# Comparative Analysis of Classical and Deep Algorithms for Text Clustering in Brazilian Portuguese

**Paulo V. Mourão**[1], **Marcela P. Pessoa**[1], **Oswald M. Ekwoge**[2], **Marcelo E. Anjos**[2]

[1] Escola Superior de Tecnologia – Universidade do Estado do Amazonas (UEA)
[2] SiDi

{pvmdc.snf22, msppessoa}@uea.edu.br
{oswald.me, marcelo.e}@sidi.org.br

***Abstract.*** *This work compares different combinations of text embedding models and clustering algorithms applied to Portuguese-language texts. Three datasets were used (poems, Reddit, and product reviews), evaluating models such as BERTimbau and ST5, combined with classic algorithms and the deep method DEC. Using accuracy, V-Measure, and ARI as metrics, results show that BERTimbau performs better on formal texts, while ST5 excels in informal content. DEC outperformed others only on the largest dataset (product reviews), highlighting the potential of deep clustering approaches for Portuguese text analysis.*

## 1. Introduction

Clustering is an unsupervised learning technique widely used to uncover patterns and structure large volumes of data without the need for labels. In Natural Language Processing (NLP), this approach has important applications in tasks such as document organization, exploratory analysis, and topic modeling.

With the advancement of vector-based text representation techniques, it has become common to use semantic *embeddings* as the foundation for clustering. These *embeddings* allow texts to be represented in vector spaces where proximity reflects semantic similarity. Tools such as *BERTopic* [Grootendorst 2022] have popularized the combined use of *embeddings* and clustering for the identification of latent topics in textual corpora.

In addition to classical methods like *K-Means*, new **clustering** techniques based on deep learning have gained prominence, such as *Deep Embedded Clustering* (DEC) [Xie et al. 2016] and its variations, which integrate representation learning and clustering in a joint manner, optimizing both processes simultaneously. These methods aim to overcome limitations of traditional algorithms, such as sensitivity to initialization and difficulty handling high-dimensional data.

Despite advancements in the field, few studies have evaluated modern clustering techniques applied to Portuguese. This work stands out by comparing monolingual and multilingual vector representation models in textual clustering tasks. The representations were combined with both classical and deep algorithms, such as *DEC*, and evaluated across three datasets. The results show that monolingual models perform better on formal texts, while multilingual models achieve better performance on informal language. The *DEC* approach, in turn, was effective only in scenarios with larger data volumes.

## 2. Related Works

Several recent studies have explored the combination of language models with clustering techniques to meet the growing demand for organizing and analyzing textual data.

In the context of Brazilian Portuguese, the study by [Borges 2025] stands out, having evaluated the performance of three classical clustering algorithms — *K-Means*, *Single Linkage* e *Gaussian Mixture Model* (GMM) — applied to a subset of real news articles from the *Fake.Br*. The *BERTimbau* model [Souza et al. 2020] was used for text vectorization, allowing for a direct comparison of the algorithms based on external metrics such as *F1-score*, *Rand Index* e *Jaccard*. The results showed that the Single Linkage algorithm outperformed the others in several criteria, demonstrating better suitability among the evaluated methods.

Expanding the analysis to more recent techniques, [Subakti et al. 2022] investigated the use of *Deep Embedded Clustering* (DEC) and its enhanced variant, *Improved DEC* (IDEC) [Guo et al. 2017], comparing them with traditional methods. The authors used *embeddings* generated by *BERT* models and evaluated the impact of different normalization and pooling strategies on clustering performance across multiple datasets. The study concluded that BERT-based *embeddings* outperformed *TF-IDF* in 28 out of 36 evaluated metrics, and that the combination of *mean pooling* with standard normalization was particularly effective for the *DEC* and *IDEC* models.

Internationally, the work of [Wehrli et al. 2024] proposed a *benchmark* for German text clustering, based on the structure of the *Massive Text Embedding Benchmark* (MTEB). The study evaluated various monolingual and multilingual models along with algorithms such as *MiniBatch K-Means*, *Agglomerative Clustering*, *HDBSCAN* and *DBSTREAM*. The results indicated that well-trained monolingual embeddings, such as *GBERT*, deliver competitive performance compared to large models like *ST5-xxl* [Ni et al. 2021]. The *benchmark* reinforces the importance of language-specific evaluations, as multilingual models do not always match the performance of dedicated models.

