CluSent – Combining Semantic Expansion and De-Noising for Dataset-Oriented Sentiment Analysis of Short Texts

Anonymous ACL submission

Abstract

The lack of sufficient information, mainly in short texts, is a major challenge to building effective sentiment models. Short texts can be enriched with more complex semantic relationships that can better capture affective information, with a potential undesired side effect of noise introduced into the data. In this work, we propose a new strategy for customized dataset-oriented sentiment analysis – **CluSent** – that exploits a powerful, recently proposed concept for representing semantically related words – CluWords. CluSent tackles the issues mentioned above of information shortage and noise by: (i) exploiting the semantic neighborhood of a given pre-trained word embedding to enrich document representation, and (ii) introducing dataset-oriented filtering and weighting mechanisms to cope with noise, which take advantage of the polarity and intensity information from lexicons. In our experimental evaluation, considering 19 datasets, 5 state-of-the-art baselines (including modern transformer architectures) and two metrics, CluSent was the best method in 30 out of 38 possibilities, with significant gains over the strongest baselines (over 14%).

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 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 class (positive or negative) associated with a document. In this domain, lexicon limitations such as the *coverage problem* affect the number of words covered by these lexicons considering the dataset vocabulary. Strategies 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. This paper focuses on supervised strategies, but we take full advantage of the information in unsupervised lexicons to build our novel solutions.

A challenging scenario to build effective (supervised or unsupervised) sentiment models is related to short texts due to their present lack of sufficient information to measure the overall sentiment of a document (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 (Huang et al., 2018). These simple models based on positional information cannot 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 models based on Transformer architectures (e.g., BERT (Devlin et al., 2018)).

An undesired side effect of such expansion/enrichment strategies is the possibility of the introduction of *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 is the proposal of a new solution for sentiment analysis – *CluSent* – that exploits a powerful, recently proposed concept – *CluWords* (Viegas et al., 2019) – to tackle the aforementioned issues of information shortage and noise. The main idea is to exploit the similarity relationships between words on pre-trained embeddings by expanding terms with their closely related neighbors to improve both the occurrence and discriminative power of words in short texts. The CluSent representation exploits the nearest words of a given pre-trained word embedding to generate "meta-words" to expand and enhance the document representation in terms of 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 words from the cluster for 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 rich word representations for the sake of sentiment analysis. 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 the filtering and weighting mechanisms to the sentiment analysis problem. To do this, we propose a new TFIDF-like representation that exploits polarity and intensity, what we call $TF\text{-}AL$. We use this $TF\text{-}AL$ concept as **a filtering/weighting mechanism** in the CluSent representation. The idea here is to build *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 the same polarity and similar intensity would belong to the same CluWord for this task. In sum, we exploit information in the sentiment lexicon, i.e., polarity and the lexicons' intensity, to filter out words from a CluWord. The intensity is also used as a weighting measure for each CluWord, collaboratively with the semantic information. All these innovations are encapsulated into **CluSent**, our novel solution for sentiment analysis, which, besides all that, also incorporates a dynamic instantiation pipeline to build dataset-oriented document representations.

In our experimental evaluation, comparing *CluSent* with five strong state-of-the-art sentiment analysis baselines in a large benchmark with 19 datasets, our solution achieved the best results in 30 out 38 possibilities (19 datasets considering MacroF1 and MicroF1), 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 guarantee the reproducibility of our solution, all the code, the documentation of how to run it and datasets are available on github[1].

To summarize, our main contributions include: (i) the *proposal* of the CluSent method to build document representations for sentiment analysis that use information from multiple word embeddings; (ii) the *exploration* of the powerful concept of CluWords combined with sentiment lexicon´s polarity and intensity to tackle the problems of information shortage and noise, commonly found in sentiment analysis applications; (iii) 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.

