Submission date: 02/10/2024
Camera ready submission date: 06/03/2025

1st round notification: 03/03/2025
Available online: 27/03/2025
Publication date: 27/03/2025

Section: regular article

Comparação de Técnicas de Clusterização em Documentos de Texto em Português

Comparison of Clustering Techniques in Text Documents in Portuguese

Beatriz Ribeiro Borges[1]

[1] Faculdade de Computação/Universidade Federal de Uberlândia (UFU)
Uberlândia, Minas Gerais – Brasil

biarborges@ufu.br

Abstract. *Managing the vast amount of text data in the digital world is a complex challenge. An effective approach to tackle it is through the technique of text document clustering. This study evaluated the performance of three clustering algorithms — K-Means, Single Linkage, and Gaussian Mixture Model (GMM) — in clustering Brazilian Portuguese news articles using BERTimBau, a Portuguese variant of the BERT model, for preprocessing. Metrics such as accuracy, F1-score, Rand index, and Jaccard coefficient were used for evaluation. The results of these metrics indicated that Single Linkage achieved the best overall performance, surpassing K-Means and GMM in most of the evaluated criteria.*

Keywords. *Text clustering; BERT; BERTimBau; K-Means; Single Linkage; Gaussian Mixture Model.*

Resumo. *Lidar com a enorme quantidade de dados de texto no mundo digital é um desafio complexo. Uma abordagem eficaz para enfrentá-lo é através da técnica de agrupamento de documentos de texto. Este estudo avaliou o desempenho de três algoritmos de clustering — K-Means, Single Linkage e Gaussian Mixture Model (GMM) — na categorização de artigos de notícias em português brasileiro utilizando o modelo BERTimBau, variação em português do modelo BERT, para pré-processamento. As métricas utilizadas foram acurácia, F1-score, índice Rand e coeficiente de Jaccard. Os resultados dessas métricas indicaram que o Single Linkage obteve a melhor performance global, superando K-Means e GMM na maioria dos critérios avaliados.*

Palavras-Chave. *Clusterização de texto; BERT; BERTimBau; K-Means; Ligação simples; Modelo de Mistura Gaussiana.*

1. Introduction

In the digital world, we are surrounded by an overwhelming amount of text data. From news articles and social media posts to academic papers and corporate documents, the volume of information is growing rapidly. Managing and making sense of this vast amount

Cite as: Borges, B. R. (2025). Comparison of Clustering Techniques in Text Documents in Portuguese (Comparação de Técnicas de Clusterização em Documentos de Texto em Português). iSys: Revista Brasileira de Sistemas de Informação (Brazilian Journal of Information Systems), Vol18(1), 4:1-4:17. DOI: 10.5753/isys.2025.5029

of text data is a significant challenge. One of the effective ways to address this challenge is through text document clustering. Text document clustering is an unsupervised learning technique that aims to identify natural groupings among text documents. The goal is for the documents within the same cluster to be similar to each other, while documents in different clusters should exhibit significant differences [Manpreet Kaur 2021].

Achieving high accuracy and low error rates in text clustering is challenging. Traditional clustering algorithms like k-means, Hierarchical Agglomerative Clustering and Gaussian Mixture Model (GMM) have been widely used, but they often struggle with the complexity of textual data. Recent advancements in natural language processing (NLP) have opened new possibilities for improving clustering performance.

One of the most frequently used methods to represent textual data is Term Frequency Inverse Document Frequency (TF-IDF). However, TF-IDF cannot account for the position and context of a word in a sentence. The Bidirectional Encoder Representation from Transformers (BERT) model, on the other hand, can produce text representations that incorporate both the position and context of a word within a sentence [Subakti et al. 2022]. BERTimBau, a variation of BERT designed for Portuguese, offers a promising approach for preprocessing Portuguese text documents, capturing the language's nuances more effectively than older methods [Souza et al. 2020a].

In this paper, the performance of three clustering algorithms — k-means, hierarchical agglomerative clustering using single linkage and Gaussian Mixture Model — on Brazilian Portuguese news articles will be compared. BERTimBau will be utilized for preprocessing to enhance the clustering quality and address the limitations of traditional approaches. The study aims to provide insights into how well these algorithms handle Portuguese text data and contribute to the field of text clustering research.

To evaluate the quality of the cluster, four primary metrics will be employed: accuracy, F1-score, rand index, and Jaccard coefficient. Accuracy measures the overall correctness of the clustering assignments, indicating the proportion of correctly classified instances among all instances. The F1-score will be used to balance precision and recall, providing an overall measure of clustering performance. The Rand Index is a measure that assesses the agreement between predicted clusters and true clusters, considering all pairs of samples and measuring both true positives and true negatives. Additionally, the Jaccard coefficient will evaluate the similarity of elements assigned to the same clusters and categories, offering insight into the overlap between clusters and true labels. These metrics were chosen for their ability to provide a comprehensive and nuanced evaluation of clustering results, particularly because the dataset consists of texts with known ground truth categories.

