# Sentiment Intensity Prediction using Neural Word Embeddings

Amal Htait
amal.htait@strath.ac.uk
University of Strathclyde
Glasgow, UK

Leif Azzopardi
leif.azzopardi@strath.ac.uk
University of Strathclyde
Glasgow, UK

## ABSTRACT

Sentiment analysis is central to the process of mining opinions and attitudes from online texts. While much attention has been paid to the sentiment classification problem, much less work has tried to tackle the problem of predicting the intensity of the sentiment. The go to method is VADER — an unsupervised lexicon based approach to scoring sentiment. However, such approaches are limited because of the vocabulary mismatch problem. In this paper, we present in detail and evaluate our AWESSOME framework (*A Word Embedding Sentiment Scorer Of Many Emotions*) [10] for sentiment intensity scoring, that capitalizes on pre-existing lexicons, does not require training and provides fine grained and accurate sentiment intensity scores of words, phrases and text. In our experiments, we used seven Sentiment Collections to evaluate the proposed approach, against lexicon based approaches (e.g., VADER), and supervised methods such as deep learning based approaches (e.g., SentiBERT). The results show that despite not surpassing supervised approaches, the AWESSOME unsupervised approach significantly outperforms existing lexicon approaches and therefore provides a simple and effective approach for sentiment analysis. The AWESSOME framework can be flexibly adapted to cater for different seed lexicons and different neural word embeddings models in order to produce corpus specific lexicons – without the need for extensive supervised learning or retraining.

## CCS CONCEPTS

• **Information systems** → **Sentiment analysis**; • **General and reference** → **Evaluation**.

## KEYWORDS

Sentiment Intensity, Pre-trained Model Language, Lexicons, BERT, VADER, LabMT

## 1 INTRODUCTION

Sentiment analysis is widely used across a variety of domains, such as sociology, psychology, and marketing to analyze and monitor people's opinions, attitudes and emotions towards people, places, products, etc. [7, 16]. And, with the rise in social media platforms providing large volumes of posts, tweets, blogs, reviews to mine and extract meaning from, sentiment analysis has grown in importance [16]. However, the extremely informal nature of online texts varies significantly from formal texts creating challenges for traditional sentiment analysis techniques, which rely mainly on direct keyword matching and scoring using a highly curated sentiment lexicon combined with a series of crafted rules (e.g. VADER [7], LabMT [4], LIWC [27], etc.). While, these dictionary based approaches required no training and play an essential role in the fast and scalable analysis of large volumes of online posts, they are fundamentally limited. This is due to the *vocabulary mismatch problem*, as the vocabulary of the target text is different from the sentiment lexicons, reducing the methods effectiveness in predicting the sentiment of the phrase or sentence. A sentiment score is generally a polarity (positive or negative) or an intensity (how positive or negative is sentiment, e.g., the score would be between -1 and 1 with -1 is very negative and 1 is very positive). In our work, we focus on sentiment intensity rather then sentiment polarity by giving the sentence a score that would represent the strength of the sentiment. A simple classification of a sentence as positive, negative or neutral would not be enough especially when we need to compare sentences of same sentiment polarity. For example, "*The movie is amazing*" is positively stronger than "*The movie is good*", therefore, has a higher sentiment intensity score.

In this paper, we present and evaluate our configurable framework for scoring the sentiment intensity that combines a seed lexicon, a neural word embedding, and a score function. The advantages of this approach are many fold. First, it benefits from the extensive work performed on lexicon based approaches by directly using their lexicons. Second, it requires no labelled data for supervised machine learning, and thus can be used directly. Third, by using neural word embeddings we overcome both the vocabulary mismatch problem, and the *semantic gap* arising because of a mismatch between the intended meaning of the sentence compared to the words in the sentence (i.e. "*It was a happy accident*" where the word "*accident*" is negative but the overall sentence polarity is positive). Forth, the approach can be bootstrapped to provide a new corpus specific lexicon that can be used with existing methods (e.g. VADER's rule based scoring mechanisms). Within our framework, we draw upon the recent innovations in language modelling and utilize BERT (Bidirectional Encoder Representations from Transformers) [3] and related language models to provide the word embeddings, and capitalise on the human curated sentiment

lexicons from VADER and LabMT to provide seed words and inputs. Our unsupervised approach, AWESSOME (*A Word Embedding Sentiment Scorer Of Many Emotions*) [10], is evaluated on seven sentiment test collections where we compare it to existing unsupervised and supervised methods. The rest of this paper is as follows. Next we present an overview of sentiment analysis methods, before describing our framework in detail, and how to configure it. Then we present our experimental analysis and results. Finally, we wrap up with a summary of this work.

## 2 RELATED WORK

Sentiment analysis has received much attention over the past decades with the rise of social media streams and its applicability to many domains to mine opinions, attitudes, and emotions towards different entities (see [8, 16, 36] for an extensive review). In this paper, our focus is on the sentiment intensity scoring, which has received much less attention – where the challenge is to accurately estimate the intensity (how strongly or weakly) the sentiment is, as opposed to trying to classify the text as positive, negative or neutral. Broadly, there are two main approaches: unsupervised approaches (which tend to be rule based), and supervised approaches (which tend to be machine learning based solutions, but which require extensive training data to be effective). Our goal is to prove that our developed unsupervised approach outperforms current rule based approaches, and is competitive or better than supervised approaches. Below, we provide an overview of relevant works on Sentiment Intensity Scoring (SIS).

### 2.1 Unsupervised approaches

One of the most used unsupervised sentiment analysis approaches is the Lexicon-based (or rule-based) approach [33]. This approach depends on a set of a sentiment lexicon according to which words are classified as positive or negative, along with their equivalent sentiment intensity score. To apply the lexicon-based approach, the input text is usually tokenized into individual words, stop words and punctuation are removed, for the pre-processed text to be ran against the sentiment lexicon which will provide the equivalent emotion while applying inference rules to obtain a combined polarity score for the sentence [1, 7].

In general, researchers tend to favor supervised approaches over unsupervised ones due to their recognised out-performance, while bearing the costly consequences in time, resources and annotating data. For example, in SemEval 2016 Task 7 workshop[1] [14] (*Determining Sentiment Intensity of English and Arabic Phrases*), the only participant team presenting an unsupervised method was the LSIS team [11]. They suggested a lexicon-based approach, supported by the web search engines' ability to find the co-occurrence of words or short phrases with predefined negative and positive words. As for SemEval 2018 Task 1 for "Valence Regression" (V-reg) [2] [20], all participants competed using supervised methods.

One of the few, but widely used, unsupervised models for sentiment intensity classification is VADER [7]. VADER is a simple rule-based model for sentiment analysis, it relies on a sentiment lexicon (7500 records) of gold-standard quality with human-validated valence scores that indicated both the sentiment polarity (positive/negative), and the sentiment intensity on a scale from −4 to +4. For example, the word *good* has a positive valence of 1.9, *great* is 3.1, while *horrible* is −2.5. VADER's sentiment lexicon was compared to seven well-known sentiment analysis lexicons and it proved its well performance, particularly in the social media domain [7].

