


# Pipelining Semantic Expansion and Noise Filtering for Sentiment Analysis of Short Documents – CluSent Method


Felipe Viegas   [ Universidade Federal de Minas Gerais | [frviegas@dcc.ufmg.br](mailto:frviegas@dcc.ufmg.br) ]


Sergio Canuto  [ Instituto Federal de Goiás | [sergio.canuto@ifg.edu.br](mailto:sergio.canuto@ifg.edu.br) ]

Washington Cunha  [ Universidade Federal de Minas Gerais | [washingtoncunha@dcc.ufmg.br](mailto:washingtoncunha@dcc.ufmg.br) ]

Celso França  [ Universidade Federal de Minas Gerais | [celsofranca@dcc.ufmg.br](mailto:celsofranca@dcc.ufmg.br) ]


Claudio Valiense  [ Universidade Federal de Minas Gerais | [claudio.valiense@dcc.ufmg.br](mailto:claudio.valiense@dcc.ufmg.br) ]

Guilherme Fonseca  [ Universidade Federal de São João del-Rei | [guilhermefonseca8426@aluno.ufsj.edu.br](mailto:guilhermefonseca8426@aluno.ufsj.edu.br) ]

Ana Machado  [ Universidade Federal de São João del-Rei | [anaclaudiamachado211@aluno.ufsj.edu.br](mailto:anaclaudiamachado211@aluno.ufsj.edu.br) ]

Leonardo Rocha  [ Universidade Federal de São João del-Rei | [lrocha@ufsj.edu.br](mailto:lrocha@ufsj.edu.br) ]

Marcos André Gonçalves  [ Universidade Federal de Minas Gerais | [mgoncalv@dcc.ufmg.br](mailto:mgoncalv@dcc.ufmg.br) ]

 Department of Computer Science, Federal University of Minas Gerais, Av. Pres. Antônio Carlos, 6627 - Pampulha, Belo Horizonte - MG, 31270-901, Brazil.

**Received:** 02 February 2024 • **Accepted:** 03 June 2024 • **Published:** 11 June 2024

**Abstract:** The challenge of constructing effective sentiment models is exacerbated by a lack of sufficient information, particularly in short texts. Enhancing short texts with semantic relationships becomes crucial for capturing affective nuances and improving model efficacy, albeit with the potential drawback of introducing noise. This article introduces a novel approach, **CluSent**, designed for customized dataset-oriented sentiment analysis. CluSent capitalizes on the CluWords concept, a proposed powerful representation of semantically related words. To address the issues of information scarcity and noise, CluSent addresses these challenges: (i) leveraging the semantic neighborhood of pre-trained word embedding representations to enrich document representation and (ii) introducing dataset-specific filtering and weighting mechanisms to manage noise. These mechanisms utilize part-of-speech and polarity/intensity information from lexicons. In an extensive experimental evaluation spanning 19 datasets and five state-of-the-art baselines, including modern transformer architectures, CluSent emerged as the superior method in the majority of scenarios (28 out of 38 possibilities), demonstrating noteworthy performance gains of up to 14% over the strongest baselines.

**Keywords:** Sentiment Analysis, Classification, Natural Language Processing

## 1 Introduction

Sentiment analysis has been one of the most active fields in NLP due to the value of revealing how people feel about a particular product, service, or topic. Strategies for classifying sentiments can be roughly divided into supervised and unsupervised. While supervised strategies train robust classification models [Socher *et al.*, 2013] using manually labeled training data from the specific domain of interest, unsupervised strategies exploit sentiment lexicons combined with grammar rules (negation, intensifiers) to infer the associated class (positive or negative) with a document.

In the unsupervised realm, lexicon limitations, such as the *coverage problem* [Arkin *et al.*, 2018] that has to do with the number of dataset's words covered by a lexicon, may hamper the potential of such unsupervised strategies. Some automatic techniques to expand the lexicon vocabulary can ameliorate the coverage problem, but it is not easy to define universally effective sentiment lexicons to cover words from many different domains [Wang *et al.*, 2020; Viegas *et al.*, 2020a].

Supervised strategies usually outperform unsupervised ones [Shaik *et al.*, 2023], this we focus on the former.

However, we exploit information from unsupervised lexicons, such as polarity and intensity, to help build our novel solutions.

Another challenge commonly faced by (supervised or unsupervised) sentiment analysis solutions is related to *information shortage*, especially in short texts (sentences), due to a lack of sufficient information to capture the overall document sentiment [Hu *et al.*, 2009]. To deal with this problem, document enrichment strategies, such as n-grams (a.k.a. Bag-of-n-grams), have been adopted by Huang *et al.* [2018]. These simple models, based on positional information, cannot, however, capture complex semantic relationships among terms, which have a large potential to determine class assignments. Recent strategies adopt techniques to enrich the data representation and deal with information shortage by capturing more complex semantic relationships based on word co-occurrence and contextual information. Examples include c-features [Figueiredo *et al.*, 2011], the use of word embeddings [Viegas *et al.*, 2019] and deep learning (Transformer) models based on attention mechanisms (e.g., BERT [Devlin *et al.*, 2019]).

An undesired side effect of such expansion/enrichment strategies is the possibility of introducing *noise* into the

data. Semantic noise may happen when: (i) the application domain is distinct from the domain in which the embeddings were created (e.g., when using pre-trained embeddings) or (ii) a small training set is used to train the embedding vector space. The absence of (enough) training information makes the vector space inaccurate in capturing semantic information among words. In both scenarios, the learned embedding models may not capture the correct information about a word, especially for infrequent words [Nooralahzadeh *et al.*, 2018]. These potential problems are exacerbated in the context of sentiment analysis due to the already-mentioned issues of information shortage. Given the small number of terms in a message, especially those carrying polarity information (necessary for sentiment inference), a single erroneous expansion or enrichment may completely change the polarity of a phrase or a whole message.

In this context, our main contribution to this article is the proposal of a new solution for sentiment analysis – *CluSent* – that uses the concept of *CluWords* [Viegas *et al.*, 2019] that exploits semantic word clustering representations to tackle the aforementioned issues of information shortage and noise. The main idea is to exploit similarity relationships between words using pre-trained embeddings. We do so by expanding terms with their closely related neighboring words, thus improving both the (co-)occurrence and discriminative power of words in short texts.

In more detail, *CluSent*'s representation exploits the nearest words of a given pre-trained word embedding to generate “meta-words” to expand and enhance the document representation regarding syntactic and semantic information. *CluSent*'s main hypothesis is that by exploiting word embeddings similarities, **and mainly**, by filtering out potential noise (i.e., irrelevant word expansions for the sake of sentiment inference) and by properly weighting them (in the case of sentiment analysis, with the appropriate polarity and intensities), we should be able to construct richer document representations for sentiment inference. In other words, by exploiting customized dataset-oriented filtering and weighting mechanisms, *CluSent* can deal with semantic noise from pre-trained embeddings, especially for short texts.

We rely on sentiment lexicons to build and adapt *CluSent*'s filtering and weighting mechanisms to the sentiment analysis problem. To do so, we propose a new TF-IDF-like (Term Frequency-Inverse Document Frequency) representation that exploits polarity and intensity, what we call *TF-AL* (Term Frequency-Adaptive Lexicon). We employ the *TF-AL* concept as a **filtering/weighting mechanism** in the *CluSent* representation. The idea is to build a *cluster of words* (a.k.a. *CluWord*) of similar polarity and intensity, keeping only words of the same Part-of-Speech (PoS) tagging into a *CluWord*, e.g., only adjectives or nouns with similar polarity and similar intensity would belong to the same *CluWord* for sentiment analysis. In sum, we exploit information in the sentiment lexicon, i.e., polarity and the lexicons' intensity, to filter out words from a *CluWord* cluster. All these innovations are encapsulated into the **CluSent**'s pipeline, which is dynamically instantiated to build dataset-oriented document representations.

In our experimental evaluation, we compare *CluSent* with a considerable spectrum of sentiment analysis methodolo-

gies encompassing; (i) conventional and widely utilized approaches, such as Vader and TextBlob; alongside (ii) newly introduced techniques, such as BERT and kNN Regression Expansion; as well as (iii) methodologies identified as top-performers in recent benchmarks on sentiment analysis. Our experimental evaluations were performed in a large benchmark with 19 datasets, our solution achieved the best results in most scenarios – 28 out of 38 possibilities, considering 19 datasets and two evaluation metrics (i.e. MacroF1 and Accuracy), with gains up to 14.21% (*ss\_bbc*), 7.60% (*ss\_digg*) and 7.17% (*ss\_rw*) compared to the *best baseline in each dataset*, in terms of MacroF1. To promote reproducibility, all the code, the documentation of how to run it, and datasets are available on GitLab<sup>1</sup>.

To summarize, the main contributions of this article include:

- The *proposal* of the *CluSent* method to build rich document representations for sentiment analysis that use information from multiple word embeddings;
- The *exploration* of the *CluWords* concept that exploits semantic word clustering representations combined with sentiment lexicon's polarity and intensity filters to tackle information shortage and noise issues;
- The *demonstration* of how to build and dynamically instantiate the *CluSent*'s filtering (aiming at de-noising) and weighting mechanisms by exploring polarity and intensity information from unsupervised lexicons.
- A thorough *evaluation* of our solution, considering 19 datasets and five strong baselines, including modern transformer-based architectures.
- Code, documentation, and datasets for all our solutions available in a public repository.

This paper is an extension of the work published in WebMedia2023 Viegas *et al.* [2023] and is organized as follows: Section 2 covers related work. Section 3 explains and details the *CluSent* method. Section 4 presents our experimental setup. Section 5 discusses our experimental results. Section 6 concludes the paper.

## 2 Related Work

We review the *CluWords* concept and the state-of-the-art (SOTA) strategies in sentiment analysis directly comparable to *CluSent*.

*CluWords* are clusters of semantically related word embeddings [Mikolov *et al.*, 2018] built by employing distance functions<sup>2</sup>. *CluWords* have been successfully applied in the realm of topic modeling [Júnior *et al.*, 2022] and hierarchical topic modeling scenarios [Viegas *et al.*, 2020b, 2019]. One of our main contributions to this article is demonstrating how to adapt and extend the *CluWords* concept for specific applications through dataset-oriented and task-oriented filtering and weighting mechanisms. We illustrate such adaptation for the realm of sentiment analysis.

BERT [Devlin *et al.*, 2019]<sup>3</sup> is an end-to-end deep

<sup>1</sup>[https://gitlab.com/feliperviegas/cluwords\\_arc](https://gitlab.com/feliperviegas/cluwords_arc)

<sup>2</sup>*CluWords* are not limited by any particular type of word embedding or distance function, being flexible enough to accommodate many options.