As the generation and consumption of vast amounts of text data persist, the efficient clustering and analysis of documents become increasingly crucial for unlocking valuable information. Through the exploration of modern techniques and their application to Portuguese text data, this paper aims to advance the understanding of text clustering and its practical applications.

2. Related Work

Researchers have been working on different ways to tackle text document clustering. Their goal is to make clustering techniques better at organizing text data, dealing with all the complex details that come with it. There are four related papers to the studies in this work, which focus on the use of GMM, Hierarchical Agglomerative Clustering, K-Means and BERT.

One notable study addresses the challenge of clustering in biomedical text documents, which have expanded significantly in recent years. This expansion has heightened the need for effective techniques to extract useful information from large volumes of text. Traditional clustering methods often fall short due to their inability to capture the semantic relationships between biomedical texts, resulting in unsatisfactory clustering performance. To overcome these challenges, the study proposes using the Gaussian Mixture Model combined with pre-trained language representations from BioBERT, a domain-specific variation of BERT for biomedical text mining. The proposed GMM-based model demonstrated good results with a silhouette coefficient of 0.3765 and an adjusted rand index of 0.4478 [Khishigsuren Davagdorj 2022].

Focusing on the classification of spam emails, there is a study that adopt a topic-based approach and employ hierarchical agglomerative clustering to label spam emails into multiple categories. The study introduces novel datasets, SPEMC-15K-E and SPEMC-15K-S, comprising approximately 15,000 emails each in English and Spanish. Through experimentation with various text representation techniques, the study identifies TF-IDF as the most effective method, achieving high classification performance and processing efficiency [Francisco Jáñez-Martino 2023].

The application of k-means clustering and topic modeling for clustering Arabic text documents is also addressed. By normalizing the weights in the document-term matrix and combining generative models with clustering techniques, the study demonstrated significant improvements in clustering quality compared to previous approaches. External measures such as purity, F-measure, entropy, and accuracy validated the superiority of the combined method in clustering Arabic text documents. Given that this article focused on a dataset from a specific language other than English, it serves as a relevant model for the present paper [Mohammad Alhawarat 2018].

In another study, it is analyzed the performance of the Bidirectional Encoder Representation from Transformers (BERT) model as a data representation for text clustering. By comparing BERT with the Term Frequency Inverse Document Frequency (TF-IDF) method and evaluating different feature extraction and normalization techniques, the study concludes that BERT outperforms TF-IDF in most metrics across various clustering algorithms [Subakti et al. 2022].

3. Methodology

This chapter presents the methodology to conduct this comparative analysis, following the Knowledge Discovery in Databases (KDD) process. The KDD process involves numerous steps. It is necessary to select a dataset for the discovery process. Next, data

cleaning and preprocessing are carried out, followed by feature extraction to find useful features for representing the data. A method and its specific data mining algorithms (summary, classification, regression, clustering, etc.) are then chosen. This leads to data mining, where patterns are searched for or results are visualized. Finally, the results are evaluated and the knowledge is used, documenting and reporting the discovered insights [Fayyad et al. 1996].

3.1. Database

Public Portuguese language databases were sought to evaluate text clustering. The choice of public data was made due to its easy accessibility. The Fake.br database[1] was selected as it is one of the labeled datasets in Portuguese. The fact that it is labeled aids in the comparison and evaluation of the clustering results to be analyzed.

Fake.br was developed by the Interinstitutional Center for Computational Linguistics (Núcleo Interinstitucional de Linguística Computacional - NILC-USP) using traditional Natural Language Processing (NLP) approaches to detect fake news, as cited in [Souto Moreira et al. 2023]. The corpus used contains news articles previously classified as true or false. In total, the dataset has 7,200 news articles, with 3,600 being true and 3,600 false. For this study, the 3,600 labeled true news articles will be used. The corpus consists of plain text files (txt) divided into two main folders: ”true,”which contains the collected true news articles, and ”true-meta-information,”which contains the metadata information of each article (such as author, URL to the article, publication date, category, among others). The ony metadata used is the category. The news articles are categorized into six different themes: politics, TV and celebrities, society and daily life, science and technology, economy, and religion. The number of news articles per category is indicated in Table 1.

Tabela 1. Quantity of news articles per category in the Fake.Br corpus - Source: Compiled by the author (2024)

Category	Quantity
Politics	2090
TV and Celebrities	772
Life and Society	638
Science and Technology	56
Economy	22
Religion	22

The txt files were previously unified and transformed into a single csv (comma-separated values) file, simplifying the data manipulation process. This file contains the complete texts along with their corresponding categories. For clustering purposes, the categories will not be utilized; however, they will serve as ground truth for evaluation.

[1] https://github.com/roneysco/Fake.br-Corpus/tree/master

3.2. Preprocessing and Feature Extraction

Pre-processing is conducted to handle noise and potential malformed expressions commonly found in data from public sources. The BERTimbau model[2], an adaptation of the BERT model, will be utilized for both text pre-processing and feature definition. An additional pre-processing step involves removing special characters using Python language functions and libraries. Furthermore, BERT will be employed for text tokenization. When using BERT, neither stopwords removal nor stemming will be necessary. This is because transformer-based models are trained to handle complete texts and consider the context in which each token is embedded.

