An Investigation Into Different Text Representations to Train an Artificial Immune Network for Clustering Texts

Matheus A. Ferraria[1], Vinicius A. Ferraria[1], Leandro N. de Castro[2,3] *

[1] Graduate Program in Electrical Engineering and Computing, Mackenzie Presbyterian University (Brazil)
[2] U.A. Whitaker College of Engineering, Florida Gulf Coast University (USA)
[3] Graduate Program in Technology, University of Campinas (Brazil)

Received 12 March 2023 | Accepted 18 August 2023 | Published 28 August 2023

Abstract

Extracting knowledge from text data is a complex task that is usually performed by first structuring the texts and then applying machine learning algorithms, or by using specific deep architectures capable of dealing directly with the raw text data. The traditional approach to structure texts is called Bag of Words (BoW) and consists of transforming each word in a document into a dimension (variable) in the structured data. Another approach uses grammatical classes to categorize the words and, thus, limit the dimension of the structured data to the number of grammatical categories. Another form of structuring text data for analysis is by using a distributed representation of words, sentences, or documents with methods like Word2Vec, Doc2Vec, and SBERT. This paper investigates four classes of text structuring methods to prepare documents for being clustered by an artificial immune system called aiNet. The goal is to assess the influence of each structuring method in the quality of the clustering obtained by the system and how methods that belong to the same type of representation differ from each other, for example both LIWC and MRC are considered grammar-based models but each one of them uses completely different dictionaries to generate its representation. By using internal clustering measures, our results showed that vector space models, on average, presented the best results for the datasets chosen, followed closely by the state of the art SBERT model, and MRC had the overall worst performance. We could also observe a consistency in the number of clusters generated by each representation and for each dataset, having SBERT as the model that presented a number of clusters closer to the original number of classes in the data.

Keywords

Artificial Immune Network, Clonal Selection, Natural Computing, Text Clustering, Text Structuring.

DOI: 10.9781/ijimai.2023.08.006

I. Introduction

Text mining corresponds to a set of techniques used to extract patterns or identify trends in documents (textual datasets), bringing together Information Retrieval, Natural Language Processing (NLP), and Data Mining techniques [1]–[5]. Thus, text mining involves knowledge in linguistics, informatics, statistics and cognitive sciences, among other areas. Whilst data mining seeks patterns in numerical and categorical data, text mining is about looking for patterns in texts. This superficial similarity between the two areas hides their main difference [6]: data mining deals with *structured data* in standard databases, whilst texts are *semi-* or *unstructured data* covered with uncertainties, context and ambiguity, which make their analysis and interpretation even more difficult. Thus, text mining deals with semi- or unstructured data that is usually pre-processed before a learning algorithm can be applied.

* Corresponding author.

E-mail addresses: 72208635@mackenzista.com.br (M. A. Ferraria), 72208643@mackenzista.com.br (V. A. Ferraria), leandron@unicamp.br, ldecastrosilva@fgcu.edu (Leandro N. de Castro).

The pre-processing step performs all the cleaning and structuring in the text to generate its structured representation suitable for the application of standard machine learning algorithms [7]. This text structuring step is usually the most sensitive and expensive one in computational terms, as it requires the processing of unstructured data [4], [8]. It can be divided into four main steps: 1) tokenization; 2) stop words removal; 3) lemmatization; and 4) representation of documents. After Steps (1) to (3), it is necessary to find a suitable representation for the documents (Step (4)), and there are different methods to do so.

The most common text representation approach is the so called *Bag of Words (BoW)* [9], [10], which models the documents only based on a specific weight calculated for each token (word) in the document, disregarding grammar, word order and context. These are called *vector space models*, in the sense that they transform texts into a set of vectors in a usually high-dimensional space, where each dimension corresponds to a word in the documents.

Alternative methods to represent documents include those that, instead of having each word as a dimension, use a pre-defined set of word categories to represent the documents and classify the words available into one of these categories. Examples include the *Linguistic Inquiry and Word Count* (LIWC), which references a dictionary of

grammatical, psychological and content word categories, the *Part-of-Speech Tagging* (sTagging), which accounts for the definition and context of words [11]–[13], and the MRC, which uses a predefined dictionary to map words into their respective psycholinguistic information [14].

A third class of text structuring method investigated in this paper is based on the concept of *word embedding* [15]–[17], that is, each word is represented as a real-valued vector that encodes its context and meaning, such that words with similar meanings appear closer to one another in the vector space. This type of distributed representation of words is generated by specific neural network architectures. Examples of these approaches include the Doc2Vec and Word2Vec algorithms, which model each paragraph and word, respectively, as a numerical vector representing its meaning and main characteristics.

The fourth and last representation is based on sentence embeddings, an extension of word embeddings generated by deep neural architectures. Sentence embedding models [18]–[21], such as SBERT, generate a vector taking into consideration both the semantics and linguistic aspects of a sentence or phrase by making use of the position, context and how every word is being used in the sentence [22]. While based on word embeddings, these models differ from them since word embeddings only hold isolated information for each word, while sentence embeddings are capable of extracting relationship between words and capture contextual information of a group of words like sentences, phrases and paragraphs [23].

After the documents are structured, any type of machine learning algorithm can be used to extract knowledge from the data. Tasks like clustering, classification and association rule mining can be performed. This paper investigates the influence of different text representation schemes, more specifically BoW, LIWC, sTagging, MRC, Doc2Vec, Word2Vec, and SBERT to prepare texts for being clustered by an *Artificial Immune Network* algorithm named aiNet [24]–[27]. To assess the performance of the algorithm, four datasets from the literature and two internal clustering measures were chosen.

The paper is organized as follows. Section II provides some background knowledge on the Immune Network Theory, the aiNet Algorithm, and the different text representation schemes used in the paper. Section III describes the implementation performed, results obtained and a discussion. The paper is concluded in Section IV with some general discussions and future research.

II. Background Knowledge

This research investigates the use of different text representation schemes combined with the aiNet algorithm to detect and extract clusters from text data. This section briefly reviews the biological phenomenon from which aiNet was inspired, the aiNet learning algorithm, and the different text representation schemes that will be used in the research.

A. Immune Network Theory

Among the most diverse components present in the immune system, *antibodies* (Ab) play a key role in its learning and evolution [28]. They work as a line of defense, recognizing and binding with *antigens* (Ag), thus generating Ag-Ab complexes that are then identified and destroyed by other immune cells [29], [30]. These cellular interactions are responsible for regulating and allowing the evolution of the Immune System (IS) [29] and the immune networks are responsible for key activities of immune cells, such as the emergence of memory cells and the *self - non-self discrimination* [30]–[32].