<sup>3</sup>Available in <https://github.com/yasarkerl/>

learning language model composed of a bidirectional Transformer encoder. The model is pre-trained with a 3.3 billion word corpus. BERT predicts missing words from a sentence using a multi-layer bidirectional Transformer encoder whose self-attention layer acts forward and backwards. SentiBERT [Yin *et al.*, 2020] is a variant of BERT that captures compositional sentiment semantics. During training, SentiBERT exploits BERT to capture contextual information by masked language modeling. The model learns the meaning composition by predicting the sentiment labels of the phrase nodes. In our experiments, due to documentation limitations and the unavailability of code description, we were unable to evaluate the SentiBERT as provided by its authors<sup>4</sup>. Thus, we include **BERT** as a baseline. Based on the experiments available in Yin *et al.* [Yin *et al.*, 2020], SentiBERT presents gains of 4% on average compared to BERT. As we shall see in our experimental evaluation (Section 4), our proposed method achieved much higher gains over BERT when compared to SentiBERT.

Thongtan and Phienthrakul proposed NB-weighted-BON+dv-cosine [Thongtan and Phienthrakul, 2019] (NB-W-B+dv-cos)<sup>5</sup>, a method that trains document embeddings using cosine similarity. The Cosine similarity helped to reduce overfitting in the embedding generation task. The generated embeddings are combined with Naive Bayes weighted bag-of-n-grams. In their experiments, NB-weighted-BON showed improved results compared to strong baselines, including BERT. In some comparative analyses, **NB-weighted-BON+dv-cosine** is the current state-of-the-art (best-known algorithm) in several sentiment analysis benchmarks, such as in sentiment analysis reviews<sup>6</sup>. We include it as a baseline in our experiments.

Socher *et al.* proposed the Recursive Neural Tensor Network [Socher *et al.*, 2013] (RNTN). RNTN uses a tree where each node contains a word, its sentiment, and its associated label (positive, negative, neutral, very positive, and very negative). This solution represents a sentence using word vectors and an analysis tree. Given a new test document, the tree of this document is generated and compared (by similarity) with existing trees in the training set for predicting the respective label of the test document. RNTN is a classical and popular neural method that explores several paradigms as trees and similarities for sentiment analysis. It is still used as a “de facto” baseline to surpass [Alissa *et al.*, 2021; Jin *et al.*, 2021], given its good average results in general. We also exploit **RNTN** as a baseline.

VADER [Hutto and Gilbert, 2014] and TextBlob [Qi and Shabrina, 2023] were recently credited as the two most prominent and widely utilized lexicon-based methods for sentiment analysis [Qi and Shabrina, 2023]. VADER is an unsupervised method that punctuates sentences according to sentiment intensity (valence-based) lexicons and general rules incorporating grammatical and syntactic conventions (for the English language) to express and emphasize the intensity of sentiments. The robustness of Vader can be attributed to the significant effort to adopt high-quality

intensity scores for rules and lexicons from the agreement of human experts. Because of its popularity and straightforward unsupervised application, recent adaptations of Vader for other languages have been proposed, such as the Portuguese “LeiA” [Jonker *et al.*, 2022], the German “GerVADER” [Tyman *et al.*, 2019], and the Bengali Vader [Amin *et al.*, 2019]. Similarly, TextBlob [Qi and Shabrina, 2023] is a popular lexicon-based implementation that exploits a combination of lexicons from multiple sources, as well as grammatical clues (e.g., negation, intensifiers, and parts of speech) to infer the general sentiment of documents. TextBlob has been recently applied to multiple domains, including politics, airline opinions, and COVID-19 tweets [Oyebode and Orji, 2019; Aljedaani *et al.*, 2022; Abiola *et al.*, 2023]. Thus, we include both **VADER** and **TextBlob** as baselines.

Sachan *et al.* proposed the *L-MIXED* [Sachan *et al.*, 2019]<sup>7</sup> strategies that exploit a BiLSTM model with pre-trained embeddings. Their training strategy achieves higher accuracy than more complex models without an extra pretraining step. To do that, the authors explored the applicability of semi-supervised learning (SSL), where no previous pretraining step exists. The authors also proposed a mixed objective function for SSL that utilizes labeled and unlabeled data. **L-MIXED** is the current SOTA solution (best-known method) in several datasets used in our experiments. We also included it as one of our baselines.

**kNN Regression Expansion** (kNN Reg. Exp.) [Viegas *et al.*, 2020a] is a lexicon-based method that exploits semantic information from word embedding models to expand lexicon dictionaries. The method exploits a lexicon dictionary (VADER lexicons) and word embeddings to map the sentiment value of new lexicons (new words that will be added to the lexicon dictionary). The method uses the nearest neighbors approach to infer the sentiment value of the new lexicons (words with polarity and intensity). To predict the polarity at the sentence-level, the method exploited the VADER’s shell [Hutto and Gilbert, 2014]. VADER shell is a method that implements four general rules incorporating grammatical and syntactic conventions (for the English language) to express and emphasize the intensity of sentiments. The shell exploits these rules and the lexicon to compute a sentiment value for a sentence. Besides being highly effective [Viegas *et al.*, 2020a], this method, similarly to CluSent, exploits word embeddings and distance-based neighborhoods. Therefore, our experiments include **kNN Regression Expansion** as a close baseline.

In another vein, a recent trend in unsupervised learning involves the application of zero-shot classification using language models, initially proposed for the sentiment analysis task on SST-2, Amazon, and Yelp datasets, with prompting engineering for the generative GPT-2 language model [Puri and Catanzaro, 2019]. An alternative approach involves employing zero-shot learning with BERT pre-trained models for entailment tasks [Yin *et al.*, 2019], where the goal is to ascertain whether a premise sentence logically entails a hypothesis sentence corresponding to a particular label, such as positive or negative sentiments. However, this methodology has been deemed ineffective for various un-

<sup>4</sup><https://github.com/DeepakDhana/SentiALBERT1>

<sup>5</sup><https://github.com/tanhtongtan/dv-cosine>

<sup>6</sup><https://paperswithcode.com/sota/sentiment-analysis-on-imdb>

<sup>7</sup>[github.com/DevSinghSachan/ssl\\_text\\_classification](https://github.com/DevSinghSachan/ssl_text_classification)

supervised classification tasks, including sentiment analysis on datasets like SST-2 [Ma *et al.*, 2021]. In order to examine the emerging trend of zero-shot classification, we delved into the capabilities of the novel open multilingual language model, BLOOM-560M, incorporating prompting engineering techniques [Yong *et al.*, 2023]. Our aim is to assess its efficacy in the realm of unsupervised sentiment analysis.

Finally, we included experiments using the traditional GloVe word embeddings, which explicitly exploit global co-occurrences of words in documents to build simplified yet generalizable semantic representations of words [Pennington *et al.*, 2014]. GloVe can be exploited by supervised or unsupervised strategies. In our experiments, we exploit such representation with the supervised logistic regression to provide a superior limit for its effectiveness.

### 3 The CluSent Method

Conceptually, CluSent comprises a pipeline with three generic steps applied to a given text (embedding) representation: clustering, filtering, and weighting that, together, build a richer (more informative) representation for a textual collection. Figure 1 illustrates how CluSent representations are instantiated for a given collection. Each dot in the Figure represents an instantiation of a method applied to compose the CluSent representation. In a nutshell, CluSent builds clusters of semantically related word embeddings [Mikolov *et al.*, 2018] through the application of distance functions (first blue dot in Figure 1) and filtering mechanisms (second and third-half dot in Figure 1). More than simple groups of (filtered) related words, CluSent’s clusters apply specific weighting schemes<sup>8</sup> to model their importance to sentiment analysis tasks (purple dots in Figure 1). In Section 3.1, we present the clustering solution. Next, we describe (Section 3.2) CluSent’s part-of-speech filtering method, followed by (Section 3.3) the filtering and weighting steps that exploit sentiment information and are used to build the document representation (Section 3.4).

#### 3.1 Clustering

Clustering is a crucial step in the CluSent’s instantiation. It employs strategies to capture semantic relationships between words captured by their embedding representations. As we can observe in Figure 1, this step requires a word embedding vector space as input. Word (vectors) can be represented using static or contextual word embeddings. Static word embedding generates a single embedding representation for every single word in a given corpus, while contextual word embeddings, induced by attention mechanisms within transformer architectures [Devlin *et al.*, 2019], produce potentially several representations for the same word  $w$  depending on the context (surrounding words) in which  $w$  appears.

CluWord’s clustering step can only receive as input static word embeddings. In this context, a preprocessing step is required to transform contextual word embeddings into static ones, employing pooling techniques for all contextual

embeddings referring to the same word  $w$ . In this work, we exploit the Average Pooling, previously proposed to improve the representative power of words embeddings for computing word similarities [Bommasani *et al.*, 2020], and to provide robust word representations for the word sense disambiguation problem [Loureiro and Camacho-Collados, 2020]. Such previous works motivated our proposed experiments to evaluate the representative power of averaged contextual embeddings for the sentiment analysis task with CluSent considering the wide range of scenarios in our benchmark. Formally, the adopted average pooling is described in Equation 1, where  $\vec{w}_i$  corresponds to a contextual embedding for word  $w$  in the contextual word space.

$$\overline{\mu w}_i = \frac{\sum_1^N \vec{w}_i}{|N|} \quad (1)$$

Let  $\mathcal{W}$  be the set of static vectors representing each word  $w$  in the dataset vocabulary (represented as  $\mathcal{V}$ ). Each word  $w \in \mathcal{V}$  has a corresponding vector  $\vec{w} \in \mathcal{W}$ . The semantic matrix in Figure 1 is defined as  $C \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ , where each dimension has the size of the vocabulary ( $|\mathcal{V}|$ ),  $w$  represents the rows of  $C$  while the dimensions of  $\vec{w}$  correspond to the columns. Finally, each index  $C_{w_i, w_j}$  is computed according to Equation 2.

$$C_{w_i, w_j} = \begin{cases} \omega(\vec{w}_i, \vec{w}_j) & , \text{ if } \omega(\vec{w}_i, \vec{w}_j) \geq \alpha \\ 0 & , \text{ otherwise} \end{cases} \quad (2)$$

where  $\omega(\vec{w}_i, \vec{w}_j)$  is the cosine similarity defined in Equation 3 and  $\alpha$  is a similarity threshold that acts as a regularizer for the representation. Larger values of  $\alpha$  lead to sparser representations. In this notation, each column  $w_i$  of the semantic matrix  $C$  will form a CluWord  $w_i$ , and each value of the matrix  $C_{w_i, w_j}$  will receive the cosine similarity between the vectors  $w_i$  and  $w_j$  in the embedding space  $\mathcal{W}$ , if it is greater than or equal to  $\alpha$ . Otherwise,  $C_{w_i, w_j}$  receives zero, according to the Equation 2.

$$\omega(\vec{w}_i, \vec{w}_j) = \frac{\sum_1^{|\mathcal{V}|} \vec{w}_i \cdot \vec{w}_j}{\sqrt{\sum_1^{|\mathcal{V}|} \vec{w}_i^2} \cdot \sqrt{\sum_1^{|\mathcal{V}|} \vec{w}_j^2}} \quad (3)$$

The vector  $\vec{C}_w$  represents the semantic information of a *cluster of words* (a.k.a., CluWord) for word  $w$ , and the  $\alpha$  values filter potential noisy words (i.e., words that do not have a significant relationship with  $w$ ). Since threshold  $\alpha$  is a cosine similarity value, it is contained within the interval  $[0, 1]$ . If  $\alpha = 0$ , the similarities of every term in  $\mathcal{V}$  are included in the CluWord of  $w$ . If  $\alpha = 1$  only the similarity of  $w$  to itself (i.e.,  $\omega(w, w)$ ) is included in CluWord  $w$ . Thus, the appropriate selection of a value for parameter  $\alpha$  is an important aspect of generating “good” CluWord for  $w$ . Moreover,  $\alpha$  controls the sparsity of the resulting document representation. With high  $\alpha$  values, only a few CluWord terms relate to a document. This representation is similar to the traditional BoW representation, where the occurrence of a word in a document determines whether that word will be used in the document representation. With low  $\alpha$  values, more CluWord terms tend

<sup>8</sup>These weighting schemes combine the raw document representation with relevant information, such as semantic and/or lexicon information.