2 Related Work

We review the CluWords concept and the state-of-the-art (SOTA) strategies in sentiment analysis directly comparable to CluSent. CluWords correspond to clusters of semantically related word embeddings (Mikolov et al., 2018) built by employing distance functions[2]. CluWords have been successfully applied in the realm of topic modeling and hierarchical topic modeling scenarios (Viegas et al., 2020b, 2019). One of our main contributions in this paper is a proposal of how to extend the Cluwords concept with dataset-oriented and task-oriented filtering and weighting mechanisms for specific applications, illustrating these extensions in the sentiment analysis task.

[1] https://github.com/link – It will be available in the camera ready version

[2] CluWords are not limited by any particular type of word embedding or distance function, being flexible enough to accommodate many options.

BERT (Devlin et al., 2018) [3], is an end-to-end deep learning classifier. 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 backward. 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. Then, the model learns the composition of meaning by predicting 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[4]. Thus, we include **BERT** as a baseline. Based on the experiments available in (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, 2019) proposed NB-weighted-BON[5], 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 when compared to strong baselines, including BERT. In some comparative analyses, **NB-weighted-BON** is the current state-of-the-art (best-known algorithm) in several sentiment analysis benchmarks, such as in sentiment analysis reviews[6]. We include it as a baseline in our experiments.

In Socher et al. (Socher et al., 2013) the authors proposed the Recursive Neural Tensor Network (**RNTN**). RNTN uses a tree where each node contains a word, its sentiment and associated label (positive, negative, neutral, very positive and very negative). The 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 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 by many recent methods (Alissa et al., 2021; Jin et al., 2021) as a "de facto" baseline to surpass, given its good average results in general. That is why we also exploit **RNTN** as a baseline.

In (Sachan et al., 2019), the authors proposed the L-MIXED[7] strategies that exploit a BiLSTM model with pre-trained embeddings. The idea is to propose a training strategy that 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 there is no previous pretraining step. The authors also proposed a mixed objective function for SSL that utilizes both labeled and unlabeled data to improve the classification. **L-MIXED** is the current SOTA solution (best-known method) in several of the datasets used in our experiments, so we include it as one of the strongest baselines.

Finally, **kNN Expanded Lexicon** (Viegas et al., 2020a) is a recently proposed 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 in the lexicon dictionary). The method uses a nearest neighbors approach to infer the sentiment value of the new lexicons (words with polarity and intensity). To predict the polarity in sentence-level, the method exploited the VADER's shell (Hutto and Gilbert, 2014). The 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 highly effective (Viegas et al., 2020a), this method, similarly to CluSent, exploits word embeddings and distance-based neighborhoods. Therefore we include **kNN Expanded Lexicon** as a close recent SOTA baseline in our experiments.

3 The CluSent Method

Conceptually, CluSent is built by applying three generic steps to a given source text representation: clustering, filtering, and weighting to build a richer (more informative) representation for a textual

