Detect Extreme Sentiments on Social Networks using BERT

Muhammad Luqman Jamil[1†], Sebastião Pais[1,2*†], João Cordeiro[1,3†] and Gaël Dias[4†]

[1*]Department of Computer Science, University of Beira Interior, Covilhã, Portugal.
[2]NOVA LINCS, New University of Lisboa, Lisboa, Portugal.
[3]LIAAD, INESC TEC, Porto, Portugal.
[4]GREYC, University of Caen Normandie, Caen, France.

*Corresponding author(s). E-mail(s): sebastiao@di.ubi.pt;
Contributing authors: luqman.jamil@ubi.pt; jpcc@ubi.pt;
gael.dias@unicaen.fr;
†These authors contributed equally to this work.

Abstract

Online social networking platforms allow people to freely express their ideas, opinions, and emotions negatively or positively. Previous studies have examined user's sentiments on these platforms to study their behaviour in different contexts and purposes. The mechanism of collecting public opinion information has attracted researchers to automatically classify the polarity of public opinions based on the use of concise language in messages, such as tweets, by analyzing social media data. In this paper, we extend the preceding work [1], by proposing an unsupervised approach to automatically detect extreme opinions/posts in social networks. We have evaluated our performance on five different social network and media datasets. In this work, we use the semi-supervised approach BERT to check the accuracy of our classified dataset. The latter task shows that, in these datasets, posts that were previously classified as negative or positive are, in fact, extremely negative or positive in many cases.

Keywords: BERT, Sentiment Analysis, Extreme Sentiment Analysis, Violent Extremism, Social Media, Social Networks

1 Introduction

Online social networks, such as Facebook, Twitter, Tumblr, and YouTube, have become a de-facto platform for hundreds of millions of Internet users to establish and maintain interpersonal relationships. In recent years, the emergence of microblogging services has greatly influenced the way people think, communicate, behave, learn and conduct business. These popular social platforms, such as Twitter or Tumblr, are new forms of blogging that facilitate communication between people. By writing posts, sharing articles, videos, links or tweeting messages, people can make their own opinions, ideas and thoughts, in a constructive or destructive tone [2].

Conversely, any collection of tweets or posts that focus on a hot topic can pose a potential threat to society and individuals. The vast majority of the information published on social networks is harmless. It represents casual, conventional or expressive crowds, as well as noisy data [3]. Researchers and policymakers are still trying to discover the rise of violent extremism among people and take appropriate measures to prevent it. For example, the work of [4] shows that the use of specifically radicalised language within crowds acting and protesting on social media can increase violent extremism. In addition, the use of social media by a terrorist organisation to study human sentiments by accessing uncensored content to collect information of public views, by monitoring data from social networks and automatically to classify the polarity of public sentiments due to the use of concise language in posts or/and tweets, [5], allows violent extremists to increase recruitment by being able to establish personal relationships with a global audience to their advantage. [5].

An unusual form of sentiment analysis is the detection and classification of extreme sentiments, which are the most negative and positive sentiments about a particular subject, object, or person [6]. In a more general form, an extreme sentiment can be seen described as the worst or best opinion, judgment, or evaluation formed in one's mind about a particular thing or person. However, this paper considers an extreme sentiment only as a personal extreme positive or extreme negative sentiment.

In the work we have recently developed, described in this paper, we have applied deep learning [7] models to detect extreme sentiments in text coming from social platforms. In particular, we propose to use pre-trained BERT [8] to improve the efficacy of the detection. BERT is based on a Semi-Supervised approach, pre-trained trained on a massive dataset. Thus, in addition, we must use *Transfer Learning* [9] which is typically done for tasks where the dataset contains too little data to train a full-scale model from scratch. The model is then fine-tuned, where we unpack the basic model and re-train it on the new data with a very low learning rate. This can potentially lead to significant improvements by gradually adapting pre-trained ~~trained~~ features to the new data.

The document is structured as follows: Section 2 discusses related work and in Section 3 we focus on BERT. Section 4 presents the experimental setup.

Section 5 presents the results and analysis. And finally in Section 6 we present conclusions and future directions of this work.