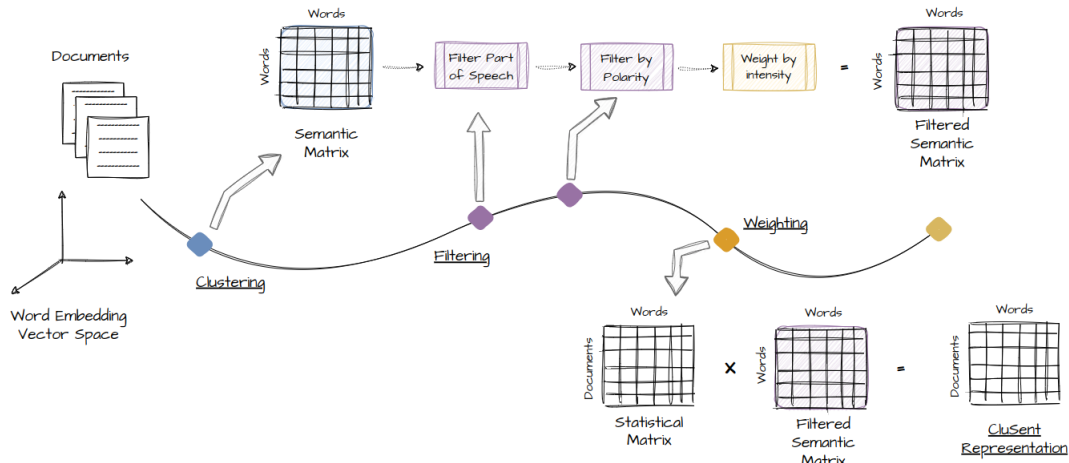


Figure 1. Diagram showing the steps for building the CluSent representation.

to be related to the document, reducing the document representation’s sparsity. Once we select an appropriate value for  $\alpha$ , each CluWord  $w$  keeps the values of similarities of the terms most similar to  $w$  according to the criteria (e.g., context, co-occurrence) established by the word embeddings.

The nearest neighbor search is critical for building the CluSent document representation. In the Clustering step, we employ Hierarchical Navigable Small World (HNSW) [Malkov and Yashunin, 2020], a data structure and algorithm designed for efficient approximate nearest neighbor search in high-dimensional spaces. HNSW builds a hierarchical graph where each node represents a data point (in our scenario, a word embedding). The connections (edges in the graph) represent their proximity in the high-dimensional vector space. The hierarchical structure speeds up the search, efficiently exploiting the search space. HNSW produces high efficiency in scenarios requiring approximate nearest-neighbor search retrieval [Foster and Kimia, 2023; Cunha et al., 2023b,a].

Figure 2 illustrates the sub-steps used to construct the approximate nearest neighbor search inside the Clustering Step (illustrated in Figure 1). First, a dataset collection and a word vector space are required as input. The dataset is used to extract all the tokens (a.k.a. words) that will be used to filter out the word vectors from the word vector space. This step keeps track of the vocabulary that CluSent will use. It also removes unnecessary tokens in the CluSent generation. Since the generation of the similarity matrix  $C$  is an exhaustive nearest neighbor search, we exploit in this step the previously described HNSW graph structure.<sup>9</sup>

To summarize, any embedding model (static or contextual) can be exploited to build the semantic matrix  $C$ . It is also important to note that it can also be applied to other languages. Nowadays, plenty of pre-trained embedding models exist for languages other than English and even multi-languages.

### 3.2 Part-of-Speech Filtering

Here, we describe the part-of-speech filtering mechanism used to smooth noise in the semantic matrix  $C \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ . This filter removes pairs of words that do not belong to the same grammatical group. Thus, this filter keeps in a neighborhood of a CluWord  $w$  ( $\vec{C}_{w}$ ) only terms ( $w_j$ ) that have a semantic similarity and share the same grammatical group. The intuition is that, for the sake of sentiment analysis, we want to keep adjectives that are semantically similar to other adjectives, verbs that are semantically similar to other verbs, same for adverbs, and so on. We will analyze the impact of this very conservative filter in our experiments.

Formally, the Part-of-Speech (PoS) filtering method uses a function  $pos(\cdot)$  to filter each term  $w_j$  of  $\vec{C}_{w}$  that does not belong to the same part-of-speech category of term  $w$  (Equation 4). We exploit the PoS tagging from Spacy<sup>10</sup> to build function  $pos(\cdot)$ . Spacy PoS tagging is available for over 24 languages and has a PoS tagging for multi-language texts. In addition, any PoS tagging can be applied to build the function  $pos(\cdot)$ .

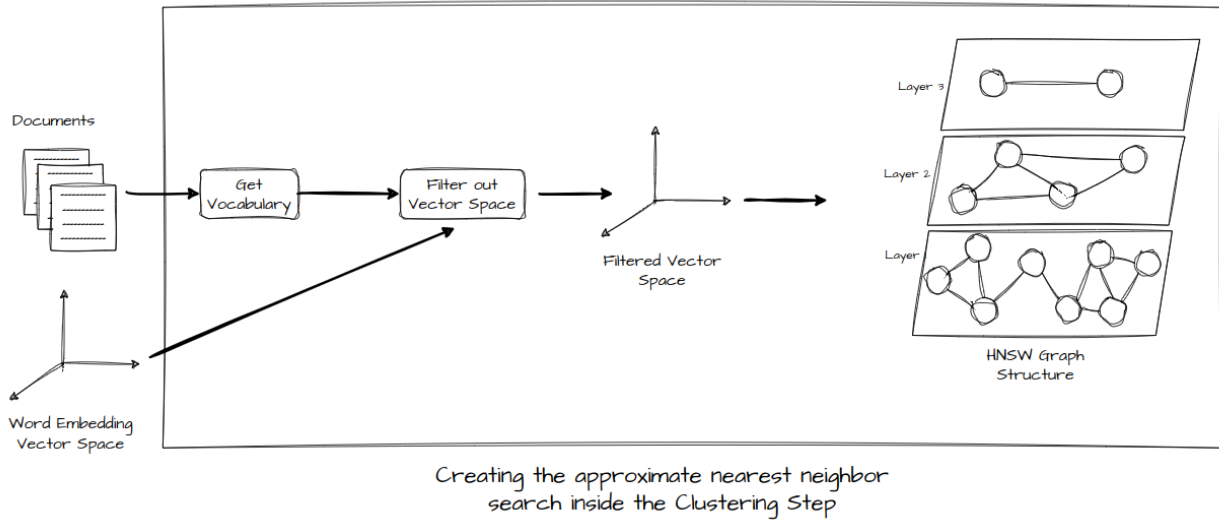
$$C_{w_i, w_j} = \begin{cases} C_{w_i, w_j} & \text{if } pos(w_i) = pos(w_j) \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

### 3.3 Sentiment Filtering and Weighting

Several sentiment analysis approaches employ lexicon dictionaries. A lexicon is formed by a set of words tagged with their respective *sentiment value*, consisting of a number (within a defined range) that expresses both the words’ polarity (given by the number’s sign) and intensity (given by number’s absolute value). The intuition of this filter block in the CluSent is to use information from a lexicon dictionary to filter out semantic noise that can affect the quality of the representation, especially in the sentiment analysis scenario.

<sup>9</sup><https://docs.vespa.ai/en/approximate-nn-hnsw.html>

<sup>10</sup><https://spacy.io/usage/linguistic-features#pos-tagging>



**Figure 2.** Creation of the HNSW graph structure inside the Clustering step. This process speeds up the construction of the similarity matrix since all the word vectors are mapped in the graph structure.

Without this filter, words with opposite polarities may be co-located in the same neighborhood of a CluWord  $w$  since the semantic similarity of embeddings correlated with positional, contextual, and co-occurrence information does not consider a word’s polarity. Thus, words of opposite polarities may belong to the same CluWord. Indeed, this phenomenon has been observed in the literature [Viegas et al., 2020a].

Indeed, for a given a set of words represented by their word embeddings and tagged with their respective sentiment values (i.e., a lexicon dictionary), it is very much possible for a single cluster of words, constructed exclusively based on the similarity between word embeddings, to encompass both positively and negatively tagged words. Consequently, if a single positive word appears within a cluster primarily composed of negative words, documents containing this positive word may exhibit a bias towards the negative words present in the cluster. Our filtering approach is designed precisely to avoid such bias.

Furthermore, without filtering, there are more clusters comprising multiple words that appear across multiple documents. Consequently, the information provided by these high-coverage clusters tends also to be more biased towards the majority class on imbalanced datasets.

We adopt this filter, which we call TF-AL (Term Frequency-Adaptive Lexicon) to keep *polarity consistency* within a CluWord. We also exploit the lexicon’s word intensity as a weighting scheme to enhance the semantic information within a CluWord. More formally, the lexicon dictionary is represented as  $\mathcal{L} = \{\langle w_1, v_1 \rangle, \dots, \langle w_{|\mathcal{L}|}, v_{|\mathcal{L}|} \rangle\}$ , where  $w_i$  is a word and  $v_i$  is the sentiment value of the word  $w_i$ ,  $1 \leq i \leq |\mathcal{L}|$ . The sentiment value  $v_i$  of a word  $w_i$  expresses the word’s polarity and intensity. The sentiment absolute values may vary according to the lexicon used. In *CluSent*, we use an expanded version of the VADER [Hutto and Gilbert, 2014] lexical dictionary proposed in [Viegas et al., 2020a], where the sentiment absolute values range between  $(-4, 4)$ . We adopt a VADER-based lexicon because its lexicons vary from  $(-4, 4)$ , and this range of values can

be more sensitive in terms of weighting the words, but any lexicon dictionary (even for languages other than English) can be applied in the building block. For instance, there is a version of the VADER lexicon for Portuguese<sup>11</sup>.