[3] Available in https://github.com/yaserkl/

[4] https://github.com/DeepakDhana/SentiALBERT1

[5] https://github.com/tanthongtan/dv-cosine

[6] https://paperswithcode.com/sota/sentiment-analysis-on-imdb

[7] github.com/DevSinghSachan/ssl_text_classification

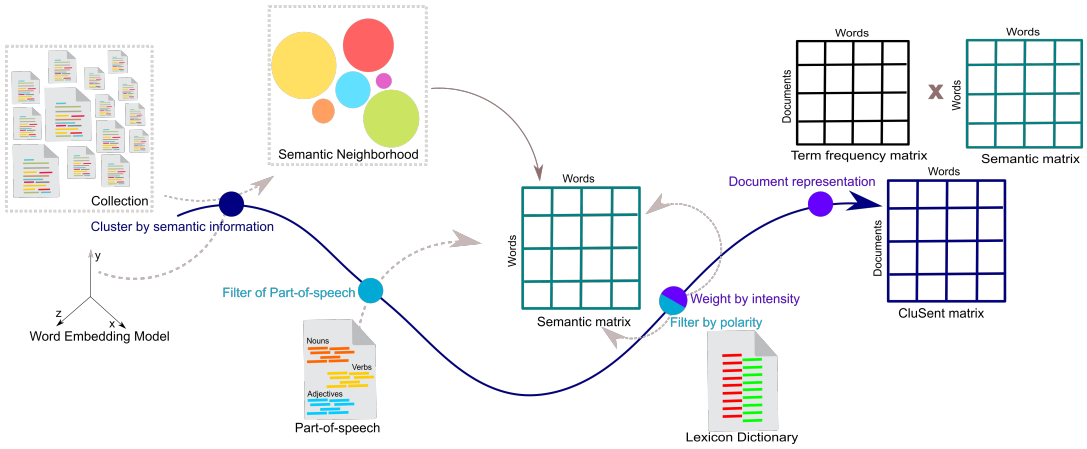

Figure 1: Diagram showing the steps for building the CluSent representation.

collection. Figure 1 illustrates how CluSent representation 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 exploits clusters of semantically related word embeddings (Mikolov et al., 2018) built 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 is coupled with specific weighting schemes[8] used to capture 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) the 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

Let $\mathcal{W}$ be the set of vectors representing each word t in the dataset vocabulary (represented as $\mathcal{V}$). Each word $t \in \mathcal{V}$ has a corresponding vector $u \in \mathcal{W}$. The semantic matrix in the Figure 1 is defined as $C \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, where each dimension has the size of the vocabulary ($|\mathcal{V}|$), t' represents the rows of C while t represents the columns. Finally, each index $C_{t',t}$ is computed according to Eq. 1.

$$C_{t',t} = \begin{cases} \omega(u_{t'}, u_t) & \text{if } \omega(u_{t'}, u_t) \geq \alpha \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where $\omega(u_{t'}, u_t)$ is the cosine similarity defined in Eq. 2 and α is a similarity threshold that acts as a regularizer for the representation. Larger values of α lead to sparser representations. In this notation, each column t of the semantic matrix C will form a CluWord t and each value of the matrix $C_{t',t}$ will receive the cosine similarity between the vectors $u_{t'}$ and u_t in the embedding space $\mathcal{W}$, if it is greater than or equal to α . Otherwise, $C_{t',t}$ receives zero, according to the Eq. 1.

$$\omega(u_{t'}, u_t) = \frac{\sum_i^l u_{t'i} \cdot u_{ti}}{\sqrt{\sum_i^l u_{t'i}^2} \cdot \sqrt{\sum_i^l u_{ti}^2}} \quad (2)$$

The vector $\overrightarrow{C_{,t}}$ represents the semantic information of a *cluster of words* (aka CluWord) t, and the α value filters potential noisy words (i.e., words that do not have a significant relationship with t). Since threshold α is a cosine similarity value, it is contained within the interval $[0, 1]$. If $\alpha = 0$, the similarities of every term in $\mathcal{V}_T$ are included in the CluWord t. If $\alpha = 1$ only the similarity of t to itself (i.e. $\omega(u_{t'}, u_t)$) is included in CluWord t. Thus, the appropriate selection of a value for parameter α is an important aspect of generating "good" CluWord t. Moreover, α controls the sparsity of the resulting document representation. With high α 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 α values, more CluWord terms tend to be related to the document, which reduces the sparsity of the document representation. Note that once we select an appropriate value for α, each CluWord t keeps the values of similarities of the terms most similar to t according to the criteria (e.g., context, co-

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

occurrence) established by the word embeddings.