2 Related Work

2.1 Detection and classification of extremist affiliations in social media.

The authors of [10] proposed a binary classification task to detect extremist affiliation. The work focuses on ML classifiers, i.e., Random Forest, Support Vector Machine, K Nearest Neighbours (KNN), Naive Bayes and Deep Learning. The authors apply a sentiment-based extremist classification technique to user tweets, which works in three modules: (i) collection of user tweets, (ii) preprocessing and (iii) classification into extremist and non-extremist classes using various Deep Learning-based sentiment models, namely *Long Short Term Memory*, *Convolutional Neural Networks*, *FastText* and *Gated Recurrent Units (GRU)*.

The work of Kaur et al. [11] uses deep learning approach for automatic detection of extremism. The data collected by the authors are divided into radical, non-radical, and irrelevant through the use of relevant annotators. Word2Vec is used to generate word embeddings from data. Researchers make use of LSTM to detect extremism and classify the data as radical, not radical and irrelevant in the context of India. The authors use specialized annotators to label the data. Labeling text is based on the characteristics specified by the authors like *abuse of Indian military personnel*, *anti-national discourse*, *endorsing terrorism*, and *terrorists and inciting others*. The authors use different ML algorithms such as Random forest, Support Vector machine (SVM) and Max Entropy.

The authors of the work [12] detect hatred and right-wing extremism in German Twitter users. The authors have identified several dehumanizing catchphrases used by right-wing extremists. The study classified and collected tweets as hate or non-hate for automatic detection. For the purpose of training the model, the authors used tweets in German and English. The study uses character trigrams as a method for feature extraction. Various features such as emojis, unigrams, bigrams, punctuation marks, etc, are also taken into account. The authors also tested their models in various unknown samples, some of which marked manually by experts. These unknown samples were collected from various sources such as some random articles, Wikipedia pages in German, and German far-right conspiracy websites.

2.2 Sentiment Analysis Tools

Sentiment Analysis (SA) is a sub-field of *Natural Language Processing* mainly concerned with an effective way to determine the polarity of a text, i.e., the prediction of whether the opinion expressed in a text is positive, negative or neutral. These analyses give a powerful tool for deriving insights

from large amounts of opinion-based data, such as social media posts and product reviews. SA is a proficient area for researchers, especially in the context of social media activity. In general, access control systems fall into two categories: knowledge-based systems and statistics-based systems. Earlier knowledge-based approaches were the most popular among researchers to identify sentiment polarity in texts. However, researchers are increasingly using statistically-based approaches with an emphasis on supervised statistical methods [13].

Wagh et al. [14] designed a general sentiment classification to analyze whether a data label is available or unavailable in the target domain. The study analyzed the public dataset of four million tweets from Stanford University to predict the sentiment polarity in user opinions. SA using Hadoop, which rapidly runs large datasets on a real-time Hadoop cluster, was presented by Mane et al. [15]. It is a platform designed to solve large, unstructured, and complex big data problems using the divide-and-conquer approach to data processing. The study used a number-based approach to scale statements into several classes that assigned an appropriate range of different sentiments. *SENTA* [16] is an SA tool that offers many features to the end-user. The authors collect texts from Twitter and use SENTA to perform multiclass SA on the texts. Most of these approaches are supervised methods; our research focuses on an unsupervised and language-independent methodology to detect extreme sentiments on social media platforms.

2.3 Sentiment based Lexicons

SentiWordNet 3.0 was developed by automatically scoring all *WordNet* synsets with the terms "positivity", "negativity", and "neutrality". Each synset has three numerical scores identifying the terms as positive, negative and objective (i.e., neutral), e.g., *majestic* score 0.75 (positive term), and *invalid* score 0.75 (negative). The study in [17] presents the use of *SentiWordNet 3.0* as the basis for the development of extremism lexical resource, a comprehensive lexical resource to be used to support sentiment classification and opinion mining applications [18].