Additionally, [Keraghel et al. 2024] investigated the influence of large language models (LLMs) — such as *Mistral-7B*, *LLaMA-2-13B* and *GPT* — on embedding quality for clustering. The study concluded that embeddings derived from LLMs, especially *GPT* (ada-002), produce more coherent groupings even on imbalanced datasets, while smaller models like *MiniLM* also demonstrated robustness with lower computational cost.

These studies highlight a clear evolution in textual clustering approaches, particularly in the combination of contextualized embeddings and deep clustering methods. However, there remains a significant gap in the literature focused on Brazilian Portuguese, especially regarding the use of modern clustering techniques. Most studies still rely on classical algorithms without exploring the potential of newer approaches. This work aims to bridge that gap through a systematic evaluation of contemporary techniques applied to Brazilian Portuguese texts.

## 3. Methodology

This section describes the methodology adopted to evaluate the performance of different combinations of vector representation techniques and clustering algorithms applied to texts in Brazilian Portuguese. The proposed approach is structured as a pipeline composed of four main stages: (i) datasets and preprocessing; (ii) *embeddings* and feature extraction; (iii) clustering algorithms; and (iv) evaluation metrics. Figure 1 presents a chart summarizing the pipeline. For better visualization, we have separated embeddings and feature extraction into distinct components.

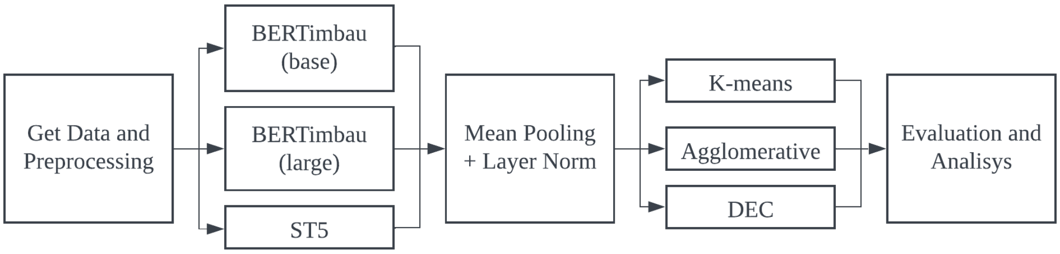

**Figure 1. *Flowchart* Summarizing the Methodology**

### 3.1. Datasets and Preprocessing

For this analysis, three datasets were selected from repositories available on Kaggle. We intentionally varied the size and language of the datasets to observe how different combinations of algorithms and embedding models perform under diverse conditions.

- **Poems[1]:** A total of 647 poems by six poets representing different national lyrical movements: Álvares de Azevedo, Castro Alves, Olavo Bilac, Augusto dos Anjos, Carlos Drummond de Andrade, and Paulo Leminski.
- **Reddit[2]:** A collection of Portuguese-language posts from 24 thematic communities on Reddit, covering informal language and a wide range of topics. The five communities with the highest number of posts were selected, totaling 8,496 instances.
- **Reviews[3]:** A dataset of 20,000 user reviews from the Mercado Livre and Amazon platforms, collected via web scraping. Only reviews related to the following six search terms used during collection were retained: *”camiseta básica”* (basic t-shirt), *”mochila”* (backpack), *”fones de ouvido sem fio”* (wireless headphones), *”relógio”* (watch), *”tênis”* (sneakers), and *”caixa de som portátil”* (portable speaker). The dataset represents spontaneous language with informal vocabulary and free structure.

The texts were subjected to a cleaning process that included character normalization, conversion to lowercase, and removal of elements such as URLs, emails, numbers, symbols, and punctuation. When applicable, the title and description fields were concatenated. For the reviews dataset, this was unnecessary since the texts were already in a unified format without titles. Additionally, in the review dataset, instances with four or fewer tokens (as counted by the `split()` operation were removed after preprocessing to eliminate overly generic reviews such as *”very good”*. After this filtering, 14,755 valid instances remained for analysis. A summary of the data after preprocessing is presented in Table 1.