In a more recent work, [12] presented an unsupervised method to classify the words in book reviews by their sentiment intensity. The method relied on the concepts of seed-words [3] and word embedding, where they considered the sentiment intensity of a word $W$ is equal to the difference between the average of cosine similarities measure of $W$'s vector with the vectors of positive seed-words and the average of cosine similarities measure of $W$'s vector with the vectors of negative seed-words. For that purpose, they manually created two sets of seed-words (positive and negative) adapted to book reviews domain, and created a word embedding model using Word2Vec [19] on 22M Amazon's book reviews [4] [17]. In our recently introduced approach [10], we drew upon the approach proposed in [12], but created a configurable framework where preexisiting validated lexicons can be used as positive and negative seed words, while state of the art language models provide the word embeddings for semantic matching. Our approach, which we present in detail, evaluate and compare to other approaches in this paper, benefits from the simplicity and speed of a lexicon-based approach, while achieving greater coverage and higher accuracy through the pre-trained language model.

### 2.2 Supervised Approaches

SemEval workshops had an important contribution in increasing the interest in the intensity aspect of sentiment. SemEval 2016 Task 7 had the objective of evaluating the ability of automatic system to predict a sentiment intensity score for a word or a short phrase. Most participants adopted supervised learning techniques, such as the ECNU team [35] achievers of best results, where they presented a supervised learning-to-rank system to predict the strength associated with positive sentiment, and where they used the automatically labelled sentiment lexicon LabMT [4]. In 2018, SemEval declared a similar task, SemEval-2018 Task 1 (V-reg), where they evaluated automatic system ability to determine the intensity of sentiment (or valence) in a tweet. The participants chose supervised learning approaches too, based mainly on neural network architecture. For instance, the TCS Research team [18] used a deep neural network architecture to generate a robust representation of the text through parallel attention mechanism, on the top of word vector representation generated from pre-trained embeddings. The PlusEmo2Vec team [25] also employed neural network models but as feature extractors for traditional machine learning models (e.g., support vector regression, logistic regression). But the team with best results, the SeerNet team [5], employed a different supervised learning method, the *Random Forest Regression* (a meta-estimator – by combining the result of several predictions – which aggregates multiple decision trees), by using the Random Forest Regressor [26].

[1] http://alt.qcri.org/semeval2016/task7/
[2] https://competitions.codalab.org/competitions/17751

[3] Seed-words are words with strong semantic orientation both positive and negative, which are characterised by a lack of sensitivity to context [34]
[4] October 2018: http://jmcauley.ucsd.edu/data/amazon/

A recent breakthrough in the use of machine learning for Natural Language Processing (NLP) appeared with the generative pre-training of language representation models with context on a diverse corpus of unlabelled text, such as ELMo [28], BERT [3], OpenAI GPT [29], XLM [15]. Such techniques demonstrated large gains on a variety of NLP tasks including sentiment analysis classification [6, 21]. In particular, BERT (Bidirectional Encoder Representations from Transformers) [2, 3], a neural network architecture designed by Google researchers, proved to be one of the most powerful tools for text classification [6, 22, 24]. BERT model is a bidirectional transformer pre-trained based on BERT architecture over a large corpus consisting of the Toronto Book Corpus and Wikipedia. A number of recent works have shown that BERT based models work well for the Sentiment Intensity task. In [37], they propose a transfer learning based approach called SentiBERT, where the BERT model is coupled with a fully connected neural layer that then feeds into a final layer that predicts the score (using a mean-square error loss function). Using the Stanford Sentiment Treebank collection (SST-5) [32] as the training corpus, they showed that SentiBERT outperforms other neural models (Tree-LSTMs, GCN, RNNs) on a Twitter Collection. In [38], they developed an approach similar to SentiBERT, but trained on the collections provided by SemEval-2018 Task 1 organisers, and their approach outperformed existing baselines (such as SeerNet [5], SVM [20], PlusEmo2Vec [25], etc.). These results suggest that the BERT based regression model, SentiBERT, provides a state of the art supervised and transfer learning approach. However, it has not been extensively tested, and so we also provide a comprehensive evaluation of SentiBERT which serves an upper bound if supervised learning were to be employed – and to show how well our unsupervised approach works in comparison.

In summary, current lexicon-based sentiment analysis approaches do not require any prior training on large annotated datasets, in addition they are often faster to execute than machine learning approach, and can be used out-of-the-box. On the other hand, Lexicon-based methods do not take contextual-awareness into consideration, and ignore terms that are not within the vocabulary. Supervised sentiment analysis approaches demonstrate high accuracy in sentiment intensity, but they require a large annotated dataset which makes them dependant on the domain of that dataset and have limited transferability – for example, we show that a model trained on tweets generalize poorly to movie reviews. However, if trained on the same collection they offer good performance at the expense of complexity. Therefore, in this work, we describe in detail and evaluate our new framework [10] for scoring sentiment intensity that draws on the benefits from both approaches advantages – being unsupervised, scalable, contextually and semantically aware, and works reliable and robustly.

## 3 A WORD EMBEDDING SENTIMENT SCORING FRAMEWORK

Our presented framework, that we call *A Word Embedding Sentiment Scorer Of Many Emotions* (AWESSOME), is unsupervised and has the purpose of predicting the sentiment intensity of words, phrases, sentences, tweets, etc.. The generalised framework is inspired by the previous work of [12], relying on sentiment seed-words and word embedding, where the cosine similarity between the vector representation of two sentences is considered as a reflection of their sentiment similarity. For example, if we take the sentence $X$ (or a word $X$) and we calculate its similarity with the word (1)"*happy*", and then with the word (2)"*miserable*". If $X$ is more similar to (1) than to (2), that would increase $X$'s probability of being positive. To apply the proposed method, we need lists of words (seed-words) with strong semantic orientation (both positive and negative lists), that lack the sensitivity to context [19] (e.g., happy, sad, good, bad), to use as a reference of sentiment polarity and compare the sentences to them. Pre-existing *high quality* sentiment lexicons make a great choice as seed-words, and can be used whole or partially (e.g., VADER lexicon, LabMT lexicon). For the similarity calculation, [12] used a Word2Vec word embedding model which lack the ability to distinguish between the use of a same word in different contexts (e.g., *bank* as a riverside or as a financial institution). To solve that problem, we suggest a more general approach employing neural word embeddings through pre-trained language models (e.g., BERT, RoBERTa, etc.) which allows the extraction of the whole sentence embedding which can then be compared to the seed-words in order to preserve the semantics of the sentence. In addition, pre-trained language models make it possible for none-existing terms to be handled during the tokenization process, as presented in the following example: 'I am the walrus.' → ['I', 'am', 'the', 'wa', '##l', '##rus', '.'], which assure the embeddings generation of such terms. To predict the sentiment intensity of sentences using seed-words and neural word embedding, we present a choice of two kernel functions (average, max) and a weighting technique, presented in Figure 1:

- Difference between the average similarities of the sentence with positive Lexicon's terms and negative Lexicon's terms (AWESSOME(AVG,Lexicon,N)): In this method, the sentence's sentiment intensity score is the difference between the average of the similarity between the sentence (S) and each of the N most positive terms in the lexicon (positive seed-words) and the average of the similarity between the sentence and each of the N most negative terms in the lexicon (negative seed-words):

$$SIS(S) = [\frac{1}{|L_p|} \sum_{l_p \in L_p} sim(e(S), e(l_p)))] - [\frac{1}{|L_n|} \sum_{l_n \in L_n} sim(e(S), e(l_n)))] \quad (1)$$