The *immune network theory* is a proposal that aims at explaining how the adaptive immune system works. It is based on the notion that antibodies contain receptors capable of recognizing one another and the foreign disease-causing agents, called antigens. This self-recognition capability implies that immune cells and molecules are naturally connected to one another forming an internal network in a dynamical equilibrium state. The invasion of antigens would then disturb the network, promoting a change in its internal state. As the network already contains the receptors for such antigens, these would be called *internal images* of the antigens [30].

Ag-Ab interactions are extremely important for the learning and evolutionary processes of the IS [32], since the affinity of these interactions help guiding the creation of *memory cells*, that is, a set of specialized cells that are rescued by the IS to promote a faster and more effective response to future invasions of the previously seen antigens [28].

Immune system adaptation to foreign antigens is based on a learning and evolutionary process that allows the maturation of antibody receptors so that they become increasingly better at recognizing antigens and, also, the increase in the sets of memory cells to known antigens. This means that after the immune system eliminates a certain disease-causing agent, its immune cells and molecules are more adapted (i.e., with greater affinity) to that specific antigen, and the concentration of these cells also increased significantly, ensuring an effective response to future invasions of the same or similar antigen [28], [30], [31].

In summary, an adaptive immune response involves the recognition of antigenic patterns, followed by the expansion (cellular reproduction) of high-affinity cells, antibody maturation (i.e., mutations that lead to better Ag-Ab affinity match), and clonal expansion (i.e., the increase in number of high-affinity cells) [28], [32]. Altogether, these processes are called *clonal selection and affinity maturation*. It is worth mentioning that antibody mutation during clonal expansion is inversely proportional to the Ag-Ab affinity, that is, the higher the affinity, the smaller the mutation rate, and vice-versa [29]. Also, the immune network theory brings an explanation for the structure (architecture) and dynamics of immune cells and molecules.

By observing the essence of the immune system processes and their computational counterpart Artificial Immune System (AIS), it is possible to find several features that make them applicable to different types of tasks. For instance, Ag-Ab interactions are intrinsically a pattern recognition process, and clonal selection and affinity maturation are akin to an evolutionary search mechanism. The immune network theory adds another sophistication level to AIS by embedding a network structure to a system that was originally composed of separated individual components. When connections are added to the components of the system, pre-defined communication pathways are created and can be subjected to varying weights. By using these immune inspirations, it is possible to design algorithms for solving a vast array of problems, such as autonomous navigation, vehicle routing, clustering, classification, pattern recognition, and anomaly detection [30], [33]–[35].

B. aiNet: An Artificial Immune Network Model

The Artificial Immune Network model called aiNet is an algorithm inspired by the immune network theory aimed at clustering spatial data [29], [36]. aiNet takes as inspiration the pattern recognition of antigens by antibodies, the clonal expansion of high-affinity cells, the affinity proportional mutation of antibodies, the maintenance of high-affinity cells as immune memories, and the immune network theory that explains structural properties of immune cell repertoires.

In the aiNet metaphor, antigens are the input data (objects) while antibodies are the prototypes representing the immune network internal representations of the antigens, learnt from the input data. For the algorithm, antigens and antibodies are represented by N-dimensional vectors, therefore, Ag-Ab recognition is calculated

using a similarity or dissimilarity measure [28]. The evaluation of the affinity between Ag and Ab is of paramount importance for the algorithm training process. Biologically, Ag-Ab recognition is based on the complementarity of their shapes, but for engineering purposes affinity can be measured either with similarity or dissimilarity measures [29].

Algorithm 1 presents a pseudo-code of the aiNet learning algorithm and its main steps. The algorithm works as follows. An initial set of antibodies Ab and a matrix of memory cells M are created, serving the purpose of maintaining sets of prototypes that will represent clusters of data in the available datasets. After initialization, an iterative search process starts until a certain number of iterations has been run. Within this loop, a number of steps occur. First, a partial random population is created and added to Ab. Then, for each input object (Ag), its affinity with all prototypes (Abs) is calculated and the N_b highest affinity ones are selected and cloned proportional to affinity (the higher the affinity, the larger the clone size) and mutated inversely proportional to affinity (the lower the affinity, the higher the mutation rate). A number $\zeta\%$ of the highest affinity mutated clones are selected, the redundant ones, based on a similarity threshold ϵ, eliminated, and those whose affinity with the antigen are smaller than σs are eliminated. After all these steps are performed for each input object, the remaining prototypes are added to the set of M memory cells, and another suppression step removes redundancy within M.

Algorithm 1 The aiNet learning algorithm.

initialize the antibody set Ab
initialize the memory matrix M
while stopping criterion is not met **do**
    initialize a random population with size m and concatenate it with Ab
    for each input object **do**
        calculate its affinity with every member of Ab
        select the N_b highest affinity antibodies
        clone the N_b antibodies proportional to their affinity
        mutate the clone inversely proportionally to their affinity
        select the $\zeta\%$ highest affinity clones
        for each clone in the selected clone set **do**
            if aff > σ_p **then remove** (prune) it from the set of selected clones
            end if
            if aff < σ_s **then remove** (suppression) it from the set of selected clones
            end if
        end for
    end for
    calculate the affinity between all objects in M and suppress those with affinity smaller than σ_s
    replace Ab with M
end while

The main parameters necessary to run the algorithm are:

- max_{it}: maximal number of iterations;
- N_b: number of antibodies to be selected for cloning;
- N_c: multiplier of the number of clones to be generated;
- $\zeta\%$: percentage of antibodies to be selected;
- σ_p: pruning threshold;
- σ_s: suppression threshold used to control redundancy.

At the end of aiNet training, the memory matrix M generated in the last iteration will contain the prototypes found based on the learning from the input data. From this matrix, it is possible to calculate the affinity among its antibodies and find groups of prototypes representing groups of data. The division into groups, also known as clustering, will be performed by applying the Minimal Spanning Tree (MST) [37] followed by a pruning method for inconsistent edges. The MST together with the pruning method will allow the separation of the data into cohesive groups, that is, groups with elements closer to one another [38], [39]. The idea is very simple. After building the MST among all memory antibodies, remove those edges whose weight are significantly larger than the average of nearby edge weights on both sides of the edge.

The aiNet dynamics followed by the application of the MST pruning method described above, makes it a clustering algorithm suitable for solving problems in which the clusters present differences in density. This is because aiNet will tend to uniformly place antibodies in regions of the space where the objects in the dataset are located. After that, the MST pruning method will look for links in the MST that are inconsistent and prune these links. Inconsistency here means being significantly longer than those in neighboring regions, what naturally implements a cluster separation method that searches for variations in the density of data in their original space.

C. Text Representation

Text Analysis refers to the process of extracting knowledge from texts based on their content [40]. As a computer is not intrinsically capable of understanding texts, it is necessary to establish an interface between the language of computers and the human language, which is obtained through computable numeric representations. Text Analysis is part of Natural Language Processing (NLP), which aims to study ways to model human language for computational purposes, thus allowing computers to be able to understand the texts to be analyzed [5], [41].