Regarding feature extraction, the decision was made to leverage BERT in its pre-training phase. This choice allows for exploring BERT's capabilities in capturing relevant features within their contexts in texts, contrasting with more conventional techniques.

3.2.1. Bert

BERT (Bidirectional Encoder Representations from Transformers), is a language representation model. It is designed for pre-training language models and is based on the Transformer architecture [Devlin et al. 2019].

The Transformer Architecture

The Transformer architecture follows an encoder-decoder structure. The encoder receives an input sequence composed of symbol representations $(x1, ..., xn)$ and produces a sequence of continuous representations $z = (z1, ..., zn)$. Using z as input, the decoder generates an output sequence $(y1, ..., ym)$ of symbols, one at a time. At each step, the model is autoregressive, consuming previously generated symbols as additional input when generating the next one. The Transformer adheres to this general structure by employing layers of self-attention and feedforward layers. The layers are fully connected point-to-point in both the encoder and decoder [Vaswani et al. 2017]. Figure 1 illustrates the Transformer architecture.

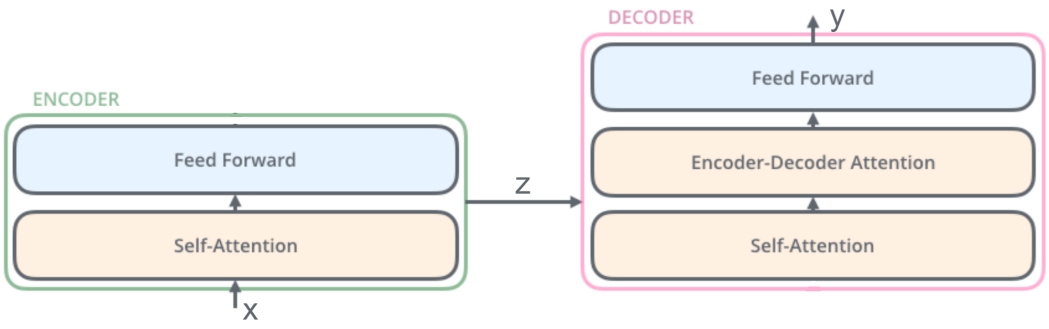

Figura 1. Transformer Architecture Model - Source: Adapted from [Lima 2021]

[2]https://github.com/neuralmind-ai/portuguese-bert

- Self-attention allows each word (or token) in a sentence to be related to all other words, which helps to efficiently capture contextual dependencies between words.
- The Feed-Forward layer is composed of:
  - Feed-Forward Input: The vector resulting from the self-attention mechanism for a token.
  - First Dense Layer: Transforms this vector into a higher-dimensional space to capture more details.
  - ReLU (Activation): Introduces non-linearity, allowing the network to capture complex relationships.
  - Second Dense Layer: Transforms the vector back to its original dimension, but now with a richer and more complex representation.
  - Feed-Forward Output: A transformed vector that still represents the token in the context of the sentence, but with an enhanced representation.

  After the self-attention mechanism processes the input vectors, capturing contextual relationships between words, the feed-forward network refines these representations. By passing through two dense layers with an activation function in between, the word representations are enriched, enabling the model to learn more complex patterns in the data.

The Bert Architecture

BERT is a bidirectional Transformer encoder composed of multiple layers, based on the original implementation of the Transformer. The use of Transformers has become widely adopted, and BERT's implementation is nearly identical to the original version, with the exception that only the encoder is used [Devlin et al. 2019]. The key difference of BERT compared to other models is its bidirectionality, which allows for a greater capacity for interpretation. Figure 2 illustrates the bidirectionality of BERT compared to OpenAI's GPT.

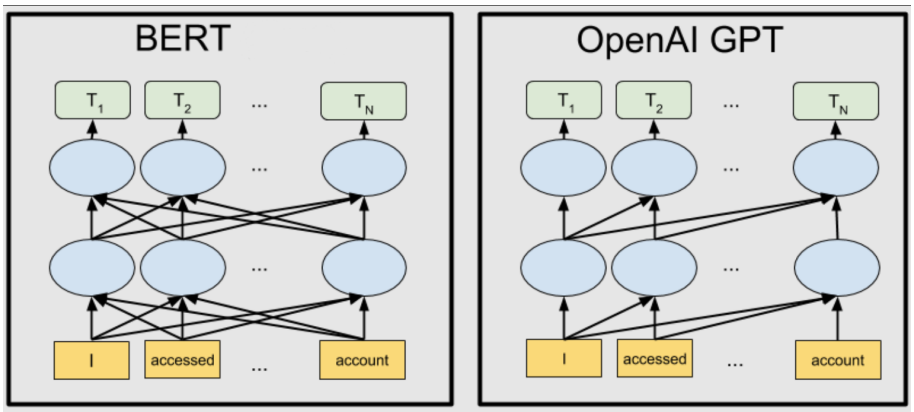

Figura 2. Comparison of the neural network architecture between BERT and OpenAI GPT - Source: Adapted from [AI 2018]

There are two distinct steps in the BERT process: pre-training and fine-tuning. During pre-training, the model is trained on unlabeled data, tackling various specific tasks.