[1] https://www.kaggle.com/datasets/oliveirasp6/poems-in-portuguese

[2] https://www.kaggle.com/datasets/bwandowando/reddit-rbrazil-subreddit-dataset

[3] https://www.kaggle.com/datasets/sampaiovitor/avaliaes-em-portugus-amazon-e-mercado-livre

**Table 1. Summary of the datasets used**

| Dataset | Quantity | No. of Clusters | Average Tokens |
|---|---|---|---|
| Poems | 647 | 6 | 194.05 |
| Reddit | 8496 | 5 | 49.37 |
| Reviews | 14755 | 6 | 14.75 |

*Note*: Average number of *tokens* calculated using `split()` after preprocessing.

## 3.2. Embeddings and Feature Extraction

To evaluate textual similarity, we selected the models BERTimbau and ST5-base, aiming to compare the performance of a monolingual model (BERTimbau) with a multilingual one (ST5), following the methodology proposed by [Wehrli et al. 2024]. BERTimbau was chosen to represent the monolingual setting, although it is not explicitly pre-trained for similarity tasks. Nevertheless, it is among the most used monolingual models for Portuguese. For embedding extraction, we applied mean pooling over the encoder outputs followed by Layer Normalization, as recommended by [Subakti et al. 2022], whose findings indicate that this combination yields effective results in clustering tasks.

### 3.2.1. BERTimbau

*BERTimbau* is a model based on the *BERT* architecture, trained exclusively on Brazilian Portuguese texts using the *brWaC* corpus, which contains approximately 2.68 billion tokens from the web. Being monolingual and fine-tuned for Portuguese, it tends to outperform multilingual models like *mBERT*, in semantic understanding tasks for the language [Wu and Dredze 2020]. In this work, we used both the *base*[4] and *large*[5] variants, extracting embeddings from the encoder's final layer. Aggregation was performed using *mean pooling*, followed by Layer Normalization.

### 3.2.2. ST5

The *ST5* (Sentence-T5) model is a modification of the *T5* (Text-to-Text Transfer Transformer) proposed by [Ni et al. 2021], trained specifically for tasks involving semantic similarity and sentence generation in vector spaces. Based on an *encoder-decoder* architecture, *ST5* underwent *fine-tuning* to produce embeddings capable of capturing the global semantics of sentences, making it particularly effective for clustering tasks. In this study, the *base* version [6] was used, as it offers a good balance between performance and computational cost. As with *BERTimbau*, the vectors were obtained from the encoder's output, using mean pooling and subsequent Layer Normalization.

## 3.3. Clustering Algorithms

Three distinct clustering algorithms with complementary approaches were used: *K-Means*, *Agglomerative Clustering* and the neural model *Deep Embedded Clustering*

[4] https://huggingface.co/neuralmind/bert-base-portuguese-cased

[5] https://huggingface.co/neuralmind/bert-large-portuguese-cased

[6] https://huggingface.co/sentence-transformers/sentence-t5-base

(DEC):

### 3.3.1. K-Means

*K-Means* is a centroid-based partitioning algorithm, widely used for its simplicity and computational efficiency [Tan et al. 2014]. It partitions data into $k$ groups by minimizing the sum of squared distances between points and their corresponding centroids. In this work, the number of clusters $k$ was set according to the number of unique labels present in the data. The implementation used the version available in the `scikit-learn` library.

### 3.3.2. Agglomerative Clustering

O *Agglomerative Clustering* is a hierarchical technique that follows a bottom-up approach: each instance starts as an individual cluster and the cluster pairs are iteratively merged based on a similarity criterion. In this study, the variant using the *Ward linkage*, criterion was adopted, which seeks to minimize intracluster variance at each merge, promoting the formation of more compact and approximately spherical groups [Murtagh and Legendre 2014]. The implementation was carried out using the `scikit-learn` library.