Over the past years, many works have been proposed involving the application of deep neural networks to text analysis and NLP [42], [43]. Despite that, not so many papers have addressed the problem of comparing the more traditional methods for text structuring (or representation) among themselves and with those based on deep network architectures. Useful review works in this direction include the papers [44]–[47].

The present paper investigates different text representation schemes commonly found in the literature: N-Gram, through Bag of Words (BoW); Linguistic Inquiry and Word Count (LIWC); Part of Speech Tagging (PoS Tagging); MRC; Doc2Vec; Word2Vec; and SBERT. These will be employed to structure text documents to be used by the aiNet clustering algorithm. This section provides a brief review of these text structuring methods.

1. BoW

N-Gram is a simple model used in natural language processing to represent textual data where every sequence of N tokens is considered a new feature [48]. Sequences with a size of one can either be referred to as a Unigram or as a Bag of Words [49]. To simplify, it will be called Bag Of Words (BoW) in this paper. While the Bag of Words is the most popular model, models using greater values of N are extremely useful for text prediction, translation techniques and search engines [48], making them more versatile.

This technique consists of creating a dictionary from the sentences used as input disregarding grammar and context. BoW is often used in conjunction with other pre-processing techniques to remove meaningless words, such as *stopwords*, and standardize the input data by removing special characters and keeping all words in uppercase or

lowercase. After these steps, it is necessary to determine the weight of each feature, formed by *N* sequential tokens, in the document, and this was performed here using the *Term Frequency Inverse Document Frequency* (TF-IDF) method.

TF-IDF is a statistical measure that determines the importance (weight) of each sequence of *N* words in the analyzed documents. This measure is calculated using the relative frequency of each sequence in the analyzed document in relation to the inverse of the number of documents that have the word being evaluated [50]:

$$TF(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad (1)$$

$$IDF(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|} \quad (2)$$

$$TFIDF(t,d,D) = TF(t,d) \cdot IDF(t,D) \quad (3)$$

where $TF(t, d)$ is the relative frequency of term t within document d, $f_{t,d}$ is the number of times the sequence of N terms t occurs in document d, $IDF(t, D)$ is the inverse document frequency, and N is the total number of documents in the corpus D. The higher the TF-IDF value, the greater the relevance of a sequence in the document [51].

2. LIWC

The *Linguistic Inquiry and Word Count* (LIWC) [12] is a textual analysis tool composed of: four categories of general descriptors (total word count, number of words per sentence, percentage of words captured by the dictionary, and percentage of words with more than six letters); seven categories of personal concern (e.g. *work*, *home*, *leisure activities*); three speech categories (consent, e.g. *agree*, *OK*, *yes*; onomatopoeia, e.g. *Er*, *hm*, *umm*; fillers, e.g. *so*, *such as*, *is*, *hum*, *well*); and 12 punctuation categories (e.g. *dots*, *commas*, *etc*). In addition, it has 22 standardized linguistic dimensions (e.g. *the percentage of words in the text that are pronouns*, *articles*, *auxiliary verbs*, *etc.*) and 32 psychological constructor word categories (e.g. *affect*, *cognition*, *biological processes*) [12]. It should be noted that LIWC extracts meta-attributes from a document rather than representing the document by its words, like BoW does.

3. Part of Speech Tagging

Part of Speech Tagging, called here sTagger, originally written by Kristina Toutanova [52], is a Part-of-Speech (POS) tool whose function is to assign each word of the text a tag, such as noun, verb, adjective, etc. In the case of sTagger, its main differential comes from the use of a bidirectional dependency network to predict tagged sequence of words. This bidirectional approach allows it to better extract words' interactions and conditioning features [52].

When structuring a document via sTagger, a count is made of the number of words in each tag. Thus, at the end of the structuring process, there is a matrix in which each attribute refers to a tag [13], [52], [53].

4. MRC

The MRC is a machine usable dictionary of psycholinguistic information containing 150,837 words, where each word is composed of up to 26 linguistic and psycholinguistic attributes. The attributes are obtained from publicly available sources and structured in a single dictionary [14]. The MRC representation consists of mapping words contained in the dictionary into a vector representing the 26 attributes mentioned before.

5. Word2Vec

Word2Vec is a family of algorithms and models that are used to learn *word embedding* from texts and the relationships between words. A word embedding is a representation of a word that encodes its meaning in a real-valued lower-dimensional vector, where the representation of words with similar meaning are closer together in the vector space.

The application of word embedding grants the *Word2Vec* model the ability to generate real-valued dense vectors for each word that is capable of capturing each linguistic regularity and linear relationships, allowing those vectors to be applied to mathematics operations like + and -.

An example of the linguistic regularities and their math properties is subtracting the vector representation of the word *man* from the vector representation of the word *king*, resulting in a vector that is close to the vector representation of the word *queen* [54].

6. Doc2Vec

Doc2Vec is a set of paragraph embedding models, inspired by the Word2Vec model, but with emphasis on documents, and that produces better results than averaging all the word vectors in a document.

A paragraph embedding is a representation of a variable-length text, such as documents, sentences and paragraphs, by real-valued vectors with fixed-length features [55]. The main difference between Word2Vec and Doc2Vec is that besides the word vectors generated by both models, Doc2Vec also has a single shared paragraph embedding which allows to better represent the document and its meaning [55].

The Doc2Vec model learns the paragraph embedding of a text by training to predict the vector representation for each word in a document in conjunction with a vector representing the paragraph, the paragraph vector. The predict task of the model concatenates the paragraph vector with word vectors to predict the next word in the context. The outcome of the learning task is a model whose vectors are capable of representing documents in a vector space.

7. SBERT

SBERT is a sentence embedding representation model built on top of BERT [56] and RoBERTa models [57]. These models present state-of-the-art performances for many text mining tasks, but have poorer performances when used for semantic-similarity, making them unsuitable for clustering tasks [58].

Since this embedding was created with the goal of extending the state-of-the-art results provided by those models for sentence embedding generation, this representation makes use of an elegant modification on the BERT/RoBERTa models by adding a pooling operation to its output. In order to provide a more contextualized and semantically meaningful embedding, the BERT/RoBERTa are first fine-tuned with siamese and triplet networks [58].

The SBERT representation can be generated by using many of the pre-trained models available in public repositories, like Hugging Face Hub. Since this representation uses pre-trained models, the quality of the embeddings generated may vary depending on the model used.

III. Performance Assessment

The goal of this paper is to investigate how different text representation methods influence the clustering performance of aiNet. To do so, three types of text representations were chosen: one standard vector space model (BoW); three grammar-based models (LIWC, sTagger, MRC); and two word embedding models (Word2Vec and Doc2Vec). Two clustering internal measures were selected for comparison: the Dunn (DU) index and the Davies Bouldin (DB) index. The methodology used, results obtained, and discussions are presented in this section.