Given the semantic matrix,  $C$ , the method exploits Equation 5 to filter terms  $w_j$  of  $C_{w_i}$  that do not share the same polarity as term  $w_i$ . In addition, the sentiment value of the term  $w_i$  is used to weight the semantic value  $C_{w_i, w_j}$ .

$$C_{w_i, w_j} = \begin{cases} C_{w_i, w_j} \times v_{w_i} & , \text{ if } \text{sign}(v_i) = \text{sign}(v_j) \\ 0 & , \text{ otherwise} \end{cases} \quad (5)$$

### 3.4 Building the CluSent Representation

This step is responsible for building the CluSent representation (the last purple dot in Figure 1), which is defined as the product between the term-frequency matrix and semantic matrix  $C$ . The term-frequency matrix ( $TF$ ) can be represented as a  $TF \in \mathbb{R}^{|\mathcal{D}| \times |\mathcal{V}|}$ , where each position  $TF_{d,t}$  relates to the frequency of a word  $w$  in document  $d$ . Thus, given a CluSent (CS) term  $w$  for a document  $d$ , its data representation corresponds to:

$$CS_{d,w} = \overrightarrow{TF}_d \times \overrightarrow{C}_{w} \quad (6)$$

where  $\overrightarrow{TF}_d$  has the term-frequencies of document  $d$ , and  $\overrightarrow{C}_{w}$  is the semantic scores for the term  $w$ .

The entire process of generating the CluSent representation and its instantiation for a given task are illustrated in Figure 3. CluSent is an unsupervised method that builds a matrix-based model. The Clustering and Filtering steps build the Semantic Matrix  $C$ . Then, the Weighting step receives the matrix-based model and the input documents, either in-batch mode, allowing the whole collection process or, on the fly, to build the CluSent representation of a new single input document.

<sup>11</sup><https://github.com/rafjaa/LeIA/blob/master/lexicons>

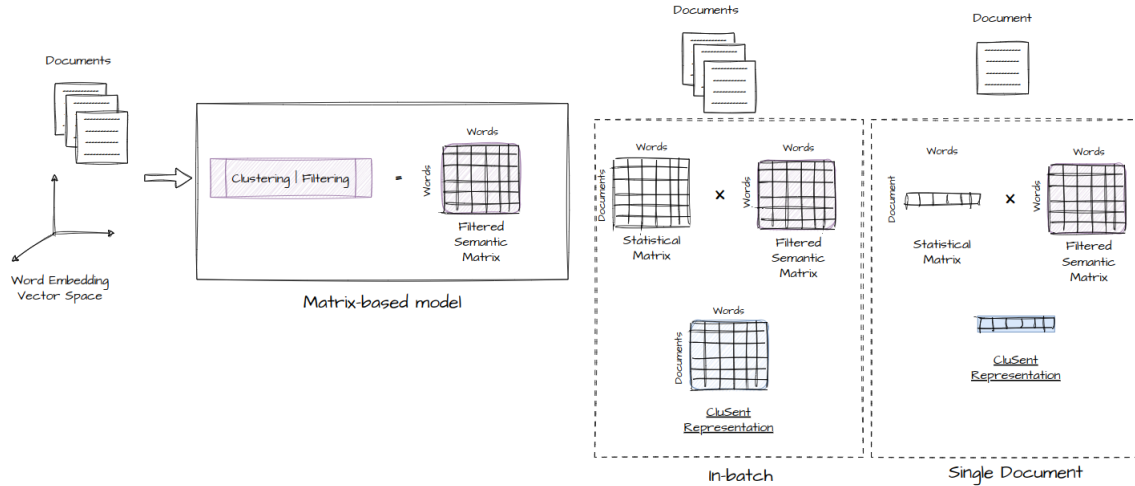


Figure 3. Diagram illustrating CluSent’s flow for in-batch and on-the-fly data representation creation.

### 3.5 Complexity of CluSent

The complexity of building the clustering step (Section 3.1) is the nearest neighbor search, which can be exploited by using the fast approximate nearest neighbor search (HNSW) [Malkov and Yashunin, 2020] with the complexity of  $\mathcal{O}(\log N)$ . The CluSent’s steps described in Sections 3.2 and 3.3 are search terms in a sparse matrix ( $\mathbb{R}^{|\mathcal{D}| \times |\mathcal{V}|}$ ) representation, and the complexity of those searches is  $\mathcal{O}(NNZ)$ , where NNZ represents the non-zero values. Finally, the complexity of Section 3.4 is the matrix multiplication ( $\overrightarrow{TF_d} \times \overrightarrow{C_{,w}}$ ) in Equation 6). Since both matrices are sparse, the complexity is defined in Equation 7, where  $|\mathcal{V}|$  is the vocabulary size.

$$\mathcal{O}\left(\frac{NNZ(\overrightarrow{TF_d})NNZ(\overrightarrow{C_{,w}})}{|\mathcal{V}|}\right) \quad (7)$$

### 3.6 CluSent Working Example

Let us consider the following sentence: “*The team did an excellent job on the project; the results were outstanding.*”. For didactic reasons, let us assume that the vocabulary is the following words  $\mathbb{V} = \{bad, do, does, excellent, great, job, on, outstanding, results, team, the, worst\}$ . All the words in  $\mathbb{V}$  have a respective embedding representation in  $\mathbb{W}$ . During the clustering step, and assuming that we selected a conservative cosine threshold to limit the nearest neighbor search, the row of the following word *excellent* in the Semantic Matrix  $C$  ( $\overrightarrow{C_{excellent}}$ ) will contain as its neighbors the words  $\{great, worst, do, does\}$ . This happens because the word embedding also considers the contextual position in the sentence to determine the semantic relationship between words.

Since the Semantic Matrix  $C$  is symmetric, the same will happen for the words *great, worst, do, does*. The importance of the PoS filtering is to filter out words with different PoS, so, in the same example, after the PoS filtering, the  $\overrightarrow{C_{excellent}}$  row will contain only adjectives –  $\{great, worst\}$ , as well as, the rows for the words *do* ( $\overrightarrow{C_{do}}$ ) and *does* ( $\overrightarrow{C_{does}}$ ) will only contain verbs.

The role of the Sentiment Filtering step is to eliminate

potential noise related to sentiment polarity. In the same example, the  $\overrightarrow{C_{excellent}}$  row will contain only adjectives that share the same (positive) polarity of the word *excellent*, in the case, the word  $\{great\}$ , after the application of this step.

Finally, the Weighting step will increase the importance of each word in the sentence and the words in their respective neighborhood. So, if we disregard the BoW representation and the weights, the final representation of the sentence “*The team did an excellent job on the project; the results were outstanding*” will be the weights of the following words “the team did an excellent job on the project the results were outstanding **does great**”. The word *great* is added in the final representation because of its semantic similarity with the word *excellent*, and the *does* because of its relationship with the word *do*.

## 4 Experiments

### 4.1 Textual datasets

To evaluate the quality of the proposed methods, we adopt nineteen real-world textual datasets gathered from various sources, such as the highly popular SEMEVAL (semeval\_tw) [Rosenthal et al., 2019], stanford\_tw [Go et al., 2009] and Stanford Sentiment Treebank v2 (SST-2)<sup>12</sup> datasets. Besides those, we exploit 16 other datasets with various news, reviews, and social media domains with different characteristics, such as class distribution, density, etc. These datasets have high relevance for sentiment analysis, used, for instance, in popular benchmarks [Ribeiro et al., 2016], as well as in highly cited papers such as the VADER lexicon one [Hutto and Gilbert, 2014].

Table 1 shows some characteristics of these 19 datasets. Each column depicts, respectively, the dataset’s name, number of *messages*, number of words, the average number of words (density) in each message, and the number of positive and negative messages. As we can see, most of the datasets are highly imbalanced, i.e., they have a skewed distribution, increasing the bias towards the largest class.

<sup>12</sup><https://www.kaggle.com/atulnandjha/stanford-sentiment-treebank-v2-sst2>

## 4.2 Evaluation, Algorithms, and Procedures

The effectiveness of the experiments was evaluated using two standard text categorization measures: *MicroF1* and *MacroF1* [Lewis et al., 2004]. While *MicroF1* measures the classification effectiveness of overall decisions, *MacroF1* measures the classification effectiveness for each class and averages them. *MacroF1* is very suitable for datasets with high imbalance as all classes have the same importance in the measure. It is worth considering both evaluation metrics since most of the datasets used in the experimental evaluation present class imbalance (i.e. debate, semeval\_tw, etc.). This information can be seen in Table 1, columns #pos and #neg.

All experiments were executed using a 5-fold cross-validation procedure. All tuning parameters for the baselines and our methods were discovered in the validation partitions while the reported results correspond to the average on the 5 test sets of the folded cross-validation procedure.

We assess the statistical significance of our results by exploiting a Two-way ANOVA test with 95% confidence. This test assures that the best results are statistically superior to all others. In Table 4, the best results were marked with a green triangle ( $\blacktriangle$ ), statistical ties are represented as a yellow dot ( $\bullet$ ), while losses are represented as red downward triangles ( $\blacktriangledown$ ).

**Table 1.** Dataset characteristics

Dataset	#msgs	#feat	density	#pos	#neg
aisopos_tw	278	1,586	83.60	159	119
debate	1,979	4,179	86.49	730	1,249
narr_tw	1,227	4,002	74.76	739	488
pappas_ted	727	1,886	92.16	318	409
sanders_tw	1,091	3,601	97.08	519	572
ss_bbc	752	7,674	396.82	99	653
ss_digg	782	5,164	188.49	210	572
ss_myspace	834	2,914	104.26	702	132
ss_rw	705	5,643	345.02	484	221
ss_twitter	2,289	8,835	94.19	1,340	949
ss_youtube	2,432	7,534	90.04	1,665	767
stanford_tw	359	1,746	81.62	182	177
semeval_tw	3,060	10,507	115.99	2,223	837
vader_amzn	3,610	5,039	88.54	2,128	1,482
vader_movie	10,568	17,759	111.67	5,242	5,326
vader_nyt	4,946	12,932	105.42	2,204	2,742
vader_tw	4,196	9,046	79.69	2,897	1,299
yelp_review	5,000	25,494	681.46	2,500	2,500
SST-2	68,221	14,583	53.17	38,013	30,208

We use as baselines popular and effective methods used in other public benchmarks [Mabrouk et al., 2020] such as RNTN, NB-weighted-BON+dv-cosine, kNN Regression Expansion, and L-MIXED. In one of these benchmarks [Mabrouk et al., 2020], L-MIXED produced the best-known results in the literature in some of the tested datasets. We also consider BERT as a solid baseline since it was surpassed only marginally (without statistical significance) by another recent SOTA baseline (SentiBERT), which could not be used in our experiments due to a lack of code and reproducibility information in the original paper. Finally, we also adopted the kNN Regression Expansion, a recent and effective sentiment analysis SOTA baseline

especially designed for short-text datasets, as is the case with most experimented datasets [Viegas et al., 2020a].

