].

As the encoder model, we opted for BERTimbau base [9], a BERT-based model already finetuned to the Portuguese language. We use the same encoder for both queries and documents.

## 3 Data set

We used the following datasets:

- mMARCO [1]: a multilingual version of the MS MARCO [6] passage ranking dataset comprising 13 languages, created using machine translation. mMARCO was used for training.
- mRobust [5]: a multilingual version of Robust04 that was translated to 8 languages using Google Translate. mRobust was used for evaluating the trained model.

[1] The following forks were created: `https://github.com/leobavila/splade` (optimizations in the data loading process) and `https://github.com/monilouise/splade` (adjustments for different encoder networks).

As well as using mMARCO for training the model, we also used a part of it to periodically evaluate information retrieval metrics such as MRR@10 on an in-memory index created and updated during training. This data split is the same one used in the original SPLADEv2 model (training and evaluated on MSCARCO), differing only in the language used (Portuguese here).

A summary of the datasets and its sizes is shown in Figures 1 and 2.

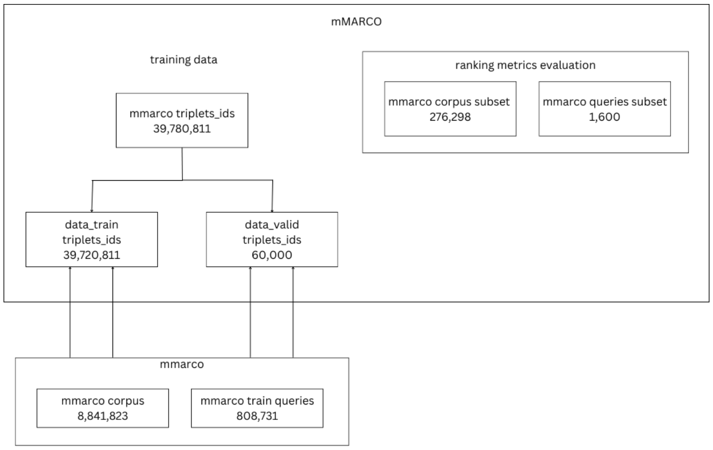

Figure 1: mMARCO Dataset.

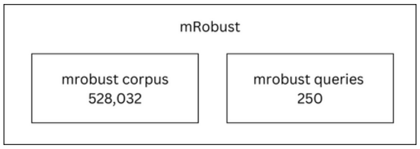

Figure 2: mRobust Dataset.

# 4 Experiments

We ran a total of about eight experiments, including:

1. First training on 10 million triplets, to get the initial results with a reduced version of the dataset - this reduction was initially necessary because of a flaw in SPLADEv2 codebase which caused the preloading of all the triplets' queries and documents texts in memory, including repeated document texts. As negative examples, we initially used the default configuration of only triplets negatives. Finally, we used truncation for long texts (default behavior in SPLADE), using only 256 characters (maximum length = 256).

2. Training on the full dataset (39 million triplets), but with a correction to the above-cited data preloading problem. Now, we preload only queries and corpus identifiers into memory, while the full corpora of documents and queries are always loaded only once and kept in memory. It led to significant memory savings because the number of triplets is far more extensive than the number of queries and documents. As negative examples, we used the default configuration of only triplets negatives. Maximum length = 256.

3. Same as 2, except for maximum length = 384.

4. Use of a different encoder network - PTT5 [2], a T5 [8] version trained on Brazilian Portuguese data. Here, we used only the last hidden states of the encoder part of the model. Same hyperparameters (e.g., learning rate, regularization, maximum length = 256, negative samples) as the previous experiments.

5. Use of PTT5 encoder with higher learning rates. Maximum length = 256.

6. Use of in-batch negatives with adjustments to default regularization hyperparameters and maximum length = 384.

We noticed a very slow convergence with models 4 and 5, so we did not conclude their training. Due to cost and time constraints, such as the long feedback time of 20 hours per training, we did not explore further T5 integration options. We hypothesize that SPLADE original configuration was biased toward MLM (masked language model) transformers: Formal et al. [3] refer explicitly to MLM transformers in their formulation of vocabulary tokens weights, as we will see shortly. However, we still intend to conduct further research on integrating T5-family models into SPLADE architecture suitably.

We obtained the best results with the last option: in-batch negatives are known to lead to better retrieval performance. Also, we reduced the FLOPS regularization [7] hyperparameters $\lambda q$ and $\lambda d$ from the default values of $5 \times 10^{-4}$ to $3 \times 10^{-4}$ and $3 \times 10^{-4}$ to $10^{-4}$, respectively.

The following formula is used to calculate FLOPS regularization loss, according to Formal et al. [3]:

$$l_{FLOPS} = \sum_{j \in V} (\frac{1}{N} \sum_{i=1}^{N} {w_j}^{d_i})^2 \tag{1}$$

where $V$ is the vocabulary size, $N$ is the sequence length, and $w_j$ means the weight for each vocabulary token, obtained by summing importance predictors over the input sequence tokens after applying a log-saturation effect:

$$w_j = \sum_{i \in t} log(1 + ReLU(w_{ij})) \tag{2}$$

given the importance $w_{ij}$ of the vocabulary token $j$ for a token $i$ (from the input sequence) :

$$w_{ij} = transform(h_i)^T E_j + b_j \tag{3}$$

where $E_j$ denotes the BERT input embedding for token $j$ , $b_j$ is a token-level bias, and transform(.) is a linear layer. As Eq. (3) is equivalent to the MLM prediction, it can be initialized from a pre-trained MLM model (e.g., BERTimbau).