For this research the *stsb-roberta-large* model [58] was chosen since some preliminary tests with four different models indicated that the *stsb-roberta-large* consistently generated the best results for all datasets.

A. Methodology

This subsection describes the experimental methodology employed. It starts by providing some distinctions among the representations and then follows with the hyperparametrization of aiNet, Word2Vec and Doc2Vec. Then, the datasets chosen are summarized and the evaluation measures described.

1. Some Comments on the Selected Representations

The three classes of text representations are considerably different from one another. BoW works by finding and weighing tokens that are expected to have a high discriminating capability among the text categories, but usually results in very high dimensional feature vectors. Grammar-based representations, such as LIWC, sTagger and MRC, are characterized by a limited number of word categories, but privilege a low dimensional representation of word categories in detriment of the context. Word embeddings, like Word2Vec and Doc2Vec, by contrast, try to extract the semantic meaning of the texts by representing the words by means of word vectors that are expected to capture the context of each word or document. SBERT generates fixed-size vector representations of sentences or short texts, extending the concept of word embeddings to the sentence level. These representation schemes will be used and compared here.

2. Some Comments on the Pre-Processing Step

In recent years, questions have been raised about the need of a pre-processing step for generating text representations. This is because most of the recently developed representations, like SBERT, are based on deep neural networks, for which the removal of any word can have an impact on the contextual and semantic understanding of the model, potentially leading to worse representations [59], [60].

3. aiNet's Hyperparameters

The aiNet parameters were chosen based on an iterative process aimed to maximize the selected evaluation metrics while making it possible to investigate how different representations behave when paired with the model, assessing their strengths and weaknesses. To maintain consistency when comparing different representation methods, most of the aiNet hyperparameters were fixed for all representations and datasets used, as follows:

- max_{it} = 50;
- N_b = 500;
- N_c = 40;
- $\zeta\%$ = 10%;
- σ_s = 0.05.

This consistency is important because if aiNet had to be fine tuned for each representation it would be very difficult to compare the results and understand how each representation method influences the clustering results.

However, it is well known that some representation models, like BoW, generate high dimensional datasets, and calculating the similarity among the immune cells and between them and the input objects may require some tuning. Some preliminary experiments showed that higher values for the pruning threshold σ_p should be adopted for higher dimensional spaces, so the defined value for BoW, Word2Vec, and Doc2Vec was 0.9, while the remaining representations had σ_p = 0.5.

4. Word2Vec and Doc2Vec Hyper-Parameters

Word2Vec and Doc2Vec are the only representations that require hyper-parameter tuning. Since both models are based on the same algorithm, they share most of their parameters and can be trained using the same package, a very popular library called *Gensim* [61]:

- *vector_size* = 50;
- *window* = 5;
- *min_count* = 2;
- *epochs* = 40;
- *negative* = 5.

The Word2Vec was trained using the preprocessed sentences of the datasets and *sg* = 0. For the Doc2Vec *dm* = 0 (PV-DBOW).

5. Datasets

To test aiNet's clustering capability for different text representation schemes, four datasets from the literature were selected:

- *Sentiment Labelled Collection*: a collection of 3 datasets from 3 different websites (Amazon, Yelp and IMDB) containing users' reviews. Each dataset has 1,000 objects with 500 positive reviews and 500 negative ones. The datasets are available at https://archive. ics.uci. edu/ml/datasets/sentiment+labelled+sentences.
- *SMS Spam Data*: a dataset collected by [62] containing 5,572 messages (4,825 ham and 747 spam). The dataset is available at https://archive.ics.uci.edu/ml/ datasets/SMS+Spam+Collection.

It is important to highlight that although the selected datasets are textual, each one of them represent a different type of text with its own characteristics. The SMS Spam Data is considered a short text dataset due to its character limitation and informal nature, making it likely the use of *slangs* and shorter texts that sometimes need to be combined to present the whole context [63]. The Sentiment Labelled Collection, by contrast, is composed of reviews extracted from three different websites with substantially different products, but grouped based on language and semantics. Reviews are different from text messages since their nature revolves around more descriptive texts and a semantics that expresses how someone feels about the reviewed product.

The difference among the selected datasets can have an impact on the representations generated. For instance, models like the Bag of Words are likely to be more sensitive to the SMS Spam Collection because its shorter length could generate a more sparse representation of each text.

6. Evaluation Measures

The Dunn and Davies-Bouldin indices [3] were used to assess the quality of the clusters obtained [3]. To calculate them, it is necessary to evaluate the cohesion (compactness) and separation of each cluster [64], what can be performed using intra (Eq. (4)) and inter-cluster distances (Eq. (5)):

$$Intra(g_i) = max_{x,y \in g_i}\{d(x,y)\} \tag{4}$$

$$Inter(g_i, g_j) = \frac{1}{|g_i| \cdot |g_j|}\sum d(x,y) | x \in g_i, y \in g_j \tag{5}$$

where g_i refers to group i, $|g_i|$ is the number of objects in group i, and $d(x, y)$ is a distance measure between objects x and y.

7. Dunn Index (DU)

The Dunn Index combines both the inter- and intra-cluster distance to provide a cluster quality measure. It ranges over the interval [0, ∞], where the higher the values, the more cohesive and separated the clusters [65]:

$$DU(g) = \min_{i=1,\ldots,k}\left\{\min_{j=1,\ldots,k; j\neq i}\left\{\frac{\text{Inter}\,(g_i, g_j)}{\max\limits_{l=1,\ldots,k}\{\text{Intra}\,(g_l)\}}\right\}\right\} \tag{6}$$

where g is the resultant clustering, k is the number of clusters, g_i is a cluster from the dataset, and *Inter*(.) and *Intra*(.) are the inter- and intracluster measures defined previously.

8. Davies-Bouldin Index (DB)

Similarly to the DU, the Davies-Bouldin Index [66] also uses the inter- and intra-cluster distances to determine its value, but combining these measures in a different way:

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{i \neq j} \left(\frac{\text{Intra}\,(g_i) + \text{Intra}\,(g_j)}{\text{Inter}\,(g_i, g_j)} \right) \tag{7}$$

where g is the resultant clustering, k is the number of clusters, g_i is a cluster from the dataset, and *Inter*(.) and *Intra*(.) are the inter- and intracluster measures defined previously.

The DB index measures the similarity between each cluster [67], [68] and it varies in the range [0, ∞], where the lower the values, the better the quality.