  Where $|L_p|$ is the size of the positive seed-words list, $sim(e(S), e(l_p))$ is the similarity between the sentence $S$ and a positive term $l_p$ in the positive seed-words $L_p$, $|L_n|$ is the size of the negative seed-words list, and $sim(e(S), e(l_n))$ is the similarity between the sentence $S$ and a negative term $l_n$ in the negative seed-words $L_n$.
- Difference between the maximum similarity of the sentence with positive Lexicon's terms and negative Lexicon's terms (AWESSOME(MAX,Lexicon,N)): The sentence's sentiment intensity score is obtained by the difference between the maximum of the similarity between the sentence (S) and each of the N most positive terms in the lexicon (positive seed-words) and the maximum of the similarity between the sentence and each of the N most negative terms in the lexicon (negative seed-words):

$$SIS(S) = [\max_{l_p \in L_p} (sim(e(S), e(l_p)))] - [\max_{l_n \in L_n} (sim(e(S), e(l_n)))] \quad (2)$$

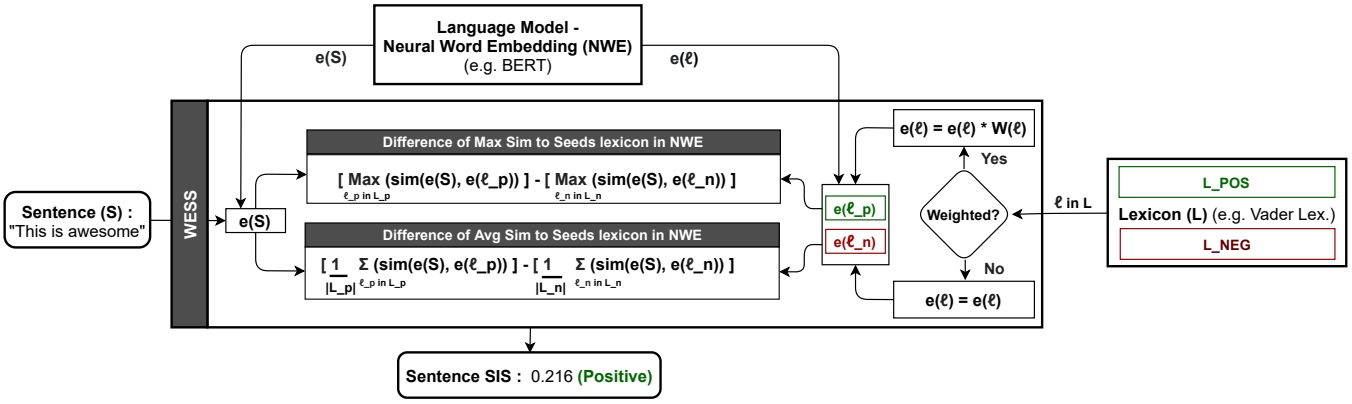


**Figure 1: The AWESSOME framework for sentiment intensity scoring consists of three components: a kernel function (Max, Avg), a neural word embedding (BERT, RoBERTa, etc.) and a seed lexicon (VADER, LabMT, etc.). An AWESSOME method is then defined by the selection of each component, e.g. AWESSOME(AVG, BERT, LabMT).**

- In addition, a weighting can be added to the previously listed methods by multiplying the value of similarity between the sentence and each term in the lexicon by the sentiment score of the term specified within the lexicon. The following equation present an example of applying weighting on the AWESSOME(AVG,Lexicon,N) method to be represented by AWESSOME(AVG-W,Lexicon,N), and calculated as follows:

$$SIS(S) = [\frac{1}{|L_p|}\sum_{l_p \in L_p} sim(e(S), e(l_p))) \cdot W(l_p))] - [\frac{1}{|L_n|}\sum_{l_n \in L_n} sim(e(S), e(l_n))) \cdot W(l_n))] \quad (3)$$

  Where $W(l)$ is the sentiment score of the term in the lexicon.

Put all together, we can define a specific AWESSOME instances by selecting the kernel function and if it is weighted or not (AVG,MAX, AVG-W,MAX-W), the seed lexicon (VADER, LabMT, etc.), and the number of seed words to use, e.g. *AWESSOME(AVG, VADER-Lex, 600)*. During our experiments we explore how the different variations perform in practice. The AWESSOME framework is written in Python and is downloadable and installable from GitHub [5].

## 4 RESEARCH QUESTIONS

Given the AWESSOME Framework, we wish to explore the following research questions:

- **RQ1**: How do different AWESSOME configurations (i.e. number of seed-words, choice of seed-words, kernel functions, etc.) affect performance ?
- **RQ2**: How does AWESSOME compare against Lexicon based approaches (i.e. VADER)?
- **RQ3**: How does AWESSOME compare against the current state of the art Supervised Deep Learning approaches (i.e. SentiBERT with and without Transfer Learning)?
- **RQ4**: Under what conditions does AWESSOME lead to increases or decreases in performance (i.e. vocab mismatch, negations, boosting, etc.)?

[5]https://github.com/cumulative-revelations/awessome

## 5 EXPERIMENTAL METHOD

The primary goal of our experiments is to evaluate our proposed sentiment intensity method, and its variant performance, against unsupervised and supervised approaches which serve as baselines and upperbounds. In addition, we tackle the impact of different variants on our suggested approach such as the employed lexicon, the number of seed-words, the use of an aggregation method. Furthermore, we address the influence of lexicon overlap between the test datasets and the employed lexicon on the results of the two main lexicon based method in this work: VADER and AWESSOME, and the influence of negation (e.g. not), boosters (e.g. very) and emojis (e.g.☺) within the test datasets on the results of the baseline (VADER), our suggested method (AWESSOME) and the upperbound (SentiBERT).

### 5.1 Test Collections

To test the performance of the methods, we used seven data collections (see Table 1 for an overview of the collection statistics). Each test collection is composed of a post containing text and a sentiment intensity score which was manually assigned (see Table 2 for an example of the posts). The first three collections were from the SemEval Evaluation forum (with sentiment intensity scores within [0,1]). The additional four datasets were provided and annotated by Gilbert and Hutto [7] (with sentiment intensity scores within [-4,4]). Also, scores for each of the collection were re-normalized to be between [-1,1]. The collections used were:

- **SE16-GE**: SemEval-2016 General English Sentiment Modifiers test collection containing 2999 phrases and short sentences [14].
- **SE16-MP**: SemEval-2016 English Twitter Mixed Polarity test collection containing 1269 phrases and short sentences [14].
- **SE18-Vreg**: SemEval-2018 Task1 (Valence regression) test collection of 937 tweets [20].
- **V-Amazon**: Amazon reviews snippets, includes 3708 sentence level snippets from 309 customer reviews on five different products [7].