The final loss is calculated according to

$$L = L_{rank-IBN} + \lambda_q l^q_{FLOPS} + \lambda_d l^d_{FLOPS} \tag{4}$$

given

$$L_{rank-IBN} = -log \frac{e^{s(q_i,d_i^+)}}{e^{s(q_i,d_i^+} + e^{s(q_i,d_i^-}) + \sum j e^{s(q_i,d_{i,j}^-)}} \tag{5}$$

where $s(q, d)$ denote the ranking score obtained via the dot product between $q$ and $d$ representations from Eq. (2). Given the query $q_i$ in a batch, a positive document $d_i^+$, a negative document $d_i^-$ and a set of negative documents in the batch (positive documents from other queries) ($d_{i,j}^-$), we consider a contrastive loss, which can be interpreted as the maximization of the probability of the document $d_i^+$ being relevant among the documents $d_i^+$, $d_i^-$ and $d_{i,j}^-$.

Also, to mitigate the contribution of the regularizer at the early stages of training, we follow Formal et al. [3] and use the same scheduler for $\lambda$, quadratically increasing it at each training iteration until a given step $T$, from which it remains constant. During experiments 1 to 5, we used T=3, but then we changed it to T=50,000 at experiment 6, following the default configurations used in SPLADEv2 for MSMARCO.

We suspect an extreme vector sparsity led to initially worse results, and these hyperparameters should be carefully tuned to the dataset, as the performance for SPLADE depends on the regularization strength, according to Format et al. [3]. So we intend to conduct further regularization tuning in the future.

The baseline results with BM25 algorithm are shown in Table 1, while our results are shown in Table 2.

Table 1: Basline evaluation on mRobust passage retrieval with BM25

| **Settings** | **ndcg@10** | **ndcg@20** | **mrr@10** |
|---|---|---|---|
| $k1 = 0.82, b = 0.68$ | 0.4098 | 0.3893 | 0.6690 |

At the time of writing this report, we were running the same experiments on mMARCO "dev" dataset - a translation of the same split for MSMARCO used in SPLADEv2 experiments.

Table 2: Evaluation on mRobust passage retrieval

| **Triplets** | **Settings** | **max length** | **ndcg@10** | **ndcg@20** | **mrr@10** |
|---|---|---|---|---|---|
| 10 MM | triplet negatives | 256 | 0.2797 | 0.2581 | 0.5183 |
| 39 MM | triplet negatives | 256 | 0.3051 | 0.2740 | 0.5529 |
| 39 MM | triplet negatives | 384 | 0.3149 | 0.2862 | 0.5885 |
| 39 MM | in-batch negatives | 384 | 0.3295 | 0.2977 | 0.5929 |

# 5 Conclusion

In this study, we trained SPLADEv2 model on mMARCO, a Portuguese automatically translated version of MSMARCO, and evaluated it on mRobust. We did not surpass BM25 performance in our model's first stable version. Still, we identified consistent trends of improvements in all metrics in positive correlation with three aspects: 1) the context's length, 2) the use of in-batch negative samples, and 3) regularizer hyperparameters. So, we consider them as promising routes for improvements.

# 6 Future Work

Our current model version has a vast space for improvements, so we expect to conduct further experiments and investigations.

In particular, as said in Section 4, we intend to publish model performance on the same split of MS MARCO used in SPLADEv2 in a future version of this report.

Secondly, as mRobust has extensive documents, we intend to break them into sliding windows of passages.

We also intend to check model performance with a maximum context length of 512 tokens (the maximum context supported by BERTimbau) and train it for more iterations. For example, the SPLADEv2 model was trained on four A100 GPUs, while our experiments were trained on a single V100/A100 GPU each.

Model distillation [4] is another improvement route, as SPLADEv2 authors strongly focus on this area.

We also intend to tune regularization and further research other ways of integrating T5-based models.

Finally, we will compare sparse models' performance with recent OpenAI embedding models such as text-embedding-ada-002.

# References

[1] Luiz Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. mmarco: A multilingual version of the ms marco passage ranking dataset, 2022.

[2] Diedre Carmo, Marcos Piau, Israel Campiotti, Rodrigo Frassetto Nogueira, and Roberto de Alencar Lotufo. PTT5: pretraining and validating the T5 model on brazilian portuguese data. *CoRR*, abs/2008.09144, 2020.

[3] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2288–2292, New York, NY, USA, 2021. Association for Computing Machinery.

[4] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. Improving efficient neural ranking models with cross-architecture knowledge distillation. *CoRR*, abs/2010.02666, 2020.

[5] Vitor Jeronymo, Mauricio Nascimento, Roberto de Alencar Lotufo, and Rodrigo Frassetto Nogueira. mrobust04: A multilingual version of the TREC robust 2004 benchmark. *CoRR*, abs/2209.13738, 2022.

[6] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. In Tarek Richard Besold, Antoine Bordes, Artur S. d'Avila Garcez, and Greg Wayne, editors, *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.

[7] Biswajit Paria, Chih-Kuan Yeh, Ian E. H. Yen, Ning Xu, Pradeep Ravikumar, and Barnabás Póczos. Minimizing flops to learn efficient sparse representations, 2020.

[8] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020.

[9] Fábio Souza, Rodrigo Frassetto Nogueira, and Roberto de Alencar Lotufo. Bertimbau: Pretrained BERT models for brazilian portuguese. In Ricardo Cerri and Ronaldo C. Prati, editors, *Intelligent Systems - 9th Brazilian Conference, BRACIS 2020, Rio Grande, Brazil, October 20-23, 2020, Proceedings, Part I*, volume 12319 of *Lecture Notes in Computer Science*, pages 403–417. Springer, 2020.