### 3.3.3. Deep Embedded Clustering (DEC)

DEC is a neural network–based algorithm proposed by [Xie et al. 2016], which combines representation learning with clustering in a joint manner. The architecture used in this research comprised a network with intermediate layers of dimensions $[500, 500, 2000, 5]$, with the last layer representing a compact latent representation of the data. The model was trained in two phases: a pretraining phase with an *autoencoder* using the Adam optimizer for 500 epochs, followed by a fine-tuning phase based on Kullback-Leibler divergence loss, with a batch size of 256 and up to 5000 iterations. The implementation followed the parameters described in [Subakti et al. 2022].

## 3.4. Metrics

To evaluate the quality of the clusters generated by the algorithms, three main metrics were used, commonly found in comparative clustering studies: *accuracy*[Subakti et al. 2022][Borges 2025], *V-Measure*[Delibasis 2019] and *Adjusted Rand Index (ARI)*[Guan et al. 2020]. These metrics compare the generated clusters to the actual labels present in the datasets, and are therefore classified as external metrics.

Clustering accuracy measures the proportion of correctly grouped instances, considering the best possible alignment between predicted and true labels. Since cluster labels have no fixed meaning, the formula applies an optimal permutation (using the Hungarian algorithm) to maximize the number of matches.

$$\text{Accuracy} = \frac{1}{n} \max_{\pi \in S_K} \sum_{i=1}^{n} 1\{y_i = \pi(\hat{y}_i)\} \tag{1}$$

- $n$: total number of samples.
- $y_i$: true label of the iii-th sample.
- $\hat{y}_i$: predicted label (assigned cluster) of the iii-th sample.
- $S_K$: set of all possible permutations of the $K$ predicted labels.
- $\pi$: specific permutation that maps predicted labels to true labels.
- $1\{\cdot\}$: indicator function.

V-Measure is based on mutual information and evaluates clustering quality as the harmonic mean between homogeneity ($h$) and completeness ($c$):

$$\text{V-Measure} = 2 \cdot \frac{h \cdot c}{h + c},$$

where:

$$h = 1 - \frac{H(C|K)}{H(C)}, \quad c = 1 - \frac{H(K|C)}{H(K)},$$

with $H(\cdot)$ denoting entropy, $C$ representing the true classes, and $K$ the generated clusters. The metric ranges from 0 (no correspondence) to 1 (perfect correspondence).

The Adjusted Rand Index (ARI) measures the degree of agreement between the clusterings, adjusted for chance. This metric considers pairs of samples that are either grouped together or separated, both in the true clustering and in the predicted one:

$$\text{ARI} = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}},$$

where:

- $n_{ij}$ is the number of elements from class iii in cluster $j$;
- $a_i = \sum_j n_{ij}$, the total number of elements in class $i$;
- $b_j = \sum_i n_{ij}$, the total number of elements in cluster $j$;
- $n$ is the total number of samples.

The ARI ranges from -1 (total disagreement) to 1 (perfect agreement), with 0 being the expected value for random clustering.

These three metrics complement each other by offering different perspectives on clustering quality: global correspondence (accuracy), informational structure (V-Measure) and pairwise agreement (ARI).

## 4. Results

The performance of the algorithms and embeddings is presented and discussed in this section. The results are summarized in Tables 2, 3, and 4, where values in **bold** indicate the best result obtained within each representation model, while values in **<u>bold and underlined</u>** highlight the overall highest value for each metric.

**Table 2. Clustering metrics for the Poems dataset**

| Model | Algorithm | Accuracy | V-measure | ARI |
|---|---|---|---|---|
| BERTimbau base | K-means | **0.5440** | **0.3379** | **0.2598** |
| | Agglomerative | 0.4745 | 0.3088 | 0.1855 |
| | DEC | 0.5008 | 0.3111 | 0.2148 |
| BERTimbau large | K-means | 0.5008 | 0.2932 | 0.2080 |
| | Agglomerative | **0.5131** | **0.3173** | **0.2365** |
| | DEC | 0.4544 | 0.2831 | 0.1742 |
| ST5 | K-means | 0.3014 | 0.0965 | 0.0420 |
| | Agglomerative | 0.3199 | **0.1365** | 0.0444 |
| | DEC | **0.3400** | 0.1288 | **0.0882** |