**Table 1: Test Collection Statistics: the total number of items and terms, the percentage of overlap terms in common with the VADER and LabMT Lexicons, the percentage of records with Negation, with Boosters, and with Emojis.**

| Dataset | #records | #terms | w/ VADER-Lex | w/ LabMT-Lex | w/ Negation | w/ Boosters | w/ Emojis |
|---|---|---|---|---|---|---|---|
| **SE16-GE** | 2999 | 1394 | 45.0% | 58.7% | 10.6% | 18.5% | 0.0% |
| **SE16-MP** | 1269 | 796 | 36.8% | 86.0% | 3.0% | 5.0% | 0.0% |
| **SE18-Vreg** | 937 | 4467 | 15.3% | 48.9% | 18.7% | 18.4% | 19.2% |
| **V-Amazon** | 3708 | 5309 | 12.5% | 54.8% | 24.5% | 25.7% | 0.0% |
| **V-Movies** | 10605 | 18098 | 13.5% | 33.5% | 20.5% | 27.5% | 0.0% |
| **V-NYT** | 5190 | 13723 | 12.3% | 42.6% | 12.5% | 13.8% | 0.0% |
| **V-Tweets** | 4200 | 9310 | 12.6% | 50.9% | 12.2% | 14.8% | 0.0% |

- **V-Movies**: Movies reviews snippets, includes 10605 sentence-level snippets from *rotten.tomatoes.com* [7].
- **V-NYT**: New York Times editorial snippets, includes 5190 sentence-level snippets from 500 New York Times opinion editorials [7].
- **V-Tweets**: Tweets, includes 4200 tweets pulled from Twitter's public timeline [7].

**Table 2: Sentiment scores from the datasets' gold rankings.**

| Dataset | Content | Score |
|---|---|---|
| **SE16-GE** | favor | 0.826 |
| | increasingly difficult | 0.208 |
| **SE16-MP** | best winter break | 0.922 |
| | breaking | 0.250 |
| **SE18-Vreg** | It's my time to shiiine ☺ | 0.919 |
| | Shocking and sad! ☹ | 0.133 |
| **V-Amazon** | i would recommend buying one | 0.134 |
| | poor price / performance | -0.197 |
| **V-Movies** | Light, cute and forgettable | 0.100 |
| | Hilarious, acidic Brit comedy. | -0.109 |
| **V-NYT** | It is time for us to take action. | 0.072 |
| | The doors close. | -0.134 |
| **V-Tweets** | Hooray for new opportunities! | 0.120 |
| | bad mood. :'( | -0.216 |

It should be noted that the VADER lexicon was created using the V-Tweets collection — so its high performance on the V-Tweets collection is because of "overfitting". However, its performance on the other test collections shows how well it generalises. For transfer learning with the supervised approaches, we used an additional collection for training the regresssors. For this, we used the test collection from SemEval 2015 Task10 (1515 short phrases extracted from tweets) [30], the train collection from SemEval 2016 Task7 (400 short phrases extracted from tweets) [14] and the train collection from SemEval 2018 Task1(V-reg) (1630 tweets) [20]. The final training collection (SIS_train_collection) consisted of 3545 items with sentiment intensity annotations that were normalised between -1 and 1.

Furthermore, to test the influence of lexicon overlap between the test datasets and the employed lexicons, we divided each datasets into three sub-datasets: (1) the sentences with zero percent overlap with the lexicon, (2) the sentences with a percentage overlap with the lexicon lower than the median, (3) and the sentences with a percentage of overlap with the lexicon higher than the median. As well, to test the influence of negation, boosters and emojis, the test datasets were divided into four sub-datasets: (1) the sentences with Negation, (2) the sentences without Negation, (3) the sentences with Boosters, and (4) the sentences without Boosters. Also, the dataset S18-Vreg (the only dataset with emojis, check Table 1) is divided into two sub-datasets: sentences with emojis and sentences without emojis.

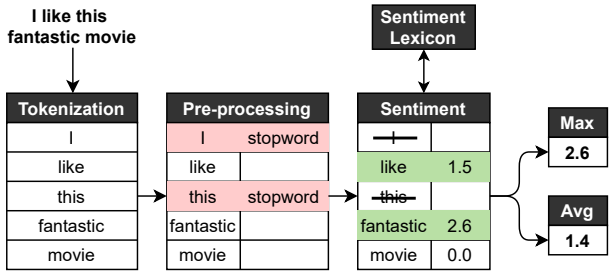


**Figure 2: The lexicon-based scoring methods of Lexicon_Avg and Lexicon_Max, where each sentence is tokenized, stop words removed, remaining words are ran against the sentiment lexicon to then select the maximum score or the average of terms scores.**

## 5.2 Baselines: Unsupervised approaches

To provide the baselines for this work we used two different lexicon-based sentiment analysis methods that have been shown to perform well along with VADER (the standard unsupervised approach):

- **Lexicon_Avg**: An aggregation of the sentence's terms sentiment scores by calculating their average (Figure 2): $SIS(S) = \frac{1}{|S|} \sum_{t \in S} SIS(t, Lexicon)$.
- **Lexicon_Max**: A selection of the sentence's maximum absolute score of the sentence's terms sentiment scores (Figure 2): $SIS(S) = \max_{t \in S} ||SIS(t, Lexicon)||$.
- **VADER**: The VADER method as described in [7].

For the unsupervised lexicon-based methods, two sentiment lexicons were employed: VADER lexicon [7] (7520 records, on a scale of -4 to 4) and LabMT lexicon, a rated lexicon of 10222 records on a scale of 1 (sad-negative) to 9 (happy-positive), which we converted its scores to the same scale as VADER (-4 to 4) by min-max normalization (e.g., $\frac{x-min}{max-min}$). Table 1 shows the number of unique terms (column *#terms*), excluding stopwords, in the seven used datasets

(SE16-GE, SE16-MP, SE18-Vreg, V-Amazon, V-Movie, V-NYT, and V-Tweets), in addition to the percentage of these terms in each of the sentiment lexicons (columns *w/VADER-Lex* and *w/LabMT-Lex*).

## 5.3 Upperbounds: Supervised Approaches

While our goal is to develop an unsupervised method for sentiment intensity prediction, we feel it is important to compare to the supervised state of the art approaches to contextualise our methods performance. Moreover, the inclusion of these comparisons also provide a more comprehension analysis of these approaches for this task (as most prior work has focused on sentiment classification, and not intensity prediction).

The first supervised method we employed was a Support Vector Regression (SVR) [31] previously used, and shown to provide a strong baseline performance [13, 23]. We employed the pre-trained model in spaCy [9], *en_core_web_md* library which includes 20k unique vectors with 300 dimensions, and we created the word embeddings used as input features for SVR.

The second supervised method we employed was the BERT-based method: SentiBERT [37, 38] which has been shown to give state of the art performance on one of the test collections (SE18-Vreg). SentiBERT is built on the HuggingFace[6] library, and the model parameters are initialized using pre-trained BERT-base model. The fine-tuning of SentiBERT is as follows. (1) The pre-training language models have a maximum input length of 512 tokens, but in our work, we defined the input sentences size to 128 tokens since most posts in our test collections were below 128 tokens in length. (2) The loss function was set to Mean Square Error and the output was the score. (3) The train batch size is set to 32. (4) The number of training epoch is set to 10.