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 is used to remove pairs of words that do not belong to the same grammatical group. Thus, this filter keeps in a neighborhood of a CluWord t ($\overrightarrow{C_{,t}}$) only terms (t') 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 $pos(.)$ to filter each term t' of $\overrightarrow{C_{,t}}$ that does not belong to the same part-of-speech category of term t (Equation 3). We exploit the Spacy[9] part-of-speech tagger available for the English language to build function $pos(.)$.

$$C_{t',t} = \begin{cases} C_{t',t} & \text{if } pos(t) = pos(t') \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

3.3 Sentiment Filtering and Weighting

Many sentiment analysis approaches of a sentence or document make use of a lexicon dictionary. 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 for the CluSent is to use information from a lexicon dictionary as another filter to remove semantic noise that can affect the quality of the representation, especially in the sentiment analysis scenario. Words with opposite polarities may be co-located in the same neighborhood of a CluWord t since the semantic similarity of embeddings correlated with positional, contextual, and co-occurrence information does not take into account the polarity of a word. Thus, words of opposite polarities may belong to the same CluWord. Indeed this phenomenon has been observed in the literature (Viegas et al., 2020a). We use this filter to keep *polarity consistency* within a CluWord. We go further and 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 $\mathcal{L} = \{\langle w_1, v_1, \rangle, \cdots, \langle w_{|\mathcal{L}|}, v_{|\mathcal{L}|}\rangle\}$, where w_i is a word and v_i is the sentiment value of word w_i, $1 \leq i \leq |\mathcal{L}|$. The sentiment value v_i of a word w_i expresses both 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 $(-4, 4)$. Given the semantic matrix, C, the method exploits Equation 4 to filter terms t' of $\overrightarrow{C_{,t}}$ that does not share the same polarity as term t. In addition, the sentiment value of the term t' is used to weight the semantic value $C_{t',t}$.

$$C_{t',t} = \begin{cases} C_{t',t} \times v_{t'} & \text{if } sign(v_t) = sign(v_{t'}) \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

3.4 Building the CluSent Representation

This step is responsible for building the CluSent representation (the last purple dot in Figure 1) 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 t in document d. Thus, given a CluSent (CS) term t for a document d, its data representation corresponds to

$$CS_{d,t} = \overrightarrow{TF_d} \times \overrightarrow{C_{,t}} \quad (5)$$

where $\overrightarrow{TF_d}$ has the term-frequencies of document d, and $\overrightarrow{C_{,t}}$ is the semantic scores for the term t.

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)[10] 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

[9] https://spacy.io

[10] https://www.kaggle.com/atulanandjha/stanford-sentiment-treebank-v2-sst2

for instance, in a very popular benchmark of unsupervised methods (Ribeiro et al., 2016) as well as in highly cited papers such as (Hutto and Gilbert, 2014). that proposed the VADER lexicon

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., have a skewed distribution increasing the bias towards the largest class.

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

Table 1: Dataset characteristics

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 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.

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 use as baselines popular and SOTA methods such as RNTN, NB-weighted-BON+dv-cosine, kNN Regression Expansion and L-MIXED, based on their performance on public benchmarks. 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, being considered a SOTA baseline in the field. 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 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 most experimented datasets (Viegas et al., 2020a).

For BERT, we configured hyperparameters as suggested by the authors (Devlin et al., 2018). 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 the regularization force. For kNN Regression Expansion, we exploit the pre-trained FastText embedding, and we performed fine-tuning of neighbors according to the author's script in the source-code.

For CluSent, we consider the pre-trained FastText embedding [11] to build the semantic matrix, described in Section 3. 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.

The α parameter (in Eq. 1 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 empiri-

[11] https://dl.fbaipublicfiles.com/fasttext/vectors-english/wiki-news-300d-1M.vec.zip

cally 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, the choice of whether to use the PosTagging filtering and the TF-AL weighting and filtering mechanisms are determined per dataset with nested-cross validation in the training set. 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} by using 5-fold nested cross-validation within the training set.

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, marked with a green triangle (▲), are statistically superior to all others. Statistical ties are represented as a yellow dot (●), while losses are represented as red downward triangles (▼).