9. Experimental Results

The experiments were performed in Python with the use of third party libraries such as *Numpy*, *Spacy*, *Scikit-learn*, *Gensim*, *Spacy-stanza* and *Pandas*. With all the representations, hyperparameters, datasets, and evaluation measures defined, the experiments were organized as follows:

- Each text representation was generated from a dataset after going through a processing pipeline adapted to the specificity of each representation. The structured texts were then used to train the aiNet model.
- As the aiNet relies heavily on the Ag-Ab affinity and considering that some of the text representations generate high-dimensional input vectors, the *cosine similarity* was chosen as the affinity measure in all experiments.
- Based on the clusters determined by aiNet, its performance was evaluated using the Dunn Index (DU) and the Davies-Bouldin Index (DB) for each representation used and for all selected datasets.
- Since aiNet is a non-deterministic algorithm, 10 experiments were performed for each text representation and the results presented are the average and standard deviation of the 10 results.

B. Results and Discussion

Table I summarizes the clustering results of the seven text structuring methods when used in conjunction with aiNet. By analyzing all results it is possible to observe similar trends between each representation across all four datasets, with the Bag of Words and MRC consistently presenting the best and the worst results for DU and DB, respectively. Another similarity is related with the total number of clusters identified, as presented in Fig. 1. It can be observed that the number of clusters resultant from each representation follows the same pattern.

While the state of the art representation, SBERT, presented only the second best results for both indices for the first three datasets, it presented, by a vast margin, the best results for the SMS Spam Collection. Another aspect to observe is that SBERT constantly generated the smallest number of clusters, closer to the number of classes of each dataset.

Considering the different sets of text representations studied in this paper, high dimensional feature vectors, grammar-based, word embedding, and sentence embedding, it is possible to note similar results between representations that belong to the same set, for example Doc2Vec and Word2Vec present similar results for both measures (DU and DB).

Interestingly, Bag of Words performed competitively even when compared with the state of the art, SBERT, and consistently outperformed word embedding and grammar-based representations, achieving the second best overall result with the Sentiment Labelled

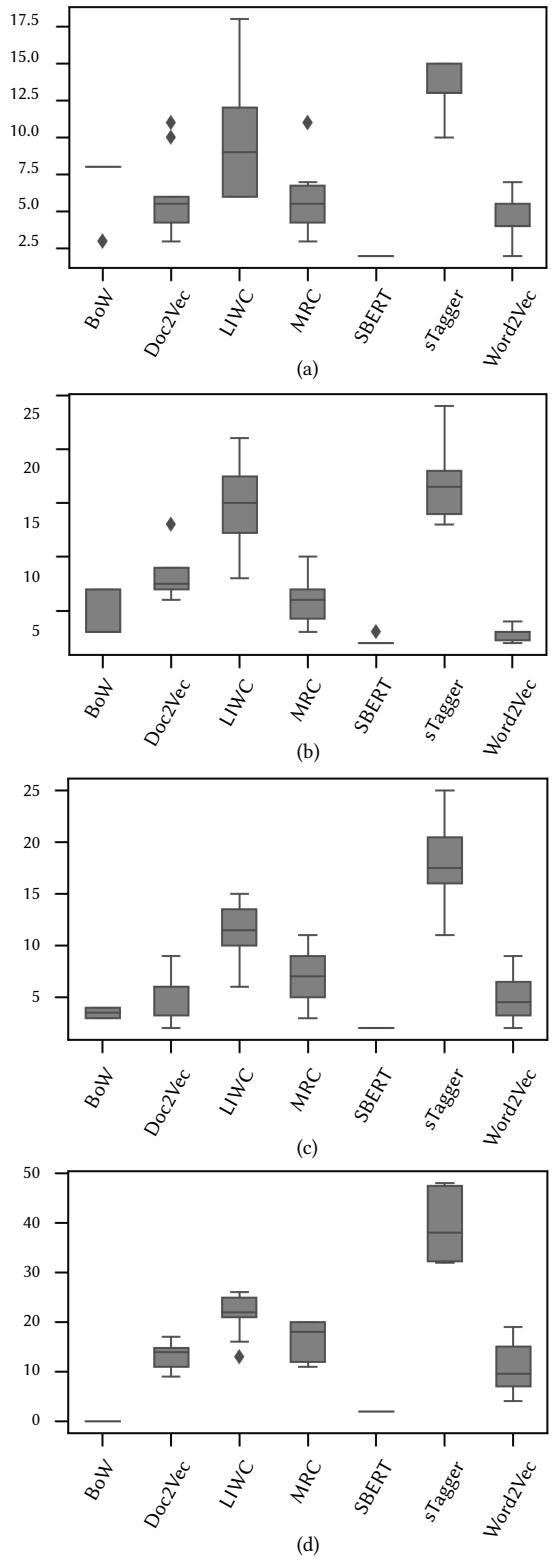

Fig. 1. Boxplots presenting the number of clusters per representation over 10 executions. (a) Amazon Labelled Reviews. (b) IMDB Labelled Reviews. (c) Yelp Labelled Reviews. (d) SMS Spam Collection.

Review Datasets. Although the BoW model in general presented very good results for the Sentiment Labelled Review Datasets, it also presented a very undesired behaviour when paired with the SMS Spam Collection Dataset. In this scenario, despite its higher value of σ_p, the aiNet paired with BoW was unable to find more than one cluster, reason why its performance was zeroed.

TABLE I. Performance Evaluation of the Seven Text Structuring Methods (TSM) in the Four Datasets Chosen, Detaching the Best Performance for Each Dataset. DU: Dunn Index; DB: Davies Bouldin Index. The SBERT Representation, As Previously Mentioned, Was Generated Using the Sentence-transformers/stsb-large-model Available at Https://huggingface.co/sentence-transformers/ Stsb-roberta-large

	TSM	DU	DB
Amazon Labelled Reviews	**BoW**	**0.98 ± 0.00**	**1.95 ± 0.13**
	Doc2Vec	0.29 ± 0.02	3.73 ± 0.55
	LIWC	0.32 ± 0.03	3.65 ± 0.20
	MRC	0.19 ± 0.07	4.71 ± 0.57
	SBERT	0.77 ± 0.02	2.49 ± 0.05
	sTagger	0.26 ± 0.03	3.70 ± 0.20
	Word2Vec	0.26 ± 0.07	4.53 ± 0.89
IMDB Labelled Reviews	**BoW**	**0.99 ± 0.00**	**2.01 ± 0.00**
	Doc2Vec	0.25 ± 0.06	4.91 ± 0.61
	LIWC	0.31 ± 0.03	3.79 ± 0.31
	MRC	0.18 ± 0.05	5.17 ± 0.53
	SBERT	0.76 ± 0.01	2.48 ± 0.14
	sTagger	0.23 ± 0.02	3.23 ± 0.24
	Word2Vec	0.23 ± 0.05	4.26 ± 0.47
Yelp Labelled Reviews	**BoW**	**0.98 ± 0.00**	**2.02 ± 0.00**
	Doc2Vec	0.30 ± 0.05	3.76 ± 0.28
	LIWC	0.32 ± 0.04	3.46 ± 0.12
	MRC	0.21 ± 0.05	4.92 ± 0.55
	SBERT	0.77 ± 0.01	2.52 ± 0.03
	sTagger	0.27 ± 0.02	3.15 ± 0.15
	Word2Vec	0.26 ± 0.07	4.85 ± 1.21
SMS Spam Collection	BoW	0.00 ± 0.00	0.00 ± 0.00
	Doc2Vec	0.33 ± 0.02	3.50 ± 0.28
	LIWC	0.25 ± 0.00	4.43 ± 0.13
	MRC	0.09 ± 0.02	7.30 ± 0.43
	SBERT	**0.77 ± 0.01**	**2.57 ± 0.03**
	sTagger	0.14 ± 0.01	4.37 ± 0.15
	Word2Vec	0.09 ± 0.02	7.30 ± 0.43