For BERT, we configured hyperparameters as suggested by [Cunha et al., 2023c; de Andrade et al., 2023]. We performed a search for the best hyperparameters following a trial-and-error process, and the best set for the remaining ones was chosen with fine-tuning using nested cross-validation within the training sets (batch size: 32, initial learning rate: 5e-5, max sequence length: 150 tokens, max patience: 5 epochs). For other baselines, we performed fine-tuning according to the appropriate author’s scripts in the source code. For RNTN, the hyperparameter word vector size, learning, and mini-batch size are adjusted with the AdaGrad algorithm, while the activation function is hyperbolic tangent. For NB-weighted-BON+dv-cosine and L-MIXED, we used grid search to optimize the number of iterations, learning rate, and regularization force. For kNN Regression Expansion, we exploited the pre-trained FastText embedding and performed fine-tuning of neighbors according to the author’s script in the source code.

For CluSent, we consider the pre-trained FastText embedding<sup>13</sup> to build the semantic matrix, described in Section 3. This embedding model was trained using data from Wikipedia 2017, UMBC webbase corpus, and statmt.org news dataset. FastText is essentially an extension of the Word2Vec model, which treats each word as composed of character n-grams, allowing to (i) generate better word embeddings for rare words and (ii) construct word vectors for a word that does not appear in the training corpus. Both improvements are not implemented in GloVE [Pennington et al., 2014].

The  $\alpha$  parameter (in Equation 2 Section 3.1) is strictly sensitive to the embedding space, being responsible for controlling the CluSent’s density. The smaller the alpha value, the greater the CluSent representation’s density. A small alpha may increase the noise in the CluSent representation, while a large alpha may impoverish it. We adopted a percentile-based strategy to select the 5% of word pairs with the highest cosine similarity scores in the embedding space. This process was performed empirically over the FastText embeddings.

We run nested cross-validation over the training set to select the best CluSent instantiation for each dataset. In other words, our aim is to automatically determine the optimal instantiation of our approach for each dataset. Consequently, the decision regarding whether to employ mechanisms such as PosTagging filtering or TF-AL weighting is automatically made based on the most effective variation observed in the averaged (cross-validation) training/test splits, exclusively sampled from the training dataset. We exploit the Linear SVM classifier in the CluSent, a top-notch method for text classification that is even superior to neural architectures such as BERT when faced with information shortage [Cunha et al., 2021]. The regularization parameter was chosen among eleven values from  $2^{-5}$  to  $2^{15}$ , also by using 5-fold nested cross-validation within the training set.

<sup>13</sup><https://dl.fbaipublicfiles.com/fasttext/vectors-english/wiki-news-300d-1M.vec.zip>



## 5 Experimental Results

In this Section, we discuss the effectiveness results of our method. We start by evaluating the impact of the word embedding representation in the method (Section 5.1). Next, we present an ablation experiment showing different CluSent instantiations and how the inclusion (or not) of its different steps impact the final solution (Section 5.2). Then we directly compare the best CluSent version with the baselines. Finally, an analysis of difficult cases as well as of the density of the CluSent representation are presented in Sections 5.4 and 5.5.

### 5.1 Impact of the Word Embedding Representation

We start our analysis by comparing the impact of the two word embedding vector spaces in the CluSent instantiations. For the static version, we adopt the pre-trained fastText embedding model described in Section 4.2, trained using data from Wikipedia 2017, UMBC WebBase corpus and statmt.org news dataset. For the contextual version, we used the BERT embedding representation transformed into a static embedding using the average pooling as described in Section 3.1. Regarding BERT, we fine-tuned the model for each dataset, using pre-trained *bert-base-cased* as a backbone model. In the fine-tuning process, we trained using batch sizes of 16, 10 epochs, AdamW optimizer with learning rate equal to  $1e - 5$ , 20 warm-up steps, and NLLLoss.

Table 2 shows the MacroF1 results of both CluSent instantiations – fastText and pooled BERT, respectively. As we can observe, in most cases (14 out of 19 datasets), both instantiations have statistical ties. In five out of 19 datasets, CluSent using fastText presents better results than CluSent using pooled BERT embeddings. We hypothesize that this happens for two reasons: (i) the average pooling mechanism that transforms BERT embeddings into static ones may lose relevant semantic information during the aggregation process *when documents are very short*; (ii) the corpus used to train the pre-trained embeddings may impact the quality of the semantic relationship. FastText embeddings were created based on some news datasets – this may favor this representation during the CluSent instantiation when evaluating news-oriented datasets such as *ss\_bbc*, *ss\_digg*. On the other hand, BERT’s pre-trained model was trained in BookCorpus and English Wikipedia, and since most of the evaluated datasets are small, the lack of information may have impacted the fine-tuning process, not giving enough information to enhance this embedding representation.

Indeed, the negative impact of limited training datasets on the fine-tuning of BERT has been previously reported in the literature [Edwards et al., 2020], which compared BERT to fastText on various training sizes and classification datasets. Empirical findings indicated that when applied to small training datasets, the utilization of fastText in conjunction with domain-specific word embeddings leads to comparable or superior performance to BERT, even when the latter is pre-trained on domain-specific data. This is consistent with our results.

Regarding reason (i), despite evidence in the literature

**Table 2.** MacroF1 results of different word vectors in the CluSent instantiations combined with the linear SVM classifier. We contrast the use of the pre-trained fastText embeddings with pooled BERT embeddings fine-tuned for each dataset.

Dataset	CluSent (fastText)	CluSent (pooled BERT)
aisopos_tw	87.74 ▲	77.5752 ▼
debate	75.13 ●	75.6995 ●
narr_tw	86.50 ▲	82.8189 ▼
pappas_ted	78.82 ●	77.6617 ●
sanders	80.37 ●	79.3968 ●
ss_bbc	68.94 ▲	61.2315 ▼
ss_digg	71.07 ●	70.9715 ●
ss_myspace	73.35 ●	70.3655 ●
ss_rw	75.62 ●	73.3187 ●
ss_twitter	75.44 ●	73.069 ●
ss_youtube	79.02 ●	78.8033 ●
stanford_tw	77.07 ●	78.1776 ●
semeval_tw	76.51 ●	74.2607 ●
vader_amzn	71.94 ●	73.5744 ●
vader_movie	75.11 ●	75.5499 ●
vader_nyt	65.56 ●	65.9122 ●
vader_tw	89.63 ▲	85.8376 ▼
yelp_review	92.36 ●	92.4193 ●
SST-2	89.02 ▲	77.6441 ▼

demonstrating benefits of the average of BERT word embeddings, it achieved significantly inferior results in comparison to fastText on our largest dataset (SST-2). By examining the document’s density for each dataset in Table 1, it is evident that SST-2 documents exhibit the shortest text lengths among all evaluated datasets. As a consequence, the process of fine-tuning using the SST-2 dataset produces highly specific contextual information for the SST-2 word embeddings, since these embeddings are prone to exhibit a strong contextual bias towards the few words within concise sentences.

The strategy of averaging such high-variance embeddings leads to a misleading summary of the contextual information within these word embeddings. This SST-2 pattern also manifests in other datasets, since almost all statistically significant losses (4 of 5) of BERT compared to fastText also occur in datasets containing the shortest texts.

From now on, we will only consider the CluWords version with the fastText static embeddings as they generalize better for most situations.

### 5.2 Ablation Analysis - CluSent Instantiations

We perform an experiment to observe the impact of varying the CluSent instantiation. The intuition of this experiment is to evaluate different forms of instantiating the method and the impact on effectiveness by turning on/off some of the proposed steps in a *per dataset basis*. Since the effectiveness of filtering and weighting may be different on distinct datasets, we also evaluate a CluSent instantiation in which the steps that will be included are automatically chosen with tuning using nested cross-validation in the training set. In other words, in CluSent, the filtering/weighting steps that will be turned on/off are automatically chosen, being potentially different for each dataset.

Each evaluated instantiation is seen in Table 3, where effectiveness the results were performed over a nested cross-validation over the training (a.k.a effectiveness results

**Table 3.** MacroF1 results of different CluSent instantiations combined with the linear SVM classifier. Auto CluSent corresponds to the results of an automatic instantiation of the CluWords, where X stands for the components used in the CluSent instantiation.

Dataset	CW				Auto CluSent		
	CW	CW + PoS	CW + TF-AL	CW + PoS + TF-AL	MacroF1	Instantiations	
						PoS	TF-AL
aisopos_tw	<b>86.95</b>	83.61	<b>87.71</b>	<b>89.38</b>	<b>87.74</b> ●	×	×
debate	<b>74.50</b>	<b>75.23</b>	<b>75.73</b>	<b>75.76</b>	<b>75.13</b> ●	×	×
narr_tw	<b>84.77</b>	82.67	<b>86.51</b>	<b>85.15</b>	<b>86.50</b> ●		×
pappas_ted	<b>78.35</b>	<b>77.75</b>	<b>77.53</b>	<b>77.86</b>	<b>78.82</b> ●		
sanders	<b>81.61</b>	<b>80.80</b>	<b>81.36</b>	<b>80.06</b>	<b>80.37</b> ●		
ss_bbc	<b>68.19</b>	64.12	<b>67.35</b>	<b>66.03</b>	<b>68.94</b> ●	×	×
ss_digg	71.73	<b>66.80</b>	71.86	71.85	<b>71.07</b> ●	×	×
ss_myspace	<b>74.76</b>	70.71	<b>71.86</b>	70.47	<b>73.35</b> ●	×	×
ss_rw	76.78	<b>70.81</b>	76.41	77.03	<b>75.62</b> ●	×	×
ss_twitter	<b>76.73</b>	<b>75.93</b>	<b>77.16</b>	<b>76.49</b>	<b>75.44</b> ●		
ss_youtube	<b>82.14</b>	<b>80.36</b>	<b>79.94</b>	<b>79.23</b>	<b>79.02</b> ●	×	×
stanford_tw	75.93	75.93	<b>79.79</b>	<b>79.89</b>	<b>77.07</b> ●		×
semeval_tw	<b>76.29</b>	<b>75.80</b>	<b>76.99</b>	<b>76.94</b>	<b>76.51</b> ●		
vader_amzn	<b>69.26</b>	<b>68.99</b>	<b>71.64</b>	<b>72.22</b>	<b>71.94</b> ●	×	×
vader_movie	<b>75.03</b>	<b>75.47</b>	<b>73.59</b>	<b>74.65</b>	<b>75.11</b> ●	×	
vader_nyt	<b>66.15</b>	<b>65.69</b>	<b>66.56</b>	<b>66.41</b>	<b>65.56</b> ●		×
vader_tw	86.62	86.41	<b>89.64</b>	<b>90.05</b>	<b>89.63</b> ●	×	×
yelp_review	<b>92.72</b>	<b>92.54</b>	<b>92.10</b>	<b>92.14</b>	<b>92.36</b> ●		
SST-2	<b>88.49</b>	<b>88.49</b>	<b>88.22</b>	<b>88.25</b>	<b>89.02</b> ●		

of the grid search). Each column in the Table represents a different instantiation described as follows: (i) CW – is the core of the CluSent representation, corresponding to the instantiation of the clusterization method (Section 3.1), and the merge between Term Frequency and Semantic information (Section 3.4); (ii) CW + PoS – adds to the previous CluSent instantiation CW, the Part-of-Speech filtering method (Section 3.2); (iii) CW + TF-AL – builds the CluSent representation adding the core methods (Sections 3.1 and 3.4), and the sentiment filtering and weighting technique (Section 3.3); CW + PoS + TF-AL – turns on all components to build the representation; (iv) Auto CluSent – is the automatic instantiation of the Part-of-Speech and Sentiment filtering and weighting approaches that choose the best components to turn on/off for each dataset based on automatic tuning.