4.3 Experimental Results

Dataset	BERT	NB-weighted-BON + dv-cosine	RNTN	L-MIXED	kNN Regression Expansion	CluSent
aisopos_tw	**86.73**	**84.74**	63.63	83.58	82.95	**87.74**●
debate	73.79	66.42	62.4	**77.41**	61.53	**75.13**●
narr_tw	79.71	63.42	74.12	**82.48**	**83.46**	**86.50**●
pappas_ted	73.52	74.85	63.42	**77.64**	65.43	**78.82**●
sanders	**78.07**	76.29	68.02	**80.47**	69.81	**80.37**●
ss_bbc	55.99	46.48	55.55	51.28	60.36	**68.94**▲
ss_digg	65.68	43.20	66.05	55.87	65.55	**71.07**▲
ss_myspace	61.02	45.67	62.47	49.88	**75.35**	**73.35**●
ss_rw	70.56	42.12	62.90	57.72	67.53	**75.62**▲
ss_twitter	72.21	55.99	68.17	**74.81**	**73.94**	**75.44**●
ss_youtube	76.55	54.40	71.31	**79.69**	**77.09**	**79.02**●
stanford_tw	75.70	72.88	77.52	**79.54**	**81.41**	77.07▼
semeval_tw	**74.09**	48.60	68.92	68.37	**75.52**	**76.51**●
vader_amzn	**71.48**	62.85	69.33	**73.89**	62.49	**71.94**●
vader_movie	78.09	76.59	75.31	**82.63**	64.59	75.11▼
vader_nyt	**65.56**	53.19	60.92	**66.92**	**66.00**	**65.56**●
vader_tw	81.92	61.23	71.67	82.53	**89.25**	**89.63**●
yelp_review	**94.08**	**93.30**	74.33	**94.59**	62.46	**92.36**●
SST-2	**94.39**	86.87	82.75	**93.13**	55.11	89.02 ▼

Table 2: MacroF1 results. CluSent is the best method (winning or tying) in 16 out of 19 datasets.

Table 2 shows the MacroF1 effectiveness results. Best results in all datasets (including ties) are marked in **bold**. As we can see, CluSent is the best overall method – it outperforms the baselines with three overall wins (statistically superior results over all others ▲) and 13 ties in first (best) place (●), considering the 19 datasets. In other words, CluSent was the best method in 16 out of 19 cases. L-MIXED was the strongest baseline, with 12 ties, five losses and only two wins when directly compared with CluSent. Remind that L-MIXED is considered a solid SOTA baseline in public benchmarks. BERT and kNN Regression Expansion lost to CluSent in most cases (9 and 10 losses), with nine and eight ties, respectively. BERT only surpassed Clusent in *SST-2*, tying with L-MIXED, while KNN Regression outperformed CluSent only in stanford_tw. In cases in which CluSent outperformed the *best baseline 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 7.17% in *ss_rw* over BERT. Among the three CluSent's losses, one was only against L-MIXED (in *vader_movie*), *stanford_tw* against L-MIXED and kNN Regression Expansion and *SST-2* against BERT and L-MIXED.

Figure 2 shows the effectiveness of the results in terms of MicroF1. In this scenario, CluSent tied in first place in 14 out of 19 cases, twelve of them with L-MIXED, the strongest baseline in terms of MicroF1. This result puts CluSent as the best overall method along with L_MIXED, as detailed in Table 3. The slightly better CluSent's MacroF1 results when compared to MicroF1 may be due to the high skewness (class imbalance) of some datasets (e.g., *debate*, *ss_bbc*, *ss_myspace*). When faced with 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 scores. However, further investigation of this hypothesis is necessary to confirm it. There is a room from improvements and further analysis that will be discussed in the Section 5

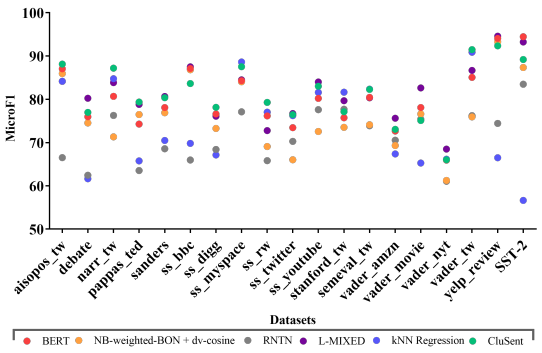

Figure 2: MicroF1 results. CluSent is the runner-up method (winning or tying) in 14 out of 19 datasets.