Further investigations showed that the higher number of objects in the SMS Spam Collection resulted in a very sparse representation for BoW, with over 7,200 dimensions, almost a three times increase when compared with the BoW dimension for the Amazon Labelled Reviews. The sparsity found with this representation posed a challenge for the aiNet model since the Ag-Ab interactions ended up presenting very high values and thus all antibodies were consistently eliminated every iteration, resulting in an empty memory matrix after the training process.

In addition to the higher dimension, the SMS Spam Collection also presented a very unbalanced proportion between its original clusters, with the spam class having six times more objects than the other class, making it much more difficult to extract patterns from each cluster. This proportion made it difficult for the SBERT representation to detect more than one cluster in a couple of executions, which indicated that a fine tuning of the algorithm can lead to an even better performance for this representation.

Another point that can be observed when analyzing the results of grammar-based representations is that their dictionaries have a significant impact on the final representation dimensionality with each representation using different dictionaries, each with its own categories. Due to the grammar-based representations dependency on a predefined dictionary, the final representation is subjected to the words available in the dictionary and mismatches of words between the texts and dictionaries can occur. Such scenario becomes evident when older dictionaries are paired with modern texts, such as internet discussions and reviews, given that it does not account for today's dialects and *slangs*. Usually, some of the grammar-based representations have a specific *category* that is used to account the mismatches, such as LIWC, but not all of them have it, as is the case for MRC.

The issues mentioned above can be observed when assessing the results of MRC, which presented the worst results for all datasets. The dictionary used by MRC was released in 1988 and has several relevant psychological attributes that are difficult to be synthesized and some of them do not have value for all the words contained in its dictionary and some of the words are not contained in it. Due to the complexity of the attributes and the date the dictionary was created, the probability of a word from the texts of the chosen datasets being present and having values for all the features is low, causing words not to have a significant value or to have a sparse representation, thus impacting its performance. It is also possible to infer from the results that the LIWC representation, which has a more complete and more recent dictionary, created in 2015, that it can also account for words that are not present in it, has better metric values and greater number of clusters of the grammar-based category.

The results also emphasize that the SMS database is the most complex to represent, resulting in lower metric values for most representations, with the exception of Doc2Vec and Word2Vec, which presented their best results among all datasets.

Fig. 1 shows the boxplot of the number of clusters found by aiNet for each of the four datasets over the 10 runs performed. Note that all datasets are originally divided into two classes, but the class labels are not used to train aiNet. It is a general tendency that sTagger generates more clusters than the other approaches, followed by LIWC. Also, it was noted that Word2Vec and Doc2Vec present similar behaviors with small numbers of clusters.

IV. Conclusions and Future Trends

This paper aimed at investigating the influence of seven different text structuring methods to be used in conjunction with the aiNet clustering algorithm. These methods fall into four categories: vector space models, grammar-based models, word embeddings, and sentence embeddings. Each category has a specific form of structuring the text, capturing or not information like syntax and context. Performance evaluation was made using four datasets from the literature, and internal clustering measures (Dunn and Davies Bouldin indices).

After running a number of experiments and analyzing the results, it was possible to observe that the aiNet's pruning threshold is sensitive to the dimensionality of the representation, especially those with more sparse representations, like the Bag of Words (BoW) model.

Considering all the results obtained in this paper, the state-of-the-art model, SBERT, consistently presented good results on all selected datasets, while other distributed representations, Doc2Vec and Word2Vec, did not perform as well, especially when paired with the Sentiment Labelled Review Dataset. The results suggest that this type of representation performs better with datasets containing a larger number of objects, that is, a larger variety of words. This observation is in contrast with the remaining representations, which performed worse when used with the SMS Spam Collection.

Although the BoW representation is the simplest in terms of generation when compared with the others studied, its results were fairly competitive, especially with the state of the art representation. While it is true that this representation was unable to present any

result for the SMS Spam Collection, this reinforces the BoW's main weakness: its dimensionalty (the larger number of objects provides a greater variety of words, which greatly increases the dimensionality of this representation making it extremely sparse and decreasing the effectiveness of similarity techniques that are intensively used by the aiNet algorithm.)

The results provided interesting insights about the peculiarity of each type of text representation. It is clear the need of running new experiments with larger datasets to further evaluate and improve the performance of the aiNet algorithm. Another point of improvement is the use of other high dimensional representations to further evaluate the impact of very sparse data matrices when paired with the aiNet algorithm. Finally, the results presented are relevant since they can be used as a baseline to fine tune the aiNet algorithm for each representation studied.

Acknowledgment

This work was financially supported by FAPESP, CNPq and MackPesquisa.

References

[1] C. S. Kumar, R. Santhosh, "Effective information retrieval and feature minimization technique for semantic web data," *Computers & Electrical Engineering*, vol. 81, p. 106518, 2020.

[2] S. S. Tandel, A. Jamadar, S. Dudugu, "A survey on text mining techniques," in *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, 2019, pp. 1022–1026, IEEE.

[3] L. N. de Castro, D. G. Ferrari, *Introdução à mineração de dados*. Saraiva Educação SA, 2017.

[4] T. Jo, "Text mining: Studies in big data," 2019.

[5] K. Chowdhary, K. Chowdhary, "Natural language processing," *Fundamentals of artificial intelligence*, pp. 603–649, 2020.

[6] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, M. DATA, "Practical machine learning tools and techniques," in *Data Mining*, vol. 2, 2005.

[7] Y. HaCohen-Kerner, D. Miller, Y. Yigal, "The influence of preprocessing on text classification using a bag- of-words representation," *PloS one*, vol. 15, no. 5, p. e0232525, 2020.

[8] G. Miner, J. Elder IV, A. Fast, T. Hill, R. Nisbet, D. Delen, *Practical text mining and statistical analysis for non-structured text data applications*. Academic Press, 2012.

[9] W. A. Qader, M. M. Ameen, B. I. Ahmed, "An overview of bag of words;importance, implementation, applications, and challenges," in *2019 International Engineering Conference (IEC)*, 2019, pp. 200–204.