In fine-tuning, BERT is initialized with the pre-trained parameters and refined using datasets for specific natural language processing tasks [Devlin et al. 2019]. For this work, only the pre-training will be used.

In pre-training, traditional left-to-right or right-to-left language models are not used. Instead, two unsupervised tasks are employed: MLM (Masked Language Model) and NSP (Next Sentence Prediction). In MLM, some words in the input are randomly masked, and the model tries to predict these words based on the context of the other words in the sentence. NSP trains the model to understand the relationship between two sentences. The objective of the NSP task is to train the BERT model to understand if a sentence B logically follows a sentence A [Devlin et al. 2019].

BERTimbau

BERT still presents various variations across different languages. In addition to the original BERT trained in a few languages, the Multilingual BERT (mBERT), trained in 104 languages, was developed by Google as the multilingual version of BERT [Wu and Dredze 2020]. Another multilingual variation is XLM-RoBERTa, a version of the RoBERTa model (also based on BERT), developed by Facebook AI Research (FAIR) [Velankar et al. 2023]. However, multilingual BERT-based models are generally considered inferior to monolingual models [Wu and Dredze 2020]. There are several versions available in various languages of monolingual models based on BERT. For Brazilian Portuguese, for example, there is BERTimbau, which uses the brWaC corpus (Brazilian Web as Corpus) for pre-training data, a crawl of Brazilian web pages containing 2.68 billion tokens from 3.53 million documents [Souza et al. 2020b]. It is BERTimbau that will be used in this paper.

3.3. Cluster Algorithms

Three distinct clustering algorithms will be employed in this study: k-means, Single Linkage Hierarchical Agglomerative Clustering, and Gaussian Mixture Model. These algorithms were selected for their diverse approaches to clustering, ranging from centroid-based partitioning to hierarchical merging and nature-inspired optimization, allowing for a comprehensive evaluation of clustering performance on Portuguese text data.

3.3.1. K-Means Algorithm

K-means is a centroid-based method that partitions data into K clusters by iteratively refining cluster centroids based on the mean of data points assigned to each cluster.

First, K initial centroids are chosen, where K is a user-specified parameter representing the desired number of clusters. Each point is then assigned to the closest centroid, and each collection of points assigned to a centroid forms a cluster. The centroid of each cluster is then updated based on the points assigned to the cluster. The assignment and update steps are repeated until no point changes clusters, or equivalently, until the centroids remain the same [Tan et al. 2014].

Pseudocode of K-Means

K-Means Clustering

Input: Set of data points
Output: K clusters and their centroids
procedure KMEANS(K, points)
    Select K points as initial centroids
    repeat
        Form K clusters by assigning each point to its closest centroid
        Recompute the centroid of each cluster
    until the centroids do not change
end procedure

3.3.2. Hierarchical Agglomerative Clustering Algorithm

The Hierarchical Agglomerative Clustering is an agglomerative clustering technique that merges clusters based on the similarity between their most similar members. This method forms clusters in a hierarchical tree structure, offering insights into the nested relationships within data. It starts with individual points as clusters, successively merge the two closest clusters until only one cluster remains [Tan et al. 2014].

Pseudocode of Hierarchical Agglomerative Clustering

Hierarchical Clustering

Input: Set of data points
Output: Dendrogram representing hierarchical clustering
procedure HIERARCHICALCLUSTERING(points)
    Compute the proximity matrix, if necessary
    repeat
        Merge the closest two clusters
        Update the proximity matrix to reflect the proximity between the new cluster and the original clusters
    until only one cluster remains
end procedure

The key operation in hierarchical agglomerative clustering is the computation of the proximity between two clusters. The definition of cluster proximity differentiates various hierarchical agglomerative techniques such as MIN (also known as single linkage), MAX (also known as complete linkage), and Group Average. In this study, the single linkage technique is used. Single linkage defines cluster proximity as the proximity between the closest two points that are in different clusters as shown in Figure 3.

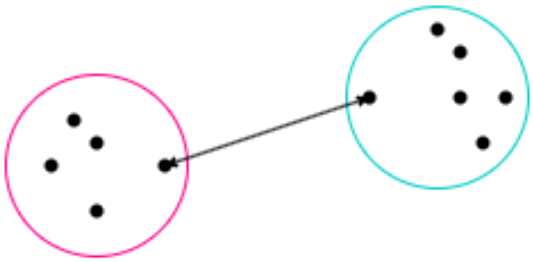

Figura 3. Single linkage between two clusters - Created by the author (2024)

3.3.3. Gaussian Mixture Model

The GMM is a probabilistic model represented as a weighted sum of multiple Gaussian distributions. It is employed to iteratively estimate a set of parameters until the model converges to a specified threshold. GMM uses a predefined number of Gaussian distributions to identify the number of clusters within a dataset [Khishigsuren Davagdorj 2022].