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, which is the mean of the ranking they would receive under ordinal rankings considering the ties.

Dataset	BERT	NB-weighted-BON + dv-cosine	RNTN	L-MIXED	kNN Regression Expansion	CluSent
aisopos_tw	2.0	2.0	5.5	4.0	5.5	2.0
debate	2.0	5.0	6.0	2.0	4.0	2.0
narr_tw	4.0	6.0	5.0	3.0	1.5	1.5
pappas_ted	3.0	4.0	6.0	1.5	5.0	1.5
sanders	2.0	4.0	6.0	2.0	5.0	2.0
ss_bbc	3.0	6.0	4.0	5.0	2.0	1.0
ss_digg	3.0	6.0	2.0	5.0	4.0	1.0
ss_myspace	4.0	6.0	3.0	5.0	1.5	1.5
ss_rw	2.0	6.0	4.0	5.0	3.0	1.0
ss_twitter	4.0	6.0	5.0	2.0	2.0	2.0
ss_youtube	2.5	6.0	5.0	2.5	2.5	2.5
stanford_tw	3.0	6.0	4.0	1.5	1.5	5.0
semeval_tw	2.0	6.0	4.0	5.0	2.0	2.0
vader_amzn	2.0	5.0	4.0	2.0	6.0	2.0
vader_movie	2.0	3.0	4.0	1.0	6.0	5.0
vader_nyt	2.5	6.0	5.0	2.5	2.5	2.5
vader_tw	4.0	6.0	5.0	3.0	1.5	1.5
yelp_review	2.5	2.5	5.0	2.5	6.0	2.5
SST-2	1.5	4.0	5.0	1.5	6.0	3.0
Aggr. Ranking	**52.0**	**95.5**	**87.5**	**56.0**	**67.5**	**41.5**

Table 3: Fractional Rank for MacroF1 results. CluSent is the best overall method in the Aggregated Ranking.

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 3 shows the fractional ranking for the MacroF1 results, and, the last row, called Aggregated (Aggr.) Ranking, is the ranking 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 it can be seen in the Aggregated Ranking, CluSent is by far the best overall method (lowest aggregated ranking: 41.5) considering the 19 datasets, with BERT coming in a distant second place (Aggr. ranking: 52.0). This analysis emphasizes CluSent´s consistency across many different domains, captured by the different datasets.

4.4 Difficult cases solved by CluSent

As an example of a problematic case that CluSent can handle and other methods can not, in ss_bbc, the raw negative document "that's why the meeting may well be just a joke" has been misclassified by CluSent's 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 combined with 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 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.

4.5 Complexity of CluSent

The complexity of building the clustering step (*Section* 3.1) is basically the nearest neighbor search, which can be exploited by using the fast approximate nearest neighbor search (HNSW) (Malkov and Yashunin, 2018) with complexity of $\mathcal{O}(logN)$. 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 are $\mathcal{O}(NNZ)$, where NNZ represents the non-zero values. Finally, the complexity of *Section* 3.4 is the matrix multiplication ($\overrightarrow{TF_d} \times \overrightarrow{C_{,t}}$) in the Eq. 5). Since both matrices are sparse, the complexity is $\mathcal{O}(NNZ(\overrightarrow{TF_d})NNZ(\overrightarrow{C_{,t}})/|\mathcal{V}|)$ in average, where $|\mathcal{V}|$ is the size of the vocabulary.

5 Conclusion

We proposed a new solution for sentiment analysis – CluSent – that exploits semantic expansion and tackles information shortage and noise issues. CluSent representation is built by a dynamic pipeline of instantiations to build dataset-oriented document representations. 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 proposed novel framework can be adapted to different NLP tasks/applications and the idiosyncrasies of each dataset by turning on/off its steps. In our experiments, CluSent outperformed the evaluated baselines in 30 out of 38 possibilities, excelling in a Fractional Ranking aggregated analysis, with gains of more than 14% against some of the strongest baselines. As future work, we will exploit CluSent in other classification tasks, perform a quantitative analysis of the impact of our solution's components, and combine CluSent with attention models and contextual embeddings (e.g., BERT's) that capture other contextual aspects of words, also aiming at explainability.