SenticNet 5 [13] encodes denotative and connotative information commonly associated with real objects, actions, events, and people. It avoids the indiscriminate use of keywords and word co-occurrences and instead relies on the implicit meaning associated with common sense concepts. Unlike purely syntactic techniques, *SenticNet 5* can detect subtly expressed emotions by analysing multi-word utterances that do not explicitly express emotion but are associated with concepts that do. Here are some examples from the *SenticNet 5* dataset: *favourite* scores 0.87 (positive), *worry* scores -0.93 (negative).

2.4 Sentiment Analysis Datasets

SA development and tuning requires sizeable labelled training datasets, also known as the SA training dataset. The first step in developing the analysis

requires an SA dataset with thousands of statements already labelled as positive, negative, or neutral. Finding training data is difficult because a human expert must determine and label the polarity of each statement in the training data. Using already available training data reduces the time and effort required to develop a new dataset. The work in [19] uses *Sentiment 140* [20] and *SentiStrength* on a prominent representative set of research articles, explicitly applying some techniques to sentiment analysis of articles circulating on Twitter. The dataset consists of two CVS files, one for testing and one for training. *Sentiment 140* provides a sentiment value per tweet on a scale from 0 (negative) to 4 (positive). The values have been converted into three sentiment categories: positive, negative, and neutral for better comparison. In our work, we chose the test file for the evaluation of our system.

The authors of the paper [21] use the Twitter for Sentiment Analysis (T4SA) [22] visual dataset, which contains text and multimedia data to examine user sentiment. The authors collected Twitter data via a continuous tracker for six months and used it for a visual assessment of SA. The study of [23], which aims to detect users' opinions on movie reviews using the RT-polarity [24] dataset, classified 2000 comments into two different categories. In general, the comments mainly consist of sentences. The authors classify user sentiments at the sentence level and then classify all comments as opinions. The resulting collection consists of two files, one for each set of 5331 positive and negative opinions.

TurntoIslam [25] and *Ansar1* [26] both with posts are organized into threads that generally indicate the topic being discussed and focus on extremist (e.g., jihad) and general Islamic religious discussions. Each post contains detailed metadata, e.g., date and member name. As advertised on the forum, this is an English-language forum aimed at correcting common misconceptions about Islam. Radical participants also occasionally express support for fundamentalist militant groups. These two corpora will help us understand whether our approach works well in extremist religious discourse (e.g., jihadist) and general Islamic discourse.

Although a large number of approaches exist and few studies have offered an explicit comparison between SA techniques, the work of [27] shows comparisons of eight popular SA methods in terms of coverage and agreement. Ribeiro et al. [28] present a sentence-level comparison of twenty-four popular sentiment analysis methods, based on a benchmark of eighteen labelled datasets. Performance was evaluated on two sentiment classification tasks: negative vs positive and three classes, namely negative, neutral and positive. However, these studies never compare the effectiveness of sentiment analysis methods or sentiment lexicons on the specific task of identifying extreme sentiments, i.e., extremely positive and extremely negative sentiments. To the best of our knowledge, the present work is one of the few direct attempts to identify extreme sentiments, i.e., extremely positive and/or extremely negative sentiments on social platforms, using BERT [8].

2.5 ExtremeSentiLex

In our previous work [1], we proposed an unsupervised approach for automatic detection of people's extreme sentiments on social networks. The approach is based on two steps: 1) Extreme Sentiment Generator (ESG) - we automatically build a standard lexicon consisting of extreme positive and negative sentiment terms, and extend that same lexicon with a method based on word embeddings; 2) Extreme Sentiment Classifier (ESC) - to validate the lexicon, using an unsupervised approach for automatic detection of extreme sentiments. We further evaluated our performance on five different social networks and media datasets (Section 2.4).