A normal distribution, or Gaussian distribution, is defined by two parameters: the mean (μ) and the variance (σ^2) of the data space. The probability density function of a normal distribution describes the probability of a data point x belonging to Gaussian component k based on its mean and variance. Figure 4 shows a distribution for one dimension (variable) and Figure 5 shows the visualization with 2 dimensions in a plane.

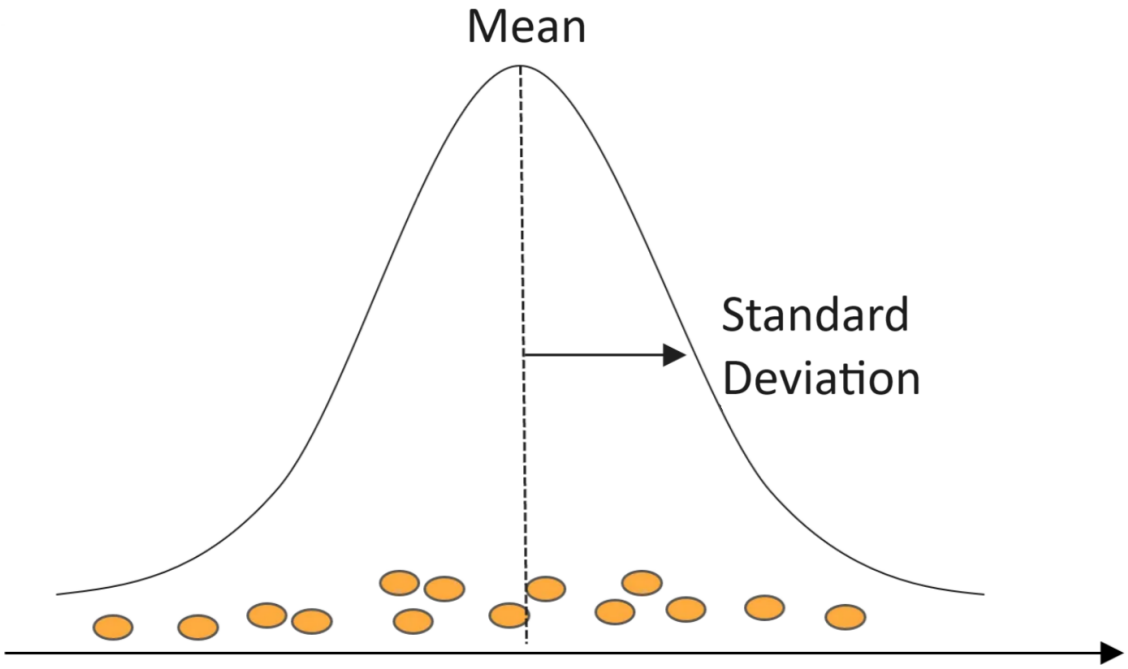

Figura 4. Gaussian visualization for one dimension - Source: Adapted from [Ferreira 2023]

The density function is given by Equation 1:

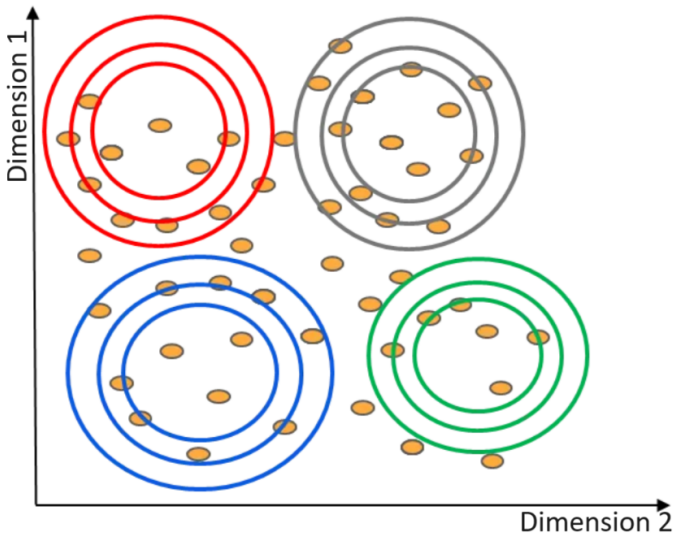

Figura 5. Gaussian visualization for two dimensions - Source: Adapted from [Ferreira 2023]

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{1}$$

The probability density function of a Gaussian mixture, for all components (clusters), is given by Equation 2:

$$p(x) = \sum_{k=1}^{K} \pi_k \cdot f(x \mid \mu_k, \sigma_k^2) \tag{2}$$

- x: It is a vector representing an object in the dataset.
- $p(x)$: It is the probability density in the region of the variable space.
- K: Number of components (Gaussian distributions).
- π_k: Weight of the k-th component, which represents the probability of any given point belonging to that component. The sum of all weights is 1: $\sum_{k=1}^{K} \pi_k = 1$.
- μ_k: Mean of the k-th component.
- σ_k^2: Variance of the k-th component.