References

Mohamad Alissa, Issa Haddad, Jonathan Meyer, Jade Obeid, Kostis Vilaetis, Nicolas Wiecek, and Sukrit Wongariyakavee. 2021. Sentiment analysis for open domain conversational agent.

Washington Cunha, Vítor Mangaravite, Christian Gomes, Sérgio Canuto, Elaine Resende, Cecilia Nascimento, Felipe Viegas, Celso França, Wellington Santos Martins, Jussara M. Almeida, Thierson Rosa, Leonardo Rocha, and Marcos André Gonçalves. 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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fábio Figueiredo, Leonardo Rocha, Thierson Couto, Thiago Salles, Marcos André Gonçalves, and Wagner Meira Jr. 2011. Word co-occurrence features for text classification. *Inf. Syst.*, 36.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *Processing*, 150.

Xia Hu, Nan Sun, Chao Zhang, and Tat-Seng Chua. 2009. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *Proceedings of CIKM*, pages 919–928. ACM.

Qi Huang, Zhanghao Chen, Zijie Lu, and Yuan Ye. 2018. Analysis of bag-of-n-grams representation's properties based on textual reconstruction. *CoRR*.

Clayton J. Hutto and Eric Gilbert. 2014. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *ICWSM'14*.

Zhigang Jin, Xiaofang Zhao, and Yuhong Liu. 2021. Heterogeneous graph network embedding for sentiment analysis on social media. *Cognitive Computation*, 13(1):81–95.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *JMLR.*, 5:361–397.

Alhassan Mabrouk, Rebeca P. Díaz Redondo, and Mohammed Kayed. 2020. Deep learning-based sentiment classification: A comparative survey. *IEEE Access*, 8:85616–85638.

Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *LREC'18*.

Farhad Nooralahzadeh, Lilja Øvrelid, and Jan Tore Lønning. 2018. Evaluation of Domain-specific Word Embeddings using Knowledge Resources. In *LREC'18*, Miyazaki, Japan. ELRA.

Filipe N Ribeiro, Matheus Araújo, Pollyanna Gonçalves, Marcos André Gonçalves, and Fabrício Benevenuto. 2016. Sentibench: A benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, 5(1):1–29.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2019. Semeval-2017 task 4: Sentiment analysis in twitter. *CoRR*, abs/1912.00741.

Devendra Singh Sachan, Manzil Zaheer, and Ruslan Salakhutdinov. 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.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP'19*, pages 1631–1642, Seattle, Washington, USA. ACL.

Tan Thongtan and Tanasanee Phienthrakul. 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.

Felipe Viegas, Mário S. Alvim, Sérgio Canuto, Thierson Rosa, Marcos André Gonçalves, and Leonardo Rocha. 2020a. Exploiting semantic relationships for unsupervised expansion of sentiment lexicons. *Information Systems*, 94:101606.

Felipe Viegas, Sérgio Canuto, Christian Gomes, Washington Luiz, Thierson Rosa, Sabir Ribas, Leonardo Rocha, and Marcos André Gonçalves. 2019. Clusword: Exploiting semantic word clustering representation for enhanced topic modeling. In *Proceedings of WSDM '19*, pages 753–761.

Felipe Viegas, Washington Cunha, Christian Gomes, Antônio Pereira, Leonardo Rocha, and Marcos Goncalves. 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.

Yanyan Wang, Fulian Yin, Jianbo Liu, and Marco Tosato. 2020. Automatic construction of domain sentiment lexicon for semantic disambiguation. *Multim. Tools Appl.*, 79(31-32):22355–22373.

Da Yin, Tao Meng, and Kai-Wei Chang. 2020. SentiBERT: 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*.