Table 3 shows the MacroF1 effectiveness. Besides the explained marks for the statistical tests (▲, ●, ▼) best results in all datasets (including ties) are also marked in **bold** in the Table. The results showed that Auto CluSent ties with the best manual CluSent instantiation in every dataset, the only instantiation to obtain the best effectiveness in all 19 experimented datasets. In the next sections, we will adopt Auto CluSent, hereafter simply called CluSent, as the method of choice to compare against the baselines.

Finally, for analysis purposes, we added in Table 3 the instantiated components (marked with ×) selected in CluSent’s tuning process. We can see that when the PosTagging component is turned on, the TF-AL filtering/weighting is also selected. There are a few cases in which only the TF-AL component is selected, such as *narr\_tw*, *stanford\_tw*, and *vader\_tw*. In these datasets, the PosTagging filtering, which is very conservative, tends to be detrimental. Finally, both components are turned off in a few other datasets, such as *pappas\_ted* and *sanders*. In these datasets, these components tend not to have much impact.

### 5.3 Effectiveness Comparison

Table 4 shows the MacroF1 effectiveness results. Best results in all datasets (including ties) are marked in **bold**. As we can see, CluSent (fasText) is the best overall method – it outperforms the baselines with three overall wins (statistically superior results over all others ▲) and 12 ties in first (best) place (●), considering the 19 datasets. In other words, CluSent was the best method in 15 out of 19 cases, either in isolation or tied with some other method.

In the cases in which CluSent outperformed the best baseline (runner-up method) in each dataset, it did by large margins, such as in *ss\_bbc* with gains of 14.21% over KNN Regression, 7.60% in *ss\_digg* over RNTN, and 7.2% in *ss\_rw* over BERT. Among the three CluSent’s losses, one was only against L-MIXED (in *vader\_movie*), in *stanford\_tw* against L-MIXED and kNN Regression Expansion, in *vader\_nyt* against GloVE, and *SST-2* against BERT and L-MIXED. This analysis also emphasizes L-MIXED as the strongest of the baselines, with 12 ties in the first place, five losses, and only two wins when directly compared with CluSent. Remind that L-MIXED is considered a very solid SOTA baseline in public benchmarks.

BERT and kNN Regression Expansion lost to CluSent in most cases - 9 and 10 losses, respectively. BERT only surpassed CluSent in *SST-2*, tying with L-MIXED, while KNN Regression outperformed CluSent only in *stanford\_tw*.

As observed, the baseline approaches demonstrated statistically superior results against CluSent in only three datasets, namely *stanford\_tw*, *vader\_movie*, and *SST-2*. Particularly, the *stanford\_tw* dataset contains only 359 instances obtained from 72 arbitrary twitter queries about multiple domains using names of locations, people, movies, products, etc. [Go et al., 2009]. Due to the substantial observed heterogeneity, CluWords are more susceptible to clustering embeddings of words from unrelated domains,

**Table 4.** MacroF1 results. CluSent is the best method (winning or tying) in 16 out of 19 datasets. ♦ indicates the supervised methods, while ■ indicates the unsupervised methods.

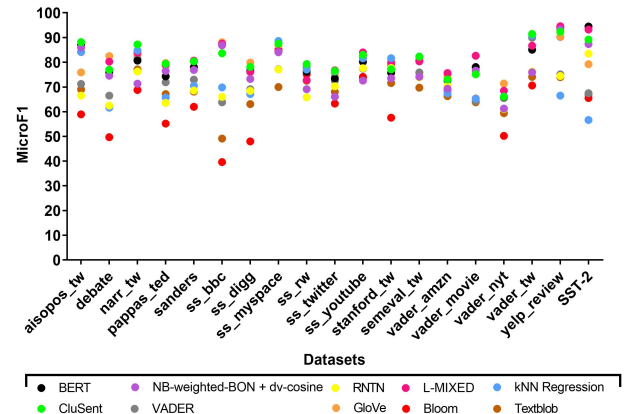
Dataset	BERT ♦	NB-W-B + dv-cos ♦	RNTN ♦	L-MIXED ♦	kNN Reg. Exp. ♦	TextBlob ■	VADER ■	GloVe ♦	Bloom ■	CluSent ♦
aisopos_tw	<b>86.73</b>	<b>84.74</b>	63.63	83.58	82.95	68.60	70.70	75.10	42.00 *	<b>87.74</b> ●
debate	73.79	66.42	62.4	<b>77.41</b>	61.53	59.80	64.90	<b>80.30</b>	47.40	<b>75.13</b> ●
narr_tw	79.71	63.42	74.12	<b>82.48</b>	<b>83.46</b>	76.60	80.90	81.80	57.70	<b>86.50</b> ●
pappas_ted	73.52	74.85	63.42	<b>77.64</b>	65.43	67.00	71.80	<b>79.00</b>	50.20	<b>78.82</b> ●
sanders	<b>78.07</b>	76.29	68.02	<b>80.47</b>	69.81	68.00	73.00	<b>80.20</b>	58.70	<b>80.37</b> ●
ss_bbc	55.99	46.48	55.55	51.28	60.36	44.30	54.60	58.70	38.20	<b>68.94</b> ▲
ss_digg	65.68	43.20	66.05	55.87	65.55	59.80	65.80	68.50	47.90	<b>71.07</b> ▲
ss_myspace	61.02	45.67	62.47	49.88	<b>75.35</b>	60.70	68.30	63.40	54.00	<b>73.35</b> ●
ss_rw	70.56	42.12	62.90	57.72	67.53	68.00	73.00	65.60	62.00	<b>75.62</b> ▲
ss_twitter	72.21	55.99	68.17	<b>74.81</b>	<b>73.94</b>	67.60	71.50	<b>73.90</b>	49.20	<b>75.44</b> ●
ss_youtube	76.55	54.40	71.31	<b>79.69</b>	<b>77.09</b>	70.90	75.00	<b>78.80</b>	58.30	<b>79.02</b> ●
stanford_tw	75.70	72.88	77.52	<b>79.54</b>	<b>81.41</b>	71.50	<b>78.50</b>	<b>80.70</b>	47.80	77.07 ▼
semeval_tw	<b>74.09</b>	48.60	68.92	68.37	<b>75.52</b>	66.60	71.60	73.40	54.80	<b>76.51</b> ●
vader_amzn	<b>71.48</b>	62.85	69.33	<b>73.89</b>	62.49	65.70	68.00	<b>73.40</b>	60.20	<b>71.94</b> ●
vader_movie	78.09	76.59	75.31	<b>82.63</b>	64.59	63.60	64.10	76.90	61.10	75.11 ▼
vader_nyt	65.56	53.19	60.92	66.92	66.00	58.90	64.50	<b>70.80</b>	42.00	65.56 ▼
vader_tw	81.92	61.23	71.67	82.53	<b>89.25</b>	72.70	88.70	81.80	61.30	<b>89.63</b> ●
yelp_review	<b>94.08</b>	<b>93.30</b>	74.33	<b>94.59</b>	62.46	66.70	73.70	90.40	90.20	<b>92.36</b> ●
SST-2	<b>94.39</b>	86.87	82.75	<b>93.13</b>	55.11	72.50	67.30	79.20	60.10	89.02 ▼

leading to poor semantic expansions. Our filtering methods aim to avoid such poor semantic expansions. However, despite the potential improvements of our proposed filtering approaches on stanford\_tw (as shown in Table 3), the proposed Auto CluSent incorrectly estimates (due to the small and heterogeneous training data) instantiations of CluSent that automatically disables such filtering methods, leading to an average decrease of 3.7% in MacroF1. Without such a decrease, there would be no statistically significant gains by the baselines compared to our proposal on the stanford\_tw dataset. This also indicates that there is room for improvements in our Auto CluSent approach in future work.

In contrast, our filtering steps do not produce any effect on the vader\_movie dataset (as shown in Table 3), which suggests the prevalence of high-quality and semantically related groups of word embeddings within the dataset vocabulary. Such conformity can be explained by the composition of the final version of vader\_movie [Hutto and Gilbert, 2014], which comprises 10,605 (sentence-level pre-processed) snippets extracted from 2000 movie reviews written by only 312 authors [Pang and Lee, 2004]. In such a scenario, multiple instances are sampled from the same review/reviewer, which mitigates the need for semantic expansions and noise filtering. The best baseline (L-MIXED) successfully exploits the bi-LSTM advantage of identifying common sequential patterns among multiple sentence-level snippets sampled from the same review.

Similarly to vader\_movie, the filtering steps do not produce any effect on the SST-2 dataset (as shown in Table 3), also suggesting the prevalence of high-quality and semantically related groups of word embeddings within the dataset vocabulary. As in vader\_movie, the highly related instances are produced by multiple samples from the same reviews. In fact, despite the large number of instances in the final version of the SST-2 [Wang et al., 2018], its 68,221 instances are very small phrases sampled from only 11,855 sentence-level snippets extracted from movie reviews. In such a scenario, the common distribution of samples mitigates the need for semantic expansions and noise filtering. The best baselines (L-MIXED and BERT) both successfully exploit the advantage of representing common contextual patterns in small sequences sampled from the same review.