- **Supervised (SentiBERT-S)**: We used 5-fold cross validation, where 20% of the test collection was held out, while the fine tuning was performed. The performance on each collection was reported given the 5 held-out folds.
- **Transfer Learning (SentiBERT-TL)**: Rather than training on the collection, we employed transfer learning, where we used the training collection we created (SIS_train_collection) which was based on completely unseen data.

## 5.4 Evaluation Measures

The proposed methods were evaluated by their abilities to correctly rank the sentences compared to their position in the gold rankings. To evaluate the performance of the sentiment intensity scoring methods, we use the official SemEval Workshops measures used in each track, for: Datasets SE16-GE and SE16-MP, Kendall's rank correlation coefficient is used (Kendall's $\tau$), and for Dataset SE18-Vreg, Pearson correlation coefficient is used (Pearson's $r$). As for Gilbert and Hutto [7] test collections, Pearson correlation coefficient is used (Pearson's $r$). To determine whether the correlations/predictions are significantly different between runs we performed significance testing – where we consider a p-value of less than 0.05 as statistically significant

## 5.5 Results and Analysis

## 5.6 RQ1: AWESSOME's variants implication

The proposed method AWESSOME, of combining lexicons with pre-trained language models, is applied by obtaining the similarity between the sentences of the datasets (SE16-GE, SE16-MP, SE18-Vreg, V_Amazon, V_Movies, V_NYT and V_Tweets) and the lexicons' terms, of VADER and LabMT lexicons. More specifically, the method requires obtaining the similarity between the sentences of the datasets and a selected N highly positive and N highly negative terms of the lexicons, which we call seed-words. Therefore, we needed to extract a certain number of words from the lexicons as seed-word lists. As presented in Figure 3, we test the effect of the seed-words' size (between 5 and 1000), the lexicons used (VADER and LabMT), and the kernel functions employed (Average, Max, in addition to the weighting option) on the sentiment intensity prediction performance. And as shown in Figure 3, the methods AWESSOME(AVG,VADER-lex,N) and AWESSOME(AVG-W,VADER-lex,N) were able to achieve best results on all test collections, with a stability in performance toward the variant seed-words number, and a weak statistical significance after seed-words number higher than 600. Note that the original Vader lexicons of 7520 terms has a majority of abbreviations and emoticons, therefore, to extract the seed-words, we reduced the lexicon to 1244 terms by extracting only English words from the lexicon and also by using Stemmer[7] to get only the root words from the lexicon. From that reduced lexicon list, the N highly positive and N highly negative terms are then selected as seed-words lists.

## 5.7 RQ2: AWESSOME vs. Lexicon Approaches

The different lexicon-based sentiment analysis approaches, applied with the different sentiment lexicons, are presented in the first two sections of Table 3 as follows

- First, we applied the *Lexicon_Avg* and the *Lexicon_Max* methods on both lexicons: VADER and LabMT.
- Second, we applied *VADER rule-based* approach on the original VADER lexicon, then on LabMT lexicon, and on a combined lexicon of VADER and LabMT (16363 records).

In the last section of Table 3 (AWESSOME), we present the best results of our suggested method: AWESSOME(AVG,VADER-lex,600), AWESSOME(AVG,Lab-MT-lex,600), AWESSOME(AVG-W,VADER-lex,600), and AWESSOME(AVG-W,Lab-MT-lex,600). Our suggested method achieved better results than the unsupervised baselines, particularly when employing VADER lexicon. This positive influence of VADER lexicon could be caused by its smaller size (after removing the emoticons) and the nature of the strong semantic orientation of its words (positive and negative) what makes them better seed-words. The lexicon-based approaches baseline were only able to perform better than AWESSOME with the V-Tweets collection, using VADER's lexicon, what can be due to the fact that VADER lexicon was created by means of the V-Tweets collection [7]. The results in Table 3 also indicated a good performance of the Lexicon approach where it exceeded VADER's result in predicting the sentiment intensity of datasets SE18-Vreg and V-Amazon. Those datasets differ from the others datasets of being built from full

[6] https://github.com/huggingface

[7] https://www.nltk.org/howto/stem.html

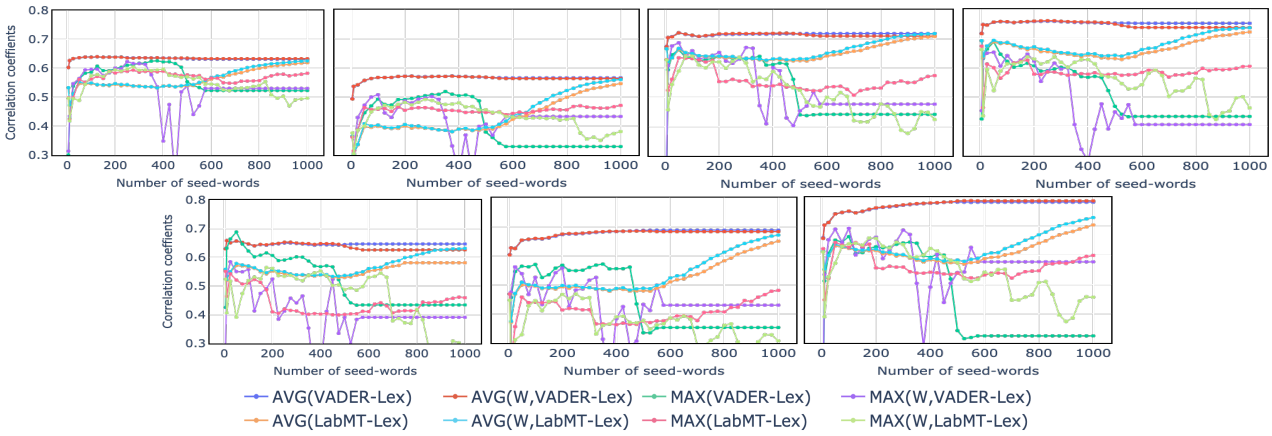


**Figure 3: The correlation coefficients (y-axis) for each of the AWESSOME methods as the number of seed-words from Vader lexicons and LabMT lexicon varies between 5 and 1000 words (x-axis), applied on seven datasets in the following order: SE16-GE, SE16-MP, SE18-Vreg, V-Amazon, V-Movies, V-NYT and V-Tweets.**

tweets. Such results uncover certain weak points within VADER's rules with a possible difficulty to process full and complex tweets other than the ones it was built on (V-Tweets).

Apart from that, in VADER's results section, the combination of both lexicons did not improve VADER's performance but instead it led to a result in between each of the lexicon's employment results. On the other hand, in an additional experiment, we applied an aggregation of linear combination between the results of lexicon-based methods and our method (AWESSOME(AVG-M,Lexicon,600)) using VADER and LabMT lexicons. The following equation was employed for the results aggregation: $SIS(S) = \lambda \cdot SIS(S, LabMT) + (1 - \lambda) \cdot SIS(S, VADER)$, where best results were achieved with $\lambda$ equals to 0.5, after an iteration by 0.1 of $\lambda$ in the interval of [0,1]. Table 4 presents the aggregation results, with $\lambda$ equals to 0.5, where an improvement of results is detected for all the suggested methods on most test collections.