To estimate the parameters of the GMM from the data, the Expectation-Maximization Algorithm (EM) is used. Expectation (Equation 3) calculates the probability of each point belonging to one of the K Gaussian curves, while Maximization updates the model parameters (π, σ, μ) using the responsibilities computed in the Expectation step.

$$\gamma(z_{ik}) = \frac{\pi_k \cdot f(x_i \mid \mu_k, \sigma_k^2)}{\sum_{j=1}^{K} \pi_j \cdot f(x_i \mid \mu_j, \sigma_j^2)} \tag{3}$$

- $\gamma(z_{ik})$: It is the probability that the data point x_i belongs to the k component of the GMM.

- The variable z is a discrete variable. z_{ik} indicates whether the observation x_i belongs (1) or not (0) to the component k.

Pseudocode of Gaussian Mixture Mode

Gaussian Mixture Model (GMM) using Expectation-Maximization (EM)

Choose a number of Gaussians K.
Initialize parameters randomly in space.
repeat
    Expectation Step:
    for each data point x_i **do**
        for each Gaussian component k **do**
            Calculate responsibility $\gamma(z_{ik})$
        end for
    end for
    Maximization Step:
    for each Gaussian component k **do**
        Update parameters π_k, μ_k, σ_k^2
    end for
until convergence criteria are met

3.4. Evaluation

Clustering validation methods can be divided into two primary types: external validation and internal validation. The key distinction lies in whether external information is utilized for the validation process. External validation methods incorporate information not contained within the dataset to assess how well the clustering results align with an external reference structure, such as predefined class labels [Aggarwal and Reddy 2013].

To assess the quality of the clusters produced by the algorithms, the methodology will utilize this main metrics: accuracy, F1-score, purity, and Jaccard coefficient. Since the dataset includes texts with established ground truth categories, these metrics will offer a detailed and thorough evaluation of clustering effectiveness.

3.4.1. Accuracy

Accuracy (Equation 4) measures the proportion of correctly classified instances among all instances in a clustering task. It serves as a fundamental metric for evaluating the performance of clustering algorithms by assessing how well the assigned clusters match the true categories or labels of the data [Tan et al. 2014]. A higher accuracy indicates that a larger percentage of data points are correctly grouped into clusters that align with their actual categories or classes. Accuracy is particularly valuable in assessing the overall correctness and effectiveness of clustering models in organizing data into meaningful groups.

$$Accuracy = \frac{Number\ of\ correct\ instances}{Total\ number\ of\ instances} = \frac{TP + TN}{TP + FN + FP + TN} \tag{4}$$

where TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative.

3.4.2. F1-Measure

To discuss the F1 measure, it's essential first to understand precision and recall. Precision (Equation 5) and recall (Equation 6) are two metrics that focus on positive data. While error rates are useful for estimating overall prediction performance, when there is a large number of negative data points, it is possible that the algorithm can achieve very high precision, meaning a very low error rate, simply by predicting all data as negative. Therefore, it is useful to measure a performance by disregarding correctly predicted negative data and instead examining the types of errors made by the classifier [Weiss et al. 2015].

$$Precision = \frac{Number\ of\ correct\ positive\ predictions}{Total\ number\ of\ positive\ predictions} = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{Number\ of\ correct\ positive\ predictions}{Number\ of\ documents\ of\ positive\ class} = \frac{TP}{TP + FN} \tag{6}$$

- Precision is a metric that measures the proportion of correctly predicted positive examples out of all examples predicted as positive by the classifier.
- Recall is a metric that measures the proportion of correctly predicted positive examples out of all actual positive examples.

The F1 score ranges from 0 to 1, with 1 being the best possible result, indicating that the model achieves both high precision and high recall simultaneously. Since the F1-measure is the harmonic mean of precision and recall, the equation is given by:

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{7}$$

where P is precision and R is recall.

3.4.3. Rand Index

The Rand index serves as an objective benchmark that enables the evaluation of performance using classification datasets. It assesses not only classifiers, which generate varied data partitions with the correct number of classes, but also clustering outcomes, where different data partitions may consist of varying numbers of clusters [Campello 2007].

The Rand Index accomplishes the evaluation by comparing all possible pairs of samples and tallying those that are classified identically in both predicted and true clusters, as well as those that are classified differently, whether correctly or incorrectly. It ranges

from 0 to 1, where 1 indicates a perfect agreement between the two clusterings, and 0 indicates completely different clusterings. The formula is:

$$RandIndex = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}} \tag{8}$$

Where:

- f_{11} represents the number of elements common to both sets,
- f_{01} represents the number of elements that are in set B but not in set A,
- f_{10} represents the number of elements that are in set A but not in set B,
- f_{00} represents the number of elements different in both sets.