**Table 3. Clustering metrics for the *Reddit* dataset**

| Model | Algorithm | Accuracy | V-measure | ARI |
|---|---|---|---|---|
| BERTimbau base | K-means | 0.3742 | 0.2465 | 0.1215 |
| | Agglomerative | **0.4018** | **0.2469** | **0.1420** |
| | DEC | 0.3345 | 0.2309 | 0.1285 |
| BERTimbau large | K-means | 0.3695 | 0.2333 | 0.1252 |
| | Agglomerative | **0.4091** | **0.2572** | **0.1477** |
| | DEC | 0.3498 | 0.2392 | 0.1158 |
| ST5 | K-means | 0.5099 | 0.2066 | 0.0231 |
| | Agglomerative | **0.5972** | **0.2383** | **0.3221** |
| | DEC | 0.3080 | 0.1465 | 0.0656 |

**Table 4. Clustering metrics for the *Reviews* dataset**

| Model | Algorithm | Accuracy | V-measure | ARI |
|---|---|---|---|---|
| BERTimbau base | K-means | 0.2539 | 0.0121 | 0.0012 |
| | Agglomerative | 0.2954 | 0.0268 | 0.0089 |
| | DEC | **0.3316** | **0.1193** | **0.0864** |
| BERTimbau large | K-means | **0.2894** | 0.0471 | **0.0337** |
| | Agglomerative | 0.2670 | 0.0434 | 0.0002 |
| | DEC | 0.2874 | **0.0547** | 0.0205 |
| ST5 | K-means | 0.2735 | 0.0064 | 0.0097 |
| | Agglomerative | **0.2844** | 0.0247 | 0.0155 |
| | DEC | 0.2638 | **0.0397** | **0.0219** |

There is a noticeable variation in the performance of clustering algorithms across the different datasets. The classical algorithms — *K-means* and *Agglomerative* — performed substantially better on the smaller datasets, while *DEC*, combined with *BERTimbau base*, was the only configuration that yielded relevant results on the **Reviews** dataset.

In the small **Poems** dataset, with fewer than 1,000 instances, the best combination was *BERTimbau base* with *K-means*, which achieved **0,5440** accuracy and **0,3379** *V-measure*, outperforming all others. The **large** variant of BERTimbau also delivered solid performance with *Agglomerative*, reaching **0,5131** accuracy. On the other hand, the *ST5* model produced modest results, with its best outcome using *DEC*, yet still significantly below the monolingual models.

In the **Reddit** dataset, with around 8,500 instances, *ST5* with *Agglomerative* achieved **0,5972** accuracy and **0,3221** ARI, outperforming even the models trained exclusively on Portuguese. This result confirms *ST5*'s affinity with this type of data, as Reddit corpora were included in its pretraining. In this scenario, the *Agglomerative* algorithm proved to be the most effective overall.

For the **Reviews** dataset, with approximately 15,000 examples, only the *DEC* with *BERTimbau base* combination showed relevant performance, reaching **0,3316** accuracy and **0,1193** *V-measure*. All other combinations showed very low values, highlighting the difficulty traditional methods face with this kind of data. The superior performance of *DEC* can be attributed to its ability to capture semantic nuances, as well as the large volume of data, which favors deep learning-based approaches.

The results confirm trends observed in the literature but also highlight specifics related to Brazilian Portuguese and the datasets used. In smaller sets, like poems, classical algorithms — especially *K-means* and *Agglomerative* with *ward linkage* — delivered superior performance. This behavior partially aligns with the findings of [Borges 2025], although that study highlighted *Single Linkage*, while here the *ward* variant proved more effective.

The multilingual *ST5* model achieved its best performance on the Reddit dataset, likely due to its familiarity with that domain, as suggested by [Keraghel et al. 2024]. This reinforces the importance of alignment between a model's pretraining domain and the target data type.