Figure 4 shows the effectiveness of the results in terms of MicroF1 (Accuracy). In this scenario, CluSent tied for first place in 13 out of 19 cases, twelve of them with L-MIXED, the strongest baseline in terms of MicroF1. This result makes CluSent the best overall method along with L\_MIXED. The slightly better CluSent results in terms of MacroF1 when compared to MicroF1 is due to the high skewness (class imbalance) of some datasets (e.g., debate, ss\_bbc, ss\_myspace). When faced with an information shortage, there is a tendency to increase the classifier’s natural bias towards the largest class. The CluSent semantic expansion helps counterbalance this natural bias, making the classification fairer to the minority class. This fact is better reflected in the MacroF1 results.

**Figure 4.** MicroF1 results. CluSent is the best method, tied with L-Mixed, winning or tying in 14 out of 19 datasets.

To summarize the results we perform an analysis using *Fractional rankings* to determine the most effective overall method across the multiple datasets. In Fractional rankings, items that perform equally (i.e., statistical ties) receive the same ranking number, the mean of the ranking they would receive under ordinal rankings considering the ties. In our scenario, we rank each method for each dataset based on the MacroF1 score and the statistical tests. As mentioned, ties receive the same rank position.

Table 5 shows the fractional ranking for the MacroF1 results, and the last row, called Aggregated (Aggr.) Ranking

**Table 5.** Fractional Rank for MacroF1 results. CluSent is the best overall method in the Aggregated Ranking. The  $R^{(*)}$  indicates the rank position.

Dataset	BERT	NB-W-B + dv-cos	RNTN	L-MIXED	kNN Reg. Exp.	TextBlob	VADER	GloVe	Bloom	CluSent
aisopos_tw	2	2	9	4	5	8	7	6	10	2
debate	4	5	7	2	8	9	6	2	10	2
narr_tw	6	9	8	3	1.5	7	5	4	10	1.5
pappas_ted	5	4	9	2	8	7	6	2	10	2
sanders	2.5	5	8	2.5	7	9	6	2.5	10	2.5
ss_bbc	4	7	5	8	2	9	6	3	10	1
ss_digg	3	10	4	8	6	7	5	2	9	1
ss_myspace	6.5	10	5	9	1.5	6.5	3	4	8	1.5
ss_rw	2	10	7	9	5	4	3	6	8	1
ss_twitter	5	9	7	2.5	2.5	8	6	2.5	10	2.5
ss_youtube	3	10	7	3	3	8	6	3	9	3
stanford_tw	7	8	5	2	2	9	4	2	10	6
semeval_tw	2.5	10	6	7	2.5	8	5	2.5	9	2.5
vader_amzn	2.5	8	5	2.5	9	7	6	2.5	10	2.5
vader_movie	2	4	5	1	7	9	8	3	10	6
vader_nyt	4.5	9	7	2	3	8	6	1	10	4.5
vader_tw	5	10	8	4	1.5	7	3	6	9	1.5
yelp_review	2.5	2.5	7	2.5	10	9	8	5	6	2.5
SST-2	1.5	4	5	1.5	10	8	7	6	9	3
Aggr. Ranking	70.5 $R^{(3)}$	136.5 $R^{(8)}$	124 $R^{(7)}$	75.5 $R^{(4)}$	94.5 $R^{(5)}$	147.5 $R^{(9)}$	106 $R^{(6)}$	65 $R^{(2)}$	177 $R^{(10)}$	48.5 $R^{(1)}$

is the summation of all datasets’ rankings for each method. For instance, in *ss\_bbc*, *ss\_digg* and *ss\_rw* where CluSent is the sole best method with no tie, it receives a ranking of 1 while in *narr\_tw*, *pappas\_ted*, *ss\_myspace*, and *vader\_tw*, where Clusent ties as the best method with another baseline, it receives a ranking of 1.5 (Rank: 1.5, 1.5, 3, ...).

As can be seen in the Aggregated Ranking, CluSent is by far the best overall method (lowest aggregated ranking: 48.5) considering the 19 datasets, with GloVe coming in a distant second place (Aggr. ranking: 65.0) and BERT in third place (Aggr. ranking: 70.5). This analysis emphasizes CluSent’s consistency across many different domains and scenarios.

## 5.4 Difficult cases solved by CluSent

As an example of a difficult case that CluSent can handle and other methods can not, in *ss\_bbc*. The negative document “that’s why the meeting may well be just a joke” has been misclassified by a simple base classifier (Linear SVM). CluSent expanded the original document representation into a vector with 47 non-zero new dimensions related to the semantic neighborhood, including new words such as “silly” and “apology”. This information and the weighting step allowed it to correct the misclassification.

Another example in the same dataset is the document “Science once again ignored by the mainstream so they can continue to collect dollars with the marketing of the green business agenda.”. Comparing the CluSent with Linear SVM, we observe that CluSent added more negative information, such as “abandoned”, “blinded”, and “blurred”. The filters also removed positive words in the same neighborhood, i.e., no positive words were added. Both actions helped to correct SVM’s misclassification.

**Table 6.** Density analysis of the CluWord Instantiations

Dataset	Density			
	CW	CW + PoS	CW + TF-AL	CW + PoS + TF-AL
ss_bbc	3,916	2,608	1,957	1,192
ss_digg	1,964	1,167	910	488
ss_myspace	1,126	661	510	268
ss_rw	2,745	1,758	1,328	770
vader_movie	4,338	2,164	1,649	766

## 5.5 Density Analysis of the CluSent Representation

Table 6 shows the document density (average number of words per document) for some CluSent instantiations in some of the datasets in which CluSent outperforms all baselines by large margins – *ss\_bbc*, *ss\_digg*, *ss\_myspace*, and *ss\_rw*). When compared with their original density (Table 1), we can see that the density increases considerably in these datasets, regardless of the CluSent instantiation. For instance, in *ss\_myspace*, the density of documents increases by at least 1822%. This is a direct consequence of Cluwords’s (CW) semantic expansion.

We also notice that in the CW + TF, CW + Pos + TF-AL, and the CluSent instantiations, the sentiment-based filtering and weighting mechanism was turned on in *ss\_bbc*, *ss\_digg*, *ss\_myspace*, and *ss\_rw*. In Table 6, the document density of the CluSent instantiation was further broken down into two sentiment polarities (positive and negative). The TF-AL instantiation makes it possible to identify the polarity of words based on the lexical dictionary used by the method.

As we can see, in all the cases shown in Table 6, the density was reduced due to the *noise filtering* mechanism, but it is still much higher than in the original representation. This justifies the robustness of CluSent in capturing the best

document representation – semantically expanded, with noise removal and sentiment-based weighting, ultimately justifying its effectiveness.

Table 6 also includes information about one dataset in which CluSent did not surpass the baselines – *vader\_movie*. We can see in Table 1 that this dataset is larger, more balanced, and have already a high density. In other words, it suffers less from information shortage problems. Although there was a similar expansion in *vader\_movie*, the sentiment-based filtering/weighting was turned off in this dataset (indicated as '-' in Table 6). This may suggest that the CluWord expansion in this dataset produced a less noisy representation. However, this expansion was not enough to surpass the baselines, which took advantage of the higher amount of information in this dataset. This indicates room for further improvements in expansion and filtering strategies for CluSent.

## 6 Conclusion

We proposed a new solution for sentiment analysis – CluSent – that exploits semantic expansion and tackles issues of information shortage and noise. It combines supervised and unsupervised solutions, taking advantage of external information from word embeddings and unsupervised lexicons. CluSent generalizes and expands the CluWords concept to sentiment analysis in a dataset-oriented manner. Indeed, our novel framework can be adapted to different NLP tasks/applications and the idiosyncrasies of each dataset by turning on/off its steps depending on the characteristics of the dataset.

In our experiments, CluSent outperformed strong baselines in 28 out of 38 possibilities, excelling in a Fractional Ranking aggregated analysis, with gains of more than 14% against some of the best baselines. Our analyses show that all components of our solution are important for the final results and that the ability to adapt the solution to different datasets' idiosyncrasies is key to CluSent's success.

In future work, we will exploit CluSent's ideas in other classification tasks, e.g., topic classification. We also want to investigate other manners to exploit contextual embeddings (other than simple average pooling) to see whether they can improve effectiveness. The use of Large Language Models (LLMs) to infer the polarities of words in order to improve the coverage of the used lexicons is also an idea worth trying. Finally, we have seen that there is space to improve the filtering and weighting steps on top of the Cluwords' semantic expansion, and this is a venue we will certainly exploit.

## Acknowledgements

This work was partially supported by CNPq, CAPES, Fapemig, AWS, and Google Research Awards.

## References

Abiola, O., Abayomi-Alli, A., Tale, O. A., Misra, S., and Abayomi-Alli, O. (2023). Sentiment analysis of

covid-19 tweets from selected hashtags in nigeria using vader and text blob analyser. *Journal of Electrical Systems and Information Technology*, 10(1):5. DOI: <https://doi.org/10.1186/s43067-023-00070-9>.

Alissa, M., Haddad, I., Meyer, J., Obeid, J., Vialaetis, K., Wiecek, N., and Wongariyakavee, S. (2021). Sentiment analysis for open domain conversational agent. *CoRR*, abs/2101.00675. DOI: <https://doi.org/10.48550/arXiv.2101.00675>.

Aljedaani, W., Rustam, F., Mkaouer, M. W., Ghallab, A., Rupapara, V., Washington, P. B., Lee, E., and Ashraf, I. (2022). Sentiment analysis on twitter data integrating textblob and deep learning models: The case of us airline industry. *Knowledge-Based Systems*, 255:109780. DOI: <https://doi.org/10.1016/j.knosys.2022.109780>.

Amin, A., Hossain, I., Akther, A., and Alam, K. M. (2019). Bengali vader: A sentiment analysis approach using modified vader. In *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pages 1–6. DOI: <https://doi.org/10.1109/ECACE.2019.8679144>.

Arkin, E. M., Banik, A., Carmi, P., Citovsky, G., Katz, M. J., Mitchell, J. S., and Simakov, M. (2018). Selecting and covering colored points. *Discrete Applied Mathematics*, 250:75–86. DOI: <https://doi.org/10.1016/j.dam.2018.05.011>.

Bommasani, R., Davis, K., and Cardie, C. (2020). Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781, Online. Association for Computational Linguistics. DOI: <https://doi.org/10.18653/v1/2020.acl-main.431>.

Cunha, W., França, C., Fonseca, G., Rocha, L., and Gonçalves, M. A. (2023a). An effective, efficient, and scalable confidence-based instance selection framework for transformer-based text classification. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 665–674. DOI: <https://doi.org/10.1145/3539618.3591638>.

Cunha, W., França, C., Rocha, L., and Gonçalves, M. A. (2023b). Tpd: A novel two-step transformer-based product and class description match and retrieval method. *arXiv preprint arXiv:2310.03491*. DOI: <https://doi.org/10.48550/arXiv.2310.03491>.