3.4.4. Jaccard Coefficient

The Jaccard coefficient is a metric used to measure the similarity and diversity between sample sets. It is calculated as the division between the size of the intersection and the size of the union of the data sets [Ayub et al. 2020].

The formula for the Jaccard coefficient is:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{9}$$

The Equation (9) can be read as follows:

$$J = \frac{f_{11}}{f_{01} + f_{10} + f_{11}} \tag{10}$$

The Jaccard index focuses only on the presence of elements in the sets, not their absence. Therefore, elements that belong to neither set (i.e., f_{00}) are not relevant for calculating the similarity between the sets. As a result, the Jaccard coefficient is frequently used to handle objects consisting of asymmetric binary attributes [Tan et al. 2014].

4. Implementation

The codes were divided into 4 Python files. The first one was dedicated to preprocessing (text_processing.py). The other 3 are for executing algorithms and their evaluations: K-means (kmeans.py), Single Linkage (singleLinkage.py), and Gaussian Mixture Model (gmm.py). All are available on GitHub[3]. The clustering algorithms have their results saved in a CSV file.

Libraries:

- It was used scikit-learn for evaluation metrics (`accuracy_score`, `f1_score`, `rand_score`, `jaccard_score`).

[3]https://github.com/biarborges/kmeansSingleGMM/tree/main

- Each file uses pandas for data loading and manipulation, and numpy for matrix manipulation.
- scikit-learn was also used for clustering models.
- The algorithms were run 10 times each.

Parameters:

- K-Means: `n_clusters=6`.
- Single Linkage: `n_clusters=6`, `linkage='single'`.
- GMM: `n_components=6`, `init_params='random_from_data'`.

text_processing.py:

- text_processing.py performs text preprocessing, vectorization using a pretrained BERT model, and saves resulting embeddings in an embeddings.npy file.
- It utilizes regex (re) for text cleaning, transformers for tokenization and BERT model usage, torch for inference using GPU (if available), and tqdm for progress bar display.
- The embeddings are saved and subsequently loaded in each clustering algorithm from embeddings.npy.

5. Result and Discussions

The performance of three clustering algorithms—K-Means, Single Linkage, and Gaussian Mixture Model (GMM)—was evaluated using four different metrics: Accuracy, F1-score, Rand Index, and Jaccard Score. The results are summarized as follows:

Tabela 2. Results of text document clustering

	Accuracy	**F1-Score**	**Rand Index**	**Jaccard Score**
K-Means	0.1232	0.1904	0.5288	0.1132
Single Linkage	0.5800	0.4268	0.4150	0.3370
GMM	0.3524	0.4045	0.5009	0.2343

K-Means shows the lowest performance among the three, with an accuracy of 0.1232 and a Jaccard score of 0.1132, indicating poor clustering quality. The F1-score of 0.1904 and Rand Index of 0.5288 suggest that while K-Means might correctly classify some clusters, it struggles significantly with the overall classification task. The relatively higher Rand Index indicates some level of agreement between the clustering results and the ground truth, though the other metrics suggest that this agreement is not substantial.

This hierarchical clustering method outperforms K-Means and GMM in terms of accuracy (0.5800) and Jaccard score (0.3370). The F1-score of 0.4268, while not exceptional, is higher compared to K-Means and comparable to GMM. However, the Rand Index is 0.4150, which is lower than both K-Means and GMM. This suggests that Single Linkage may be better at forming clusters that align with the ground truth, but the clusters themselves might not be as distinct.

GMM demonstrates moderate performance with an accuracy of 0.3524 and a Jaccard score of 0.2343. The F1-score of 0.4045 is comparable to Single Linkage, indicating

that GMM is also relatively effective in identifying the correct clusters. The Rand Index of 0.5009, slightly lower than K-Means, suggests a moderate level of agreement with the ground truth.

Single Linkage provides the best performance in terms of accuracy and Jaccard score, making it the most suitable method among the three for this particular dataset. K-Means, while commonly used for its simplicity and speed, performs the worst in this context. GMM offers a balanced performance but does not surpass Single Linkage in most metrics. Therefore, Single Linkage demonstrates the most balanced performance across all metrics.

6. Conclusion

In this study, the performance of three text clustering algorithms — K-Means, Single Linkage, and Gaussian Mixture Model (GMM) — was evaluated using metrics such as accuracy, F1-score, Rand index, and Jaccard coefficient. The results demonstrate significant variations in the effectiveness of each algorithm when handling Brazilian Portuguese news articles preprocessed with BERTimBau.

While K-Means is often favored for its simplicity and speed, its performance in this context was unsatisfactory. GMM exhibited balanced performance and is widely used in text document contexts, yet it consistently performed inferiorly to Single Linkage across most criteria. Thus, hierarchical clustering techniques, specifically Single Linkage, show promise for text analysis in Portuguese.