For larger and semantically richer datasets, like Reviews, only the combination of *DEC* and *BERTimbau base* produced meaningful results. This supports the findings of [Subakti et al. 2022], who emphasize the superiority of deep learning approaches in contexts rich in data and semantic complexity.

To improve the understanding of how the clustering algorithms performed, we present below a simple analysis of the poems dataset, including its 3D projection with the real labels (Figure 2) and the per-class accuracy of the best combination of embeddings and clustering algorithm (Table 5). In the 3D projection, it can be observed that the instances corresponding to Olavo Bilac's poems are the most scattered in the vector representation, which is reflected in the accuracy results, since he shows 0% correspondence after the application of the Hungarian algorithm. This outcome indicates that the clusters formed by the model were not able to create a consistent grouping predominantly representing his works, causing all instances to be assigned to other labels during the optimal matching process. In contrast, Augusto dos Anjos, known for a singular style and markedly personal language, was the poet who achieved the best accuracy metric, reaching 75.95%.

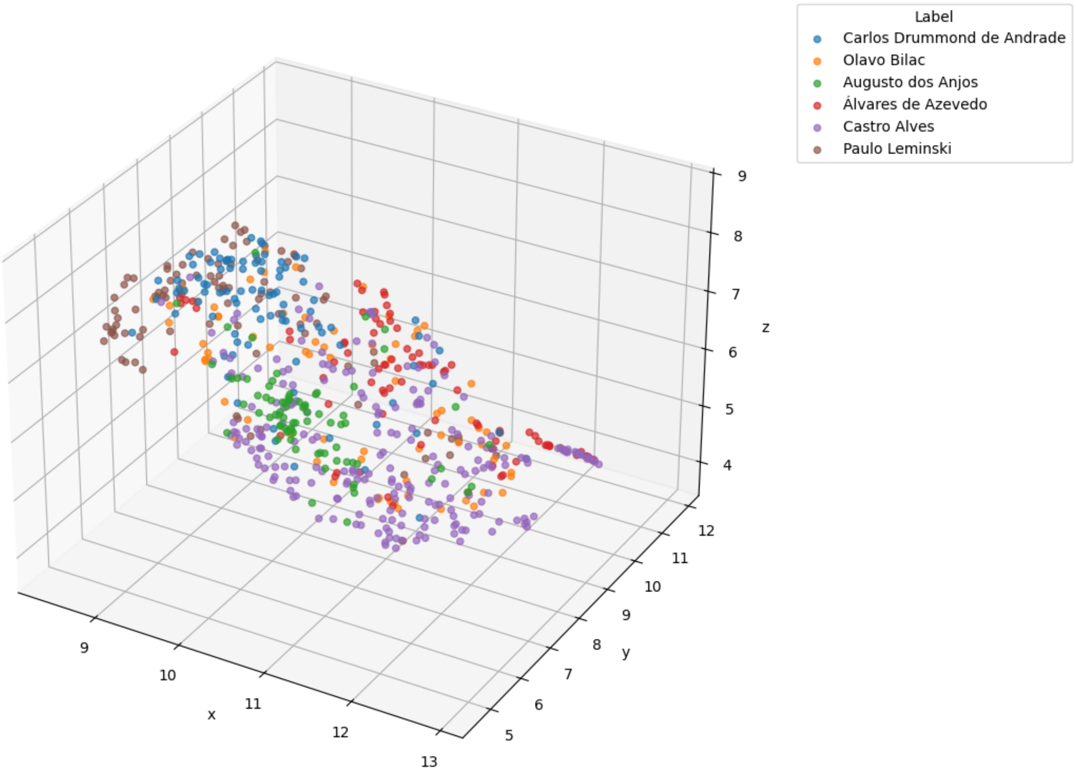

**Figure 2. *UMAP* reduced representation of the poems dataset with their real labels**

**Table 5. Accuracy per poet after applying the Hungarian algorithm**

| **Author** | **Accuracy (%)** |
|---|---|
| Augusto dos Anjos | 75.95 |
| Paulo Leminski | 69.07 |
| Carlos Drummond de Andrade | 63.74 |
| Álvares de Azevedo | 60.47 |
| Castro Alves | 51.80 |
| Olavo Bilac | 0.00 |

## 5. Conclusion

This study presented a systematic evaluation of the performance of different combinations of vector representation models and clustering algorithms applied to Portuguese-language texts. Three distinct datasets were used — poems, Reddit posts, and product reviews — in order to cover different domains, sizes, and linguistic styles.