Cunha, W., Mangaravite, V., Gomes, C., Canuto, S., Resende, E., Nascimento, C., Viegas, F., França, C., Martins, W. S., Almeida, J. M., Rosa, T., Rocha, L., and Gonçalves, M. A. (2021). On the cost-effectiveness of neural and non-neural approaches and representations for text classification: A comprehensive comparative study. *IP&M*, 58(3):102481. DOI: <https://doi.org/10.1016/j.ipm.2020.102481>.

Cunha, W., Viegas, F., França, C., Rosa, T., Rocha, L., and Gonçalves, M. A. (2023c). A comparative survey of instance selection methods applied to nonneural and transformer-based text classification. *ACM Computing*

- Surveys*. DOI: <https://doi.org/10.1145/3582000>.
- de Andrade, C. M., Belém, F. M., Cunha, W., França, C., Viegas, F., Rocha, L., and Gonçalves, M. A. (2023). On the class separability of contextual embeddings representations—or “the classifier does not matter when the (text) representation is so good!”. *Information Processing & Management*, 60(4):103336. DOI: <https://doi.org/10.1016/j.ipm.2023.103336>.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186. DOI: <https://doi.org/10.18653/v1/N19-1423>.
- Edwards, A., Camacho-Collados, J., De Ribaupierre, H., and Preece, A. (2020). Go simple and pre-train on domain-specific corpora: On the role of training data for text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5522–5529. DOI: <https://doi.org/10.18653/v1/2020.coling-main.481>.
- Figueiredo, F., Rocha, L., Couto, T., Salles, T., Gonçalves, M. A., and Meira Jr., W. (2011). Word co-occurrence features for text classification. *Inf. Syst.*, 36. DOI: <https://doi.org/10.1016/j.is.2011.02.002>.
- Foster, C. and Kimia, B. (2023). Computational enhancements of hnsw targeted to very large datasets. In Pedreira, O. and Estivill-Castro, V., editors, *Similarity Search and Applications*, pages 291–299, Cham. Springer Nature Switzerland. DOI: [https://doi.org/10.1007/978-3-031-46994-7\\_25](https://doi.org/10.1007/978-3-031-46994-7_25).
- Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.
- Hu, X., Sun, N., Zhang, C., and Chua, T.-S. (2009). Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *Proceedings of CIKM*, pages 919–928. ACM. DOI: <https://doi.org/10.1145/1645953.1646071>.
- Huang, Q., Chen, Z., Lu, Z., and Ye, Y. (2018). Analysis of bag-of-n-grams representation’s properties based on textual reconstruction. *CoRR*. DOI: <https://doi.org/10.48550/arXiv.1809.06502>.
- Hutto, C. J. and Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *ICWSM’14*. DOI: <https://doi.org/10.1609/icwsm.v8i1.14550>.
- Jin, Z., Zhao, X., and Liu, Y. (2021). Heterogeneous graph network embedding for sentiment analysis on social media. *Cognitive Computation*, 13(1):81–95. DOI: <https://doi.org/10.1007/s12559-020-09793-7>.
- Jonker, R. A. A., Poudel, R., Fajarda, O., Matos, S., Oliveira, J. L., and Lopes, R. P. (2022). Portuguese twitter dataset on covid-19. In *2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 332–338. DOI: <https://doi.org/10.1109/ASONAM55673.2022.10068592>.
- Júnior, A. P. D. S., Cecilio, P., Viegas, F., Cunha, W., Albergaria, E. T. D., and Rocha, L. C. D. D. (2022). Evaluating topic modeling pre-processing pipelines for portuguese texts. In *Proceedings of the Brazilian Symposium on Multimedia and the Web*, pages 191–201. DOI: <https://doi.org/10.1145/3539637.3557052>.
- Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *JMLR.*, 5:361–397. DOI: <https://doi.org/10.5555/1005332.1005345>.
- Loureiro, D. and Camacho-Collados, J. (2020). Don’t neglect the obvious: On the role of unambiguous words in word sense disambiguation. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3514–3520, Online. Association for Computational Linguistics. DOI: <https://doi.org/10.18653/v1/2020.emnlp-main.283>.
- Ma, T., Yao, J.-G., Lin, C.-Y., and Zhao, T. (2021). Issues with entailment-based zero-shot text classification. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 786–796, Online. Association for Computational Linguistics. DOI: <https://doi.org/10.18653/v1/2021.acl-short.99>.
- Mabrouk, A., Redondo, R. P. D., and Kayed, M. (2020). Deep learning-based sentiment classification: A comparative survey. *IEEE Access*, 8:85616–85638. DOI: <https://doi.org/10.1109/ACCESS.2020.2992013>.
- Malkov, Y. A. and Yashunin, D. A. (2020). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(4):824–836. DOI: <https://doi.org/10.1109/TPAMI.2018.2889473>.
- Mikolov, T., Grave, E., Bojanowski, P., Puhersch, C., and Joulin, A. (2018). Advances in pre-training distributed word representations. In *LREC’18*. DOI: <https://doi.org/10.48550/arXiv.1712.09405>.
- Nooralahzadeh, F., Øvreliid, L., and Lønning, J. T. (2018). Evaluation of Domain-specific Word Embeddings using Knowledge Resources. In chair), N. C. C., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., and Tokunaga, T., editors, *LREC’18*, Miyazaki, Japan. ELRA.
- Oyebode, O. and Orji, R. (2019). Social media and sentiment analysis: The nigeria presidential election 2019. In *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 0140–0146. DOI: <https://doi.org/10.1109/IEMCON.2019.8936139>.
- Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain. DOI: <https://doi.org/10.3115/1218955.1218990>.
- Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In Moschitti, A., Pang, B., and Daelemans, W., editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha,

- Qatar. Association for Computational Linguistics. DOI: <https://doi.org/10.3115/v1/D14-1162>.
- Puri, R. and Catanzaro, B. (2019). Zero-shot text classification with generative language models. *CoRR*, abs/1912.10165. DOI: <https://doi.org/10.48550/arXiv.1912.10165>.
- Qi, Y. and Shabrina, Z. (2023). Sentiment analysis using twitter data: a comparative application of lexicon- and machine-learning-based approach. *Social Network Analysis and Mining*, 13(1):31. DOI: <https://doi.org/10.1007/s13278-023-01030-x>.
- Ribeiro, F. N., Araújo, M., Gonçalves, P., Gonçalves, M. A., and Benevenuto, F. (2016). Sentibench: A benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, 5(1):1–29. DOI: <https://doi.org/10.1140/epjds/s13688-016-0085-1>.
- Rosenthal, S., Farra, N., and Nakov, P. (2019). Semeval-2017 task 4: Sentiment analysis in twitter. *CoRR*, abs/1912.00741. DOI: <https://doi.org/10.18653/v1/S17-2088>.
- Sachan, D. S., Zaheer, M., and Salakhutdinov, R. (2019). Revisiting lstm networks for semi-supervised text classification via mixed objective function. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6940–6948. DOI: <https://doi.org/10.1609/aaai.v33i01.33016940>.
- Shaik, T., Tao, X., Dann, C., Xie, H., Li, Y., and Galligan, L. (2023). Sentiment analysis and opinion mining on educational data: A survey. *Natural Language Processing Journal*, 2:100003. DOI: <https://doi.org/10.1016/j.nlp.2022.100003>.
- Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP'19*, pages 1631–1642, Seattle, Washington, USA. ACL.
- Thongtan, T. and Phienthrakul, T. (2019). Sentiment classification using document embeddings trained with cosine similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 407–414, Florence, Italy. Association for Computational Linguistics. DOI: <https://doi.org/10.18653/v1/P19-2057>.
- Tymann, K., Lutz, M., Palsbröcker, P., and Gips, C. (2019). Gervader - A german adaptation of the VADER sentiment analysis tool for social media texts. In Jäschke, R. and Weidlich, M., editors, *Proceedings of the Conference on "Lernen, Wissen, Daten, Analysen", Berlin, Germany, September 30 - October 2, 2019*, volume 2454 of *CEUR Workshop Proceedings*, pages 178–189. CEUR-WS.org.
- Viegas, F., Alvim, M. S., Canuto, S., Rosa, T., Gonçalves, M. A., and Rocha, L. (2020a). Exploiting semantic relationships for unsupervised expansion of sentiment lexicons. *Information Systems*, 94:101606. DOI: <https://doi.org/10.1016/j.is.2020.101606>.
- Viegas, F., Canuto, S., Cunha, W., França, C., Valiense, C., Rocha, L., and Gonçalves, M. A. (2023). Clusent – combining semantic expansion and de-noising for dataset-oriented sentiment analysis of short texts. In *Proceedings of the 29th Brazilian Symposium on Multimedia and the Web*, WebMedia '23, page 110–118, New York, NY, USA. Association for Computing Machinery. DOI: <https://doi.org/10.1145/3617023.3617039>.
- Viegas, F., Canuto, S., Gomes, C., Luiz, W., Rosa, T., Ribas, S., Rocha, L., and Gonçalves, M. A. (2019). Cluwords: Exploiting semantic word clustering representation for enhanced topic modeling. In *Proceedings of WSDM '19*, pages 753–761. DOI: <https://doi.org/10.1145/3289600.3291032>.
- Viegas, F., Cunha, W., Gomes, C., Pereira, A., Rocha, L., and Gonçalves, M. (2020b). CluHTM - semantic hierarchical topic modeling based on CluWords. In *Proc. of the 58th Annual Meeting of the Assoc. for Computational Linguistics (ACL 2020)*, pages 8138–8150. Association for Computational Linguistics. DOI: <https://doi.org/10.18653/v1/2020.acl-main.724>.
- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Linzen, T., Chrupala, G., and Alshahi, A., editors, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics. DOI: <https://doi.org/10.18653/v1/W18-5446>.
- Wang, Y., Yin, F., Liu, J., and Tosato, M. (2020). Automatic construction of domain sentiment lexicon for semantic disambiguation. *Multim. Tools Appl.*, 79(31-32):22355–22373. DOI: <https://doi.org/10.1007/s11042-020-09030-1>.
- Yin, D., Meng, T., and Chang, K.-W. (2020). SentBERT: A transferable transformer-based architecture for compositional sentiment semantics. In *Proceedings of the 58th Conference of the Association for Computational Linguistics, ACL 2020, Seattle, USA*. DOI: <https://doi.org/10.18653/v1/2020.acl-main.341>.
- Yin, W., Hay, J., and Roth, D. (2019). Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics. DOI: <https://doi.org/10.18653/v1/D19-1404>.
- Yong, Z. X., Schoelkopf, H., Muennighoff, N., Aji, A. F., Adelani, D. I., AlmuBarak, K., Bari, M. S., Sutawika, L., Kasai, J., Baruwa, A., Winata, G., Biderman, S., Raff, E., Radev, D., and Nikoulina, V. (2023). BLOOM+I: Adding language support to BLOOM for zero-shot prompting. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11682–11703, Toronto, Canada. Association for Computational Linguistics. DOI: <https://doi.org/10.18653/v1/2023.acl-long.653>.