[10] D. Yan, K. Li, S. Gu, L. Yang, "Network-based bag-of- words model for text classification," *IEEE Access*, vol. 8, pp. 82641–82652, 2020.

[11] W. A. Qader, M. M. Ameen, B. I. Ahmed, "An overview of bag of words; importance, implementation, applications, and challenges," in *2019 international engineering conference (IEC)*, 2019, pp. 200–204, IEEE.

[12] J. W. Pennebaker, M. E. Francis, R. J. Booth, "Linguistic inquiry and word count: Liwc 2001," *Mahway: Lawrence Erlbaum Associates*, vol. 71, no. 2001, p. 2001, 2001.

[13] A. Chiche, B. Yitagesu, "Part of speech tagging: a systematic review of deep learning and machine learning approaches," *Journal of Big Data*, vol. 9, no. 1, pp. 1–25, 2022.

[14] M. D. Wilson, "*MRC Psycholinguistic Database: Machine Usable Dictionary: Version 2.00*," *Behavior Research Methods, Instruments, & Computers*, vol. 20, pp. 6–10, 1988.

[15] J. Lastra-Díaz, J. Goikoetxea, M. A. Hadj Taieb, A. Garcia-Serrano, M. Ben Aouicha, E. Agirre, "A reproducible survey on word embeddings and ontology-based methods for word similarity: Linear combinations outperform the state of the art," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 645–665, 2019, doi: 10.1016/j.engappai.2019.07.010.

[16] U. Naseem, I. Razzak, S. K. Khan, M. Prasad, "A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 5, pp. 1–35, 2021.

[17] F. Almeida, G. Xexéo, "Word embeddings: A survey," *arXiv preprint arXiv:1901.09069*, 2019.

[18] B. Li, H. Zhou, J. He, M. Wang, Y. Yang, L. Li, "On the sentence embeddings from pre-trained language models," *arXiv preprint arXiv:2011.05864*, 2020.

[19] M. N. Moghadasi, Y. Zhuang, "Sent2vec: A new sentence embedding representation with sentimental semantic," in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 4672–4680.

[20] T. Gao, X. Yao, D. Chen, "Simcse: Simple contrastive learning of sentence embeddings," *arXiv preprint arXiv:2104.08821*, 2021.

[21] T. Jiang, J. Jiao, S. Huang, Z. Zhang, D. Wang, F. Zhuang, F. Wei, H. Huang, D. Deng, Q. Zhang, "PromptBERT: Improving BERT sentence embeddings with prompts," *arXiv preprint arXiv:2201.04337*, 2022.

[22] X. Zhu, T. Li, G. De Melo, "Exploring semantic properties of sentence embeddings," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 632–637.

[23] M. K. Mishra, J. Viradiya, "Survey of sentence embedding methods," *International Journal of Applied Science and Computations*, vol. 6, no. 3, pp. 592–592, 2019.

[24] S. A. Hofmeyr, S. Forrest, "Immunity by design: An artificial immune system," in *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 2*, 1999, pp. 1289–1296, Citeseer.

[25] F. A. González, D. Dasgupta, "Anomaly detection using real-valued negative selection," *Genetic Programming and Evolvable Machines*, vol. 4, pp. 383– 403, 2003.

[26] E. Bendiab, M. K. Kholladi, "The negative selection algorithm: a supervised learning approach for skin detection and classification," *International Journal of Computer Science and Network Security*, vol. 10, pp. 86– 92, 2010.

[27] M. Ayara, J. Timmis, R. de Lemos, L. N. de Castro, R. Duncan, "Negative selection: How to generate detectors," in *Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS)*, vol. 1, 2002, pp. 89–98, University of Kent at Canterbury Printing Unit Canterbury, UK.

[28] J. Timmis, M. Neal, J. Hunt, "An artificial immune system for data analysis," *Biosystems*, vol. 55, no. 1-3, pp. 143–150, 2000.

[29] L. N. de Castro, F. J. Von Zuben, "aiNet: an artificial immune network for data analysis," in *Data mining: a heuristic approach*, IGI Global, 2002, pp. 231–260.

[30] D. Dasgupta, S. Yu, F. Nino, "Recent advances in artificial immune systems: models and applications," *Applied Soft Computing*, vol. 11, no. 2, pp. 1574–1587, 2011.

[31] J. Greensmith, A. Whitbrook, U. Aickelin, "Artificial immune systems," *Handbook of Metaheuristics*, pp. 421– 448, 2010.

[32] J. Timmis, "Artificial immune systems–today and tomorrow," *Natural computing*, vol. 6, no. 1, p. 1, 2007.

[33] N. Bayar, S. Darmoul, S. Hajri-Gabouj, H. Pierreval, "Fault detection, diagnosis and recovery using artificial immune systems: A review," *Engineering Applications of Artificial Intelligence*, vol. 46, pp. 43–57, 2015, doi: https://doi.org/10.1016/j.engappai.2015.08.006.

[34] S. Alhasan, G. Abdul-Salaam, L. Bayor, K. Oliver, "Intrusion detection system based on artificial immune system: A review," in *2021 International Conference on Cyber Security and Internet of Things (ICSIoT)*, 2021, pp. 7–14.

[35] L. N. de Castro, J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer- Verlag UK, 2002.

[36] I. Čisar, S. M. Čisar, B. Popović, K. Kuk, I. Vuković, "Application of artificial immune networks in continuous function optimizations," *Acta Polytechnica Hungarica*, vol. 19, no. 7, pp. 53–164, 2022.

[37] P. C. Pop, "The generalized minimum spanning tree problem: An overview of formulations, solution procedures and latest advances," *European Journal of Operational Research*, vol. 283, no. 1, pp. 1–15, 2020.

[38] D. Cheng, Q. Zhu, J. Huang, Q. Wu, L. Yang, "Clustering with local density peaks-based minimum spanning tree," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 2, pp. 374–387, 2019.

[39] G. Mishra, S. K. Mohanty, "A fast hybrid clustering technique based on local nearest neighbor using minimum spanning tree," *Expert Systems with Applications*, vol. 132, pp. 28–43, 2019.

[40] M. L. Jockers, R. Thalken, *Text analysis with R*. Springer, 2020.

[41] J. Hirschberg, C. D. Manning, "Advances in natural language processing,"

Science, vol. 349, no. 6245, pp. 261–266, 2015.

[42] J. Chai, A. Li, "Deep learning in natural language processing: A state-of-the-art survey," in *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*, 2019, pp. 1–6.

[43] D. W. Otter, J. R. Medina, J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, 2021, doi: 10.1109/TNNLS.2020.2979670.

[44] K. Singh, H. Devi, A. Mahanta, "Document representation techniques and their effect on the document clustering and classification: A review.," *International Journal of Advanced Research in Computer Science*, vol. 8, 2017.