## 5.8 RQ3: AWESSOME vs. Superv. Approaches

In Table 5, we present the results of predicting sentences' sentiment intensity using supervised approaches. The first row of that table display the results by AWESSOME(AVG-W,VADER-Lex,600), exported from Table 3 for comparison purposes. In the second raw we present the results achieved by SVR, where promising scores were only reached with the collections of full tweets (SE18-Vreg and V-Tweets). The selective high performance of SVR could be caused by spaCy's pre-trained model which we used for word embeddings generation – the pre-trained model is trained over a collection of: telephone conversations, news wire, news groups, broadcast news, broadcast conversation, weblogs and religious texts [8]. Therefore, being trained over a majority of informal language text, the model would logically perform better on tweets of similar informal nature. The third raw of Table 5 indicates an extremely high efficiency of SentiBERT-S, trained on a dataset of same type and nature of the test dataset, especially for the datasets SE16-GE, SE16-MP, V-Amazon, V-Movies, V-Tweets. The fourth row presents the results of SentiBERT-TL, and it highlighted SentiBERT-TL's competence in predicting the sentiment intensity, with all collections except V-Tweets, where VADER continues to have better results than SentiBERT-TL. Those results marked again the over-fitting of the V-Tweets collection to the VADER's lexicon, since the best set of results is often connected to VADER or the use of VADER's lexicon.

[8] https://spacy.io/models/en

## 5.9 RQ4: AWESSOME on different conditions

In this subsection, we drill down on comparing: VADER (as a baseline), our method AWESSOME(AVG-W,VADER-Lex,600), and SentiBERT-TL (as an upperbound) to see how differences in the test collections or in the lexicons would effect the performance of the methods.

First, we tested the influence of lexicon overlap between the test collections and the employed lexicons by the lexicon dependant methods: VADER and AWESSOME(AVG-W,VADER-Lex,600). As shown in Table 6, the performance of VADER, a total lexicon-based model, is highly dependant on the percentage of overlapped terms with the lexicons (Table 1), but our proposed method is less sensitive to such impact. Then, we examined the negation, boosters and emojis effect on the performance of VADER, AWESSOME(AVG-W,VADER-Lex,600) and SentiBERT-TL. And as reported in Table 7, the presence of negation and emojis make it more challenging for all methods to predict correctly the sentiment intensity, but we can also see that AWESSOME and SentiBERT are much better in handling such impact than VADER. As for boosters (e.g. very), their presence improves all methods performance.

Furthermore, for a clearer understanding of the results, we created result visualisations, presented in Figure 4, where the distribution of sentiment intensity prediction by SentiBERT-TL (first row), AWESSOME(AVG-W,VADER-Lex,600) (second row), and VADER (third row) are displayed in regards to the gold results (y-axis), or correct results, in the collections SE16-GE, SE16-MP, SE18-Vreg, V_Amazon, V_Movies, V_NYT, and V_Tweets (from left to right).

**Table 3: Experiments results of Unsupervised methods, with * denote significant differences with Lexicon(AVG(VADER-Lexicon)), † denote significant differences with VADER(VADER-Lexicon), and ‡ denote significant differences with AWESSOME(Avg(W,VADER-Lex-600)), with p-value < 0.01.**

| | Method | SE16-GE | SE16-MP | SE18-Vreg | V-Amazon | V-Movies | V-NYT | V-Tweets |
|---|---|---|---|---|---|---|---|---|
| **Lexicon** | **AVG(VADER-Lexicon)** | 0.468 | 0.342 | 0.674 | 0.535 | 0.390 | 0.468 | 0.791 |
| | AVG(LabMT-Lexicon) | 0.447 | 0.381* | 0.604 | 0.447 | 0.341 | 0.393 | 0.704 |
| | MAX(VADER-Lexicon) | 0.467 | 0.341 | 0.580 | 0.511 | 0.346 | 0.433 | 0.784 |
| | MAX(LabMT-Lexicon) | 0.441 | 0.385* | 0.562 | 0.420 | 0.249 | 0.366 | 0.695 |
| **VADER** | **VADER-Lexicon** | 0.586* | 0.365* | 0.517 | 0.570* | 0.419* | 0.489* | **0.878***† |
| | LabMT-Lexicon | 0.566 | 0.432† | 0.534† | 0.372 | 0.284 | 0.369 | 0.656 |
| | VADER+LabMT-Lexicon | 0.572 | 0.432† | 0.530† | 0.388 | 0.333 | 0.489 | 0.697 |
| **AWESSOME** | **(AVG-W,VADER-Lex,600)** | **0.636***† | **0.570***† | **0.718***† | **0.759***† | **0.650***† | 0.680 | 0.773 |
| | (AVG-W,LabMT-Lex,600) | 0.574 | 0.480 | 0.633 | 0.652 | 0.539 | 0.491 | 0.602 |
| | (AVG,VADER-Lex,600) | 0.631 | 0.563 | **0.718** | 0.737 | 0.647 | **0.688***†‡ | 0.783‡ |
| | (AVG,LabMT-Lex,600) | 0.587 | 0.479 | 0.642 | 0.648 | 0.552 | 0.511 | 0.589 |

**Table 4: Linear aggregation results achieved by VADER and LabMT lexicons, with $\lambda$=0.5, and where * denote significant differences with the results prior the linear combination, with a p-value < 0.05.**

| Method | SE16-GE | SE16-MP | SE18-Vreg | V-Amazon | V-Movies | V-NYT | V-Tweets |
|---|---|---|---|---|---|---|---|
| AVG(VADER-Lex) | | | | | | | |
| +AVG(LabMT-Lex) | 0.487* | 0.413* | 0.695* | 0.552* | 0.416* | 0.499* | 0.809* |
| VADER(VADER-Lex) | | | | | | | |
| +VADER(LabMT-Lex) | 0.479* | 0.407* | 0.622* | 0.546 | 0.405 | 0.485 | 0.835 |
| AWESSOME(AVG-W,VADER-Lex,600)) | | | | | | | |
| +AWESSOME(AVG-W,LabMT-Lex,600)) | **0.639*** | **0.580*** | **0.722*** | **0.767*** | **0.652*** | 0.485 | 0.752 |

**Table 5: Experiments results of Supervised methods, with * denote significant differences with AWESSOME(AVG-W,VADER-Lex,600)) exported from Table 3, with p-value < 0.01.**

| Method | SE16-GE | SE16-MP | SE18-Vreg | V-Amazon | V-Movies | V-NYT | V-Tweets |
|---|---|---|---|---|---|---|---|
| AWESSOME(AVG-W,VADER-Lex,600)) | 0.636 | 0.570 | 0.718 | 0.759 | 0.650 | 0.680 | 0.773 |
| SVR | 0.270 | 0.405 | 0.709 | 0.475 | 0.457 | 0.421 | 0.751 |
| SentiBERT (Train/Test same collection) | **0.943*** | **0.999*** | 0.779* | **0.765*** | **0.945*** | 0.675 | **0.977*** |
| SentiBERT (Transfer Learning) | 0.717* | 0.638* | **0.835*** | 0.749 | 0.672* | **0.705*** | 0.797* |