We designed and developed a prototype system composed of two components, i.e., ESG and ESC. ESG, based on statistical methods, is applied on *SentiWordNet 3.0* and *SenticNet 5* to generate a standard lexical resource known as ExtremeSentiLex that contains only extreme positive and negative terms. Additionally, we extend this new lexicon with new terms through the word embedding method [29], so we can study the behaviour of our tools when tested with more terms. Antiextremism agencies can also use these lexical resources to find an extreme opinion(s) on social networks to counter violent extremism. We embed the lexicons in the ESC and run them on the compilation of five different datasets, constituted of social network and media posts, (Section 2.4). The purpose of this experimentation is to assess the performance of our tool, and this evaluation will validate our hypothesis that the ESC finds posts with extremely negative and positive sentiments in these datasets. To obtain more objective results, we use a confusion matrix to calculate recall, precision, F1 score, and accuracy to check the performance of the ESC.

In preceding work, we presented and discussed the initial results of each dataset individually. The arrangements for these tables are different, according to each dataset itself from the original settings. In our case, **P** - Positive, **N** - Negative and **Neutral** are the original polarity of the posts, EP are posts classified as positive extremes, EN means posts classified as negative extremes, and E+INC are posts classified as non-extreme or inconclusive.

We concluded that the extended lexicon detects more extreme posts. There is an almost 2%-5% increase in each category for RT-polarity, Sentiment 140, TurntoIslam and Ansar1datasets. The significant increase is in the result of T4SA data, almost 22% to 24% for a total number of the extreme and total number of positive extreme and 1% of total negative extreme. We also concluded that by extending the original lexicon with related terms, our tool identified more extreme posts, making sense since social media posts tend to be short, so a more extensive lexicon has a higher probability of detecting extreme sentiments on these short texts. The results obtained by using an extended lexicon can be seen in Table 1.

	Datasets		
	RT-polarity	Sentiment140	T4SA
Recall $_{EP}$	**92**%	**97**%	**98**%
Recall $_{EN}$	41%	45%	43%
Precision $_{EP}$	64%	64%	**81**%
Precision $_{EN}$	81%	**93**%	**89**%
F1 Score $_{EP}$	75%	**77**%	**88**%
F1 Score $_{EN}$	54%	60%	58%
Accuracy	67%	71%	**82**%

Table 1: Results obtained using the extended lexicon.

3 Methodology

The objective of our work is to validate the extended lexicon which classified extreme posts. To carry out this assignment, we will utilize deep learning transformer based model introduced by Google, known as BERT. Section 3.1 and 3.2 briefly explain the working of BERT and its application for our use case.

3.1 Understanding BERT

BERT stands for "Bidirectional Encoder Representations from Transformers" [30]. It is designed to pre-train deep bidirectional representations from the unlabelled text by jointly conditioning the left and right contexts. Its pre-trained model acts as the brain, which can then learn and adjust to the increasingly large resources of discoverable content and queries and be fine-tuned to user's specifications. This process is called as transfer learning. Therefore, the pre-trained BERT model can be fine-tuned with a single additional output layer to create state-of-the-art models for different NLP problems. It encodes context bidirectionally and requires minimal architectural changes for a wide range of natural language processing tasks [8]. Using a pre-trained transformer encoder, BERT can represent any token based on its bidirectional context.

BERT is pre-trained on a massive corpus of unlabelled text, including Wikipedia (2,500 million words) and Book Corpus (800 million words). This pre-training step is half of the magic behind BERT's success. As the model is trained on a large text corpus, the model starts to pick up a more profound and intimate understanding of how the language works. This knowledge is the backbone that is useful for almost any NLP task. The most helpful feature of BERT is fine-tuning, whereby by adding just some of the additional output layers, we can create state-of-the-art models for various NLP tasks. BERT is currently being used by Google to optimize the interpretation of search engine queries. Initially, it was limited to the English language, but by

December 2019, the model had already been rolled out in over 70 languages. BERT perform exceptionally well on various NLP and sequence-to-sequence based language generation related tasks such as question answering, abstract summarization, Sentence prediction, Conversational response generation, polysemy and coreference (words that sound or look the same but have distinct meanings) resolution, word sense disambiguation, natural language inference, and Sentiment classification (Text Classification).