The results show that the effectiveness of the methods varies significantly depending on the type of data analyzed. Classical algorithms such as *K-means* and *Agglomerative* proved effective on smaller datasets with more well-defined structures, such as the poem collection. In contrast, *DEC*, especially when combined with *BERTimbau base*, stood out on more complex and larger datasets, such as product reviews, highlighting the potential of deep clustering approaches for handling large-scale data with diffuse semantic structure.

Moreover, the experiments indicated that monolingual models like *BERTimbau* tend to provide more suitable representations for Portuguese texts, outperforming *ST5* in contexts where the language is more formal or archaic. On the other hand, *ST5* performed better on the Reddit dataset, possibly due to its training on data with similar style and domain.

It is therefore concluded that the choice of embedding and clustering algorithm should consider not only the overall quality of the methods but also the specific characteristics of the corpus, such as size, domain, and textual style. Future work may explore not only larger models for text representation but also traditional models like *TF-IDF* as well as combinations with even larger datasets and more sophisticated feature extraction techniques to further enhance the analysis.

## Acknowledgments

The authors would like to thank SiDi and the State University of Amazonas (UEA) for their support during the development of this work.

## References

Borges, B. R. (2025). Comparison of clustering techniques in text documents in portuguese (comparacao de tecnicas de clusterizacao em documentos de texto em portugues). *iSys: Revista Brasileira de Sistemas de Informação (Brazilian Journal of Information Systems)*, 18(1):4:1–4:17.

Delibasis, K. K. (2019). A new topology-preserving distance metric with applications to multi-dimensional data clustering. In MacIntyre, J., Maglogiannis, I., Iliadis, L., and Pimenidis, E., editors, *Artificial Intelligence Applications and Innovations*, pages 155–166, Cham. Springer International Publishing.

Grootendorst, M. (2022). Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Guan, R., Zhang, H., Liang, Y., Giunchiglia, F., Huang, L., and Feng, X. (2020). Deep feature-based text clustering and its explanation. *IEEE Transactions on Knowledge and Data Engineering*, PP:1–1.

Guo, X., Gao, L., Liu, X., and Yin, J. (2017). Improved deep embedded clustering with local structure preservation. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 1753–1759. AAAI Press.

Keraghel, I., Morbieu, S., and Nadif, M. (2024). Keraghel, i., morbieu, s., & nadif, m. (2024). beyond words: a comparative analysis of llm embeddings for effective clustering. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP).*

Murtagh, F. and Legendre, P. (2014). Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion? *Journal of Classification*, 31(3):274–295.

Ni, J., Ábrego, G. H., Constant, N., Ma, J., Hall, K. B., Cer, D., and Yang, Y. (2021). Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. Google Research, Mountain View, CA.

Souza, F., Nogueira, R., and Lotufo, R. (2020). BERTimbau: pretrained BERT models for Brazilian Portuguese. In *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23*.

Subakti, A., Murfi, H., and Hariadi, N. (2022). Subakti, a., murfi, h., & hariadi, n. (2022). the performance of bert as data representation of text clustering. *Journal of Big Data*, 9(15).

Tan, P.-N., Steinbach, M., and Kumar, V. (2014). *Introduction to Data Mining*. Pearson, New York.

Wehrli, S., Arnrich, B., and Irrgang, C. (2024). Wehrli, s., arnrich, b., & irrgang, c. (2024). german text embedding clustering benchmark. *arXiv preprint*, arXiv:2401.02709.

Wu, S. and Dredze, M. (2020). Are all languages created equal in multilingual BERT? In Gella, S., Welbl, J., Rei, M., Petroni, F., Lewis, P., Strubell, E., Seo, M., and Hajishirzi, H., editors, *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Xie, J., Girshick, R., and Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. *arXiv preprint arXiv:1511.06335*.