**Table 6: Comparing the efficiency of VADER vs AWESSOME (Lexicon dependant methods) in different sub-datasets divided based on the overlapped of terms with Vader and LabMT lexicons.**

| | Section | S16-GE | | S16-MP | | S18-Vreg | | V-Amazon | | V-Movie | | V-NYT | | V-tweet | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | VADER | AWS | VADER | AWS | VADER | AWS | VADER | AWS | VADER | AWS | VADER | AWS | VADER | AWS |
| **Vader** | Zero | 0.083 | **0.600** | 0.056 | **0.403** | 0.360 | **0.670** | 0.119 | **0.599** | 0.077 | **0.564** | 0.081 | **0.583** | **0.804** | 0.371 |
| | > Median | **0.628** | 0.611 | 0.415 | **0.583** | 0.505 | **0.701** | 0.579 | **0.722** | 0.367 | **0.596** | 0.448 | **0.658** | **0.852** | 0.727 |
| | < Median | **0.676** | 0.624 | 0.441 | **0.608** | 0.603 | **0.730** | 0.629 | **0.798** | 0.461 | **0.663** | 0.570 | **0.734** | **0.897** | 0.832 |
| **LabMT** | Zero | 0.013 | **0.530** | 0.013 | **0.413** | 0.139 | **0.481** | 0.124 | **0.302** | 0.001 | **0.739** | 0.210 | **0.463** | 0.022 | **0.040** |
| | > Median | 0.437 | **0.575** | 0.402 | **0.426** | 0.413 | **0.615** | 0.248 | **0.634** | 0.190 | **0.527** | 0.257 | **0.500** | **0.625** | 0.545 |
| | < Median | **0.661** | 0.558 | - | - | 0.468 | **0.686** | 0.256 | **0.669** | 0.211 | **0.552** | 0.258 | **0.533** | **0.696** | 0.659 |

The visualisations indicated VADER's model tendency to over classify sentences as positive in the SemEval's collections (SE16-GE, SE16-MP, and SE18-Vreg). That behaviour is viewed clearer in the case of SE18-Vreg, where VADER falsely classified most sentences as positive (close to 1, on the far right). On the other hand, our method AWESSOME, maintained a balanced prediction in regards to the gold results, mostly for the SemEval's collections, something also seen in the SentiBERT graphs over all the test collections.

Finally, a comparison between the methods Lexicon_Avg, VADER, AWESSOME(AVG-W,VADER-lex,600) and SentiBERT-TL, in their ability to correctly predict the sentiment intensity in complex sentences, is presented with few examples in Table 8. All methods were able to predict correctly the sentiment intensity in a simple sentence and its negation (*I am happy*, *I am not happy*), but they were not able to detect a complex negation and considered the sentence "*I am the opposite of happy*" as a positive sentence, except

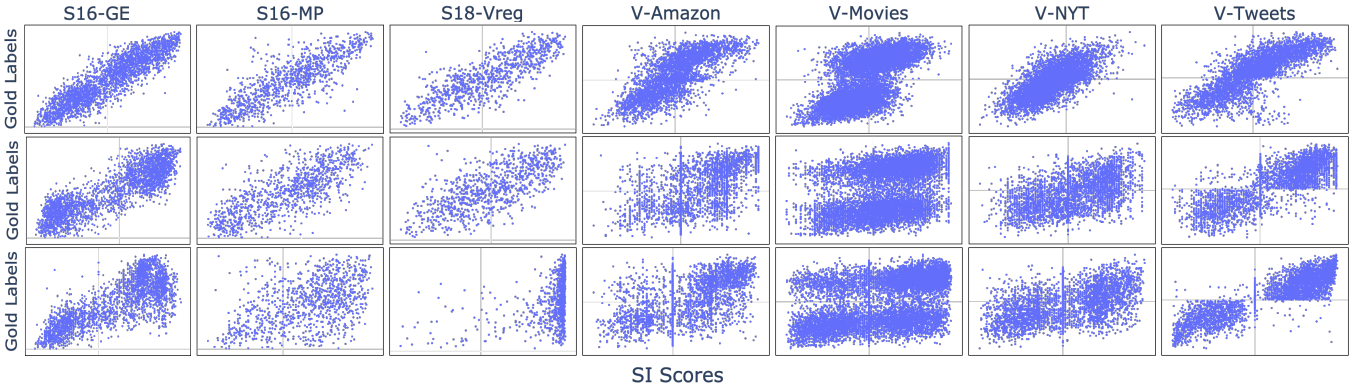


**Figure 4: The distribution of the SentiBERT-TL results (first row), AWESSOME(AVG-W,VADER-lex,600) results (second row), and VADER model's results (third row) in comparison to the gold results in the tested datasets: SE16-GE, SE16-MP, SE18-Vreg, V_Amazon, V_Movies, V_NYT and V_Tweets. Gold Label (y-axis of interval [0,1]) and SIS scores (x-axis of interval [-1,1]).**

**Table 7: Performance of VADER, AWESSOME(AVG-W,VADER-lex,600) and SentiBERT-TL in different sub-datasets divided based on Negation, Boosters and Emojis occurrence.**

| | Section | VADER | AWESSOME | SentiBERT |
|---|---|---|---|---|
| **SE16-GE** | w/ Neg | 0.431 | 0.448 | **0.454** |
| | w/o Neg | 0.587 | 0.642 | **0.725** |
| | w/ Boost | 0.659 | 0.628 | **0.712** |
| | w/o Boost | 0.543 | 0.611 | **0.703** |
| **SE16-MP** | w/ Neg | 0.118 | 0.502 | **0.562** |
| | w/o Neg | 0.382 | 0.562 | **0.641** |
| | w/ Boost | 0.600 | 0.666 | **0.742** |
| | w/o Boost | 0.360 | 0.561 | **0.626** |
| **SE18-Vreg** | w/ Neg | 0.362 | 0.604 | **0.675** |
| | w/o Neg | 0.598 | 0.729 | **0.801** |
| | w/ Boost | 0.628 | 0.779 | **0.837** |
| | w/o Boost | 0.542 | 0.703 | **0.756** |
| | w/ Emoji | 0.670 | 0.708 | **0.735** |
| | w/o Emoji | 0.513 | 0.717 | **0.771** |
| **V-Amazon** | w/ Neg | 0.439 | **0.703** | 0.676 |
| | w/o Neg | 0.597 | **0.751** | 0.640 |
| | w/ Boost | 0.540 | **0.770** | 0.746 |
| | w/o Boost | 0.583 | **0.744** | 0.721 |
| **V-Movies** | w/ Neg | 0.330 | 0.606 | **0.610** |
| | w/o Neg | 0.439 | 0.651 | **0.654** |
| | w/ Boost | 0.423 | 0.653 | **0.672** |
| | w/o Boost | 0.420 | 0.645 | **0.661** |
| **V-NYT** | w/ Neg | 0.349 | **0.558** | 0.525 |
| | w/o Neg | 0.508 | 0.697 | **0.720** |
| | w/ Boost | 0.488 | 0.675 | **0.680** |
| | w/o Boost | 0.490 | 0.691 | **0.703** |
| **V-Tweets** | w/ Neg | .831 | **0.703** | 0.685 |
| | w/o Neg | 0.884 | 0.789 | **0.780** |
| | w/ Boost | 0.882 | 0.848 | **0.825** |
| | w/o Boost | 0.878 | 0.768 | **0.783** |

for AWESSOME, it correctly classified it as negative (-0.308). Also, VADER and Lexicon_Avg were not able to predict the sentiment of a sentence with a slang word (*happytastic* = *happy* + *fantastic*) and classified it as neutral (0.0), but AWESSOME and BERT classified it correctly as positive, since they both employ pre-trained language models what makes it possible for none-existing terms to be handled during the tokenization process.