Classic algorithms are widely adopted for their efficiency but face challenges in handling textual documents. These limitations can be mitigated by employing new text processing techniques that excel in natural language processing, as demonstrated by the use of BERTimBau in this study. BERTimBau's ability to capture linguistic and contextual nuances of texts significantly enhances clustering quality.

Despite promising results, it is crucial to acknowledge the limitations of this study. The evaluation was conducted exclusively on a specific dataset of Brazilian Portuguese news articles, which may limit the generalizability of the findings. Furthermore, the selection of algorithm parameters can significantly influence the outcomes. Future research should explore a broader range of language models, clustering techniques, and diverse text types to further advance the field of document clustering.

7. Acknowledgement

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Referências

[Aggarwal and Reddy 2013] Aggarwal, C. C. and Reddy, C. K. (2013). *Data Clustering: Algorithms and Applications*. Data Mining and Knowledge Discovery Series. Chapman & Hall/CRC, Boca Raton, FL.

[AI 2018] AI, G. (2018). Open sourcing bert: State-of-the-art pre-training for natural language processing. https://research.google/blog/open-sourcing-bert-state-of-the-art-pre-training-for-natural-language-processing/. Acesso em: 27 mar. 2025.

[Ayub et al. 2020] Ayub, M., Ghazanfar, M., Khan, T., et al. (2020). An effective model for jaccard coefficient to increase the performance of collaborative filtering. *Arab Journal of Science and Engineering*, 45:9997–10017.

[Campello 2007] Campello, R. (2007). A fuzzy extension of the rand index and other related indexes for clustering and classification assessment. *Pattern Recognition Letters*, 28(7):833–841.

[Devlin et al. 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[Fayyad et al. 1996] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37.

[Ferreira 2023] Ferreira, W. (2023). Desvendando padrões de dados: Uma jornada pelos métodos de clusterização (k-means, gaussian mixture, ...). https://medium.com/@willferreira08/desvendando-padrões-de-dados-uma-jornada-pelos-métodos-de-clusterização-k-means-gaussian-mixture-0046d3daae7c. Acesso em: 27 mar. 2025.

[Francisco Jáñez-Martino 2023] Francisco Jáñez-Martino, Rocío Aláiz-Rodríguez, V. G.-C. E. F. E. A. (2023). Classifying spam emails using agglomerative hierarchical clustering and a topic-based approach. *FREEDOM*, 139:110226.

[Khishigsuren Davagdorj 2022] Khishigsuren Davagdorj, Ling Wang, M. L. V.-H. P. K. H. R. N. T.-U. (2022). Discovering thematically coherent biomedical documents using contextualized bidirectional encoder representations from transformers-based clustering. 19(10):5893.

[Lima 2021] Lima, A. (2021). Introdução aos transformers. https://acervolima.com/introducao-aos-transformers/. Acesso em: 27 mar. 2025.

[Manpreet Kaur 2021] Manpreet Kaur, . V. (2021). An improved k-means based text document clustering using artificial bee colony with support vector machine. 8(7).

[Mohammad Alhawarat 2018] Mohammad Alhawarat, M. O. H. (2018). Revisiting k-means and topic modeling, a comparison study to cluster arabic documents. 6:42740.

[Souto Moreira et al. 2023] Souto Moreira, L., Machado Lunardi, G., de Oliveira Ribeiro, M., Silva, W., and Paulo Basso, F. (2023). A study of algorithm-based detection of fake news in brazilian election: Is bert the best. *IEEE Latin America Transactions*, 21(8):897–903.

[Souza et al. 2020a] Souza, F., Nogueira, R., and Lotufo, R. (2020a). Bertimbau: Pretrained bert models for brazilian portuguese. In Cerri, R. and Prati, R. C., editors, *Intelligent Systems*, pages 403–417, Cham. Springer International Publishing.

[Souza et al. 2020b] Souza, F., Nogueira, R., and Lotufo, R. (2020b). Bertimbau: Pretrained bert models for brazilian portuguese. In Cerri, R. and Prati, R. C., editors, *Intelligent Systems*, pages 403–417, Cham. Springer International Publishing.

[Subakti et al. 2022] Subakti, A., Murfi, H., and Hariadi, N. (2022). The performance of bert as data representation of text clustering. *Journal of Big Data*, 9(1):15.

[Tan et al. 2014] Tan, P.-N., Steinbach, M., and Kumar, V. (2014). *Introduction To Data Mining*. Pearson, New York.

[Vaswani et al. 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, Cambridge, MA. MIT Press.

[Velankar et al. 2023] Velankar, A., Patil, H., and Joshi, R. (2023). Mono vs multilingual bert for hate speech detection and text classification: A case study in marathi. In El Gayar, N., Trentin, E., Ravanelli, M., and Abbas, H., editors, *Artificial Neural Networks in Pattern Recognition*, pages 121–128, Cham. Springer International Publishing.

[Weiss et al. 2015] Weiss, S. M., Indurkhya, N., and Zhang, T. (2015). *Fundamentals of Predictive Text Mining*. Springer International Publishing, London, second edition edition.

[Wu and Dredze 2020] 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.