[45] M. H. Ahmed, S. Tiun, N. Omar, N. S. Sani, "Short text clustering algorithms, application and challenges: A survey," *Applied Sciences*, vol. 13, no. 1, p. 342, 2023.

[46] K. Babić, S. Martinčić-Ipšić, A. Meštrović, "Survey of neural text representation models," *Information*, vol. 11, no. 11, p. 511, 2020.

[47] S. A. Farimani, M. V. Jahan, A. Milani Fard, "From text representation to financial market prediction: A literature review," *Information*, vol. 13, no. 10, p. 466, 2022.

[48] G. E. Pibiri, R. Venturini, "Handling massive n-gram datasets efficiently," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 2, pp. 1–41, 2019.

[49] M. Schonlau, N. Guenther, "Text mining using n- grams," *Schonlau, M., Guenther, N. Sucholutsky, I. Text mining using n-gram variables. The Stata Journal*, vol. 17, no. 4, pp. 866–881, 2017.

[50] D. E. Cahyani, I. Patasik, "Performance comparison of TF-IDF and word2vec models for emotion text classification," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 5, pp. 2780–2788, 2021.

[51] M. Das, S. Kamalanathan, P. Alphonse, "A comparative study on tf-idf feature weighting method and its analysis using unstructured dataset.," in *COLINS*, 2021, pp. 98–107.

[52] K. Toutanova, D. Klein, C. D. Manning, Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 2003, pp. 252–259.

[53] J. Awwalu, S. E.-Y. Abdullahi, A. E. Evwiekpaefe, "Parts of speech tagging: a review of techniques," *Fudma Journal of Sciences*, vol. 4, no. 2, pp. 712–721, 2020.

[54] T. Mikolov, K. Chen, G. Corrado, J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[55] Q. Le, T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, 2014, pp. 1188–1196, PMLR.

[56] A. Rogers, O. Kovaleva, A. Rumshisky, "A primer in BERTology: What we know about how BERT works," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 842–866, 2021.

[57] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019.

[58] N. Reimers, I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[59] Z. Rahimi, M. M. Homayounpour, "The impact of preprocessing on word embedding quality: A comparative study," *Language Resources and Evaluation*, vol. 57, no. 1, pp. 257–291, 2023.

[60] K. V. Ghag, K. Shah, "Comparative analysis of effect of stopwords removal on sentiment classification," in *2015 international conference on computer, communication and control (IC4)*, 2015, pp. 1–6, IEEE.

[61] "Gensim." https://github.com/ RaRe-Technologies/gensim.

[62] T. A. Almeida, J. M. G. Hidalgo, A. Yamakami, "Contributions to the study of sms spam filtering: new collection and results," in *Proceedings of the 11th ACM symposium on Document engineering*, 2011, pp. 259–262.

[63] H. Yin, X. Song, S. Yang, G. Huang, J. Li, "Representation learning for short text clustering," in *Web Information Systems Engineering–WISE 2021: 22nd International Conference on Web Information Systems Engineering, WISE 2021, Melbourne, VIC, Australia, October 26–29, 2021, Proceedings, Part II 22*, 2021, pp. 321–335, Springer.

[64] W. Wu, H. Xiong, S. Shekhar, J. He, A.-H. Tan, C.-L. Tan, S.-Y. Sung, "On quantitative evaluation of clustering systems," *Clustering and information retrieval*, pp. 105– 133, 2004.

[65] C.-E. B. Ncir, A. Hamza, W. Bouaguel, "Parallel and scalable dunn index for the validation of big data clusters," *Parallel Computing*, vol. 102, p. 102751, 2021.

[66] D. Davies, D. Bouldin, "A cluster separation measure," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-1, pp. 224 – 227, 05 1979, doi: 10.1109/TPAMI.1979.4766909.

[67] I. F. Ashari, R. Banjarnahor, D. R. Farida, S. P. Aisyah, A. P. Dewi, N. Humaya, *et al.*, "Application of data mining with the k-means clustering method and davies bouldin index for grouping imdb movies," *Journal of Applied Informatics and Computing*, vol. 6, no. 1, pp. 07– 15, 2022.

[68] M. Mughnyanti, S. Efendi, M. Zarlis, "Analysis of determining centroid clustering x-means algorithm with davies-bouldin index evaluation," in *IOP Conference Series: Materials Science and Engineering*, vol. 725, 2020, p. 012128, IOP Publishing.

Matheus A. Ferraria

Matheus is an aspiring student who started his master's degree at Mackenzie Presbyterian University (UPM) back in the first trimester of 2022. His study area is focused on Electrical Engineering and Computing, more specifically Natural Computing and Data Mining. He received his bachelor's degree in computer science from UPM at the end of 2021. His passion for research started during its finals days as a bachelor, after working with Leandro N. de Castro, writing his final paper, where he became fascinated by the nature's complexity and how it can inspire solutions to solve daily recurring problems. Passionate about learning and exploring his most recent academic experience is publishing and presenting an article at DCAI22 and he expects to contribute much more to the computer science community.

Vinicius A. Ferraria

A student inspired by studying the nature and its repercussion who graduated with a Bachelor's degree in Computer Science at Mackenzie Presbyterian University (UPM) in the second semester of 2021 and followed his studies by ingressing on a master's degree in Eletctrical Engineering and Computing in the field of Natural Computing and Data Mining also at UPM. His passion for natural computing started when he watched one of lecture about nature inspiring algorithms and since then he has worked in a couple article surrounding the Immune System and its infinity possibilities, ranging from his final thesis to scientific article published in conferences, the most recent one was an article published at DCAI22 related with an Artificial Immune System.

Leandro N. de Castro

Leandro has a B.Sc., M.Sc., and Ph.D. in Electrical Engineering from the Federal University of Goiás and Unicamp. He also holds an MBA in Strategic Business Management from the Catholic University of Santos. He was a Research Associate at the Computer Laboratory of the University of Kent in Canterbury (2001-2002), a Visiting Professor at the Technological University of Malaysia in 2005, a Visiting Professor at Unicamp (2012), and a Visiting Researcher at the University of Salamanca (2014). He was a research professor at the Master's Program in Informatics at Unisantos (2003-2008), and a research professor at the Graduate Program in Electrical Engineering and Computing at Universidade Presbiteriana Mackenzie (2008-2022). His main lines of research are Natural Computing and Machine Learning, with applications in Intelligent Data Analysis and Optimization. Leandro is the author of four academic books and has more than 250 papers published in national and international journals and conferences. He was the proponent and Editor-in-Chief of the International Journal of Natural Computing Research (IJNCR) from 2010 to 2015, published by IGI-Global. Leandro also has extensive entrepreneurial and leadership experience, having already participated in the founding of four Artificial Intelligence startups and invested, as an angel investor, in three others. He is currently a Visiting Professor at the Faculty of Technology at Unicamp and an Artificial Intelligence and Data Science Professor at the Florida Gulf Coast University.