**Table 8: Comparison between Lexicon_Avg, VADER, AWESSOME(AVG-W,VADER-lex,600) and SentiBERT-TL (SIS scores normalised between [-1,1], with -1 very negative, 1 very positive.**

| Sentence | Lexicon_Avg | VADER | AWS | SentiBERT |
|---|---|---|---|---|
| I am happy | 0.900 | 0.572 | 0.412 | 0.999 |
| I am not happy | 0.900 | -0.458 | -0.350 | -0.998 |
| I am opposite of happy | 0.900 | 0.572 | -0.308 | 0.826 |
| I am happytastic | 0.000 | 0.000 | 0.388 | 0.998 |

## 6 SUMMARY AND CONCLUSION

In this paper, we presented in detail and evaluated a configurable framework under the name of AWESSOME, for sentiment intensity scoring. The AWESSOME framework combines together a seed lexicon, a neural word embedding, and a score function. In our evaluation, seven sentiment test collections were used to evaluate our approach, comparing it against typically used lexicon based approaches, and comparing it against state of the art supervised methods. Our framework outperformed existing lexicon approaches but it did not surpass supervised approaches. Clearly, the supervised SentiBERT approach provides greater accuracy when properly trained. However, as we have shown AWESSOME provides a simple and effective approach over other unsupervised approached by addressing the short-comings inherent in lexicon based approaches. In addition, the AWESSOME framework provides the flexibility to cater for different seed lexicons and different neural word embeddings models to further tailor the scoring more specifically to the corpus, task or domain.

## ACKNOWLEDGMENTS

*Cumulative Revelations of Personal Data* This project is supported by the UKRI's EPSRC under Grant Number: EP/R033854/1.

# REFERENCES


[1] Yejin Choi and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. 793–801.

[2] Jacob Devlin and Ming-Wei Chang. 2018. Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing. *Google AI Blog, November* 2 (2018).

[3] 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* (2018).

[4] Peter Sheridan Dodds, Kameron Decker Harris, Isabel M Kloumann, Catherine A Bliss, and Christopher M Danforth. 2011. Temporal patterns of happiness and information in a global social network: Hedonometrics and Twitter. *PloS one* 6, 12 (2011), e26752.

[5] Venkatesh Duppada, Royal Jain, and Sushant Hiray. 2018. SeerNet at SemEval-2018 Task 1: Domain Adaptation for Affect in Tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*. ACL, 18–23.

[6] Zhengjie Gao, Ao Feng, Xinyu Song, and Xi Wu. 2019. Target-dependent sentiment classification with BERT. *IEEE Access* 7 (2019), 154290–154299.

[7] CHE Gilbert and Erric Hutto. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf*, Vol. 81. 82.

[8] Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford* 1, 12 (2009), 2009.

[9] Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. (2017). To appear.

[10] Amal Htait and Leif Azzopardi. 2021. AWESSOME: An unsupervised sentiment intensity scoring framework using neural word embeddings. (2021).

[11] Amal Htait, Sebastien Fournier, and Patrice Bellot. 2016. LSIS at SemEval-2016 Task 7: Using web search engines for English and Arabic unsupervised sentiment intensity prediction.

[12] Amal Htait, Sébastien Fournier, Patrice Bellot, Leif Azzopardi, and Gabriella Pasi. 2020. Using sentiment analysis for pseudo-relevance feedback in social book search. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*. 29–32.

[13] Siddharth Jain and Sushobhan Nayak. 2012. Sentiment analysis of movie reviews: A study of features and classifiers. *CS221 Course Project: Artificial Intelligence, Stanford* (2012).

[14] Svetlana Kiritchenko, Saif Mohammad, and Mohammad Salameh. 2016. Semeval-2016 task 7: Determining sentiment intensity of english and arabic phrases. In *Proceedings of the 10th international workshop on semantic evaluation (SEMEVAL-2016)*. 42–51.

[15] Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291* (2019).

[16] Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5, 1 (2012), 1–167.

[17] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*. 43–52.

[18] Hardik Meisheri and Lipika Dey. 2018. TCS research at SemEval-2018 Task 1: Learning robust representations using multi-attention architecture. In *Proceedings of The 12th International Workshop on Semantic Evaluation*. 291–299.

[19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *ICLR Workshop* (2013).

[20] Saif M. Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 Task 1: Affect in Tweets. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*. New Orleans, LA, USA.

[21] Manish Munikar, Sushil Shakya, and Aakash Shrestha. 2019. Fine-grained sentiment classification using bert. In *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, Vol. 1. IEEE, 1–5.

[22] Alex Nikolov and Victor Radivchev. 2019. Nikolov-Radivchev at SemEval-2019 Task 6: Offensive Tweet Classification with BERT and Ensembles. In *Proceedings of the 13th International Workshop on Semantic Evaluation*. 691–695.

[23] Itir Onal, Ali Mert Ertugrul, and Ruket Çakıcı. 2014. Effect of using regression on class confidence scores in sentiment analysis of twitter data. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. 136–141.

[24] Pulkit Parikh, Harika Abburi, Pinkesh Badjatiya, Radhika Krishnan, Niyati Chhaya, Manish Gupta, and Vasudeva Varma. 2019. Multi-label Categorization of Accounts of Sexism using a Neural Framework. *arXiv preprint arXiv:1910.04602* (2019).

[25] Ji Ho Park, Peng Xu, and Pascale Fung. 2018. Plusemo2vec at semeval-2018 task 1: Exploiting emotion knowledge from emoji and# hashtags. *arXiv preprint arXiv:1804.08280* (2018).

[26] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.

[27] James W Pennebaker, Martha E Francis, and Roger J Booth. 2001. Linguistic inquiry and word count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates* 71, 2001 (2001), 2001.

[28] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* (2018).

[29] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf* (2018).

[30] Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. SemEval-2015 Task 10: Sentiment Analysis in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*.

[31] Alexander J Smola et al. 1996. *Regression estimation with support vector learning machines*. Ph.D. Dissertation. Master's thesis, Technische Universität München.

[32] Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*. 1201–1211.

[33] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics* 37, 2 (2011), 267–307.

[34] Peter D Turney and Michael L Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *TOIS* 21, 4 (2003), 315–346.

[35] Feixiang Wang, Zhihua Zhang, and Man Lan. 2016. Ecnu at semeval-2016 task 7: An enhanced supervised learning method for lexicon sentiment intensity ranking. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. 491–496.

[36] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*. 606–615.

[37] Da Yin, Tao Meng, and Kai-Wei Chang. 2020. Sentibert: A transferable transformer-based architecture for compositional sentiment semantics. *arXiv preprint arXiv:2005.04114* (2020).

[38] Linrui Zhang, Hsin-Lun Huang, Yang Yu, and Dan Moldovan. 2020. Affect inTweets: A Transfer Learning Approach. In *Proceedings of The 12th Language Resources and Evaluation Conference*. 1511–1516.