3.2 Fine-tuning BERT for Text Classification

The BERT-base model incorporates an encoder with 12 transformer blocks, 12 self-attention heads, and 768 units of hidden embedding parameters, a sequence of hidden states of the last layer of the model. The original BERT achieved state of the art results on eleven NLP tasks. However, we are only interested in its classification task. BERT has two versions of different model sizes [8]. The base model (BERT-base) uses 12 layers (transformer encoder blocks) with 768 hidden units (hidden size) and 12 self-attention heads. The large model (BERT-large) uses 24 layers with 1024 hidden units and 16 self-attention heads. Notably, the former has 110 million parameters while the latter has 340 million parameters. In our work, here presented, we have fine-tuned our model on pre-trained BERT-base, using 12 layers, 768 hidden units, and 12 self-attention heads. BERT takes an input of a sequence of up to 512 tokens and outputs the sequence representation. The sequence has one or two segments, where the first token of the sequence is always [CLS] and contains the particular classification embedding, and another special token [SEP] is used to separate the segments. BERT picks the final hidden state h of the first token [CLS] for text classification tasks to represent the complete sequence. In order to get the predicted probabilities from the trained model, a softmax classifier is added to the top of the BERT model.

Firstly, the dataset is vectorized for feeding it to the classifier since it is originally in text format. Different models are available for vectorizing text but BERT learns contextual-embedding rather than learning context-free, such as in the case of word2vec. It performs tokenization using the WordPiece [31] method. In addition to [CLS] and [SEP], it adds a new token called [PAD], to make the length of all sentences equal to the specified sequence length required by the model, and an attention mask is introduced to tell the model about [PAD] tokens. These are then used to input the model to obtain vector representation of each token. Since the base model has 12 layers of encoders, tokens are fed into the first encoder, and the output of the first encoder is then given as input for the second encoder, and so on until the last encoder. The last encoder, which is encoder 12 returns the embeddings for all tokens in the sentences. The representation size of each token is 768 in BERT-base model. This phenomenon is shown in Figure 1.

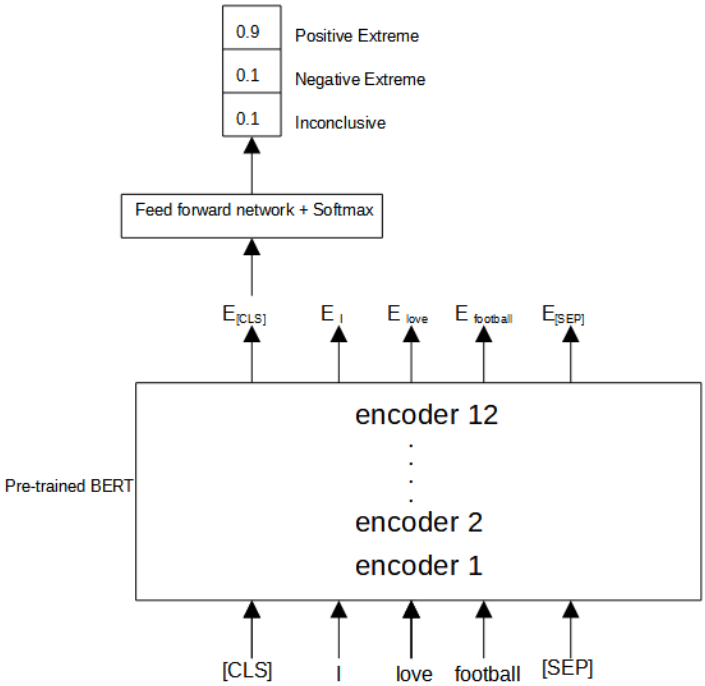

Fig. 1: Tokens are embedded using 12 encoders in BERT-base model and fed into a feed-forward network and softmax function to get the classification probabilities.

For single text classification applications, the BERT representation of the special classification token "[CLS]" encodes information about the entire input text string. The single input text representation is fed into a small MLP[1] consisting of fully connected (dense) layers to produce the distribution of all discrete label values [32].

4 Experimental Setup

We use BERT for this experimental setup, a transformer-based machine learning technique for natural language processing. We use the datasets classified in [1] in the first phase for training while our target variable is polarity which we have calculated. We experiment on each dataset, and at the end, we combine all datasets to check overall performance. This comes in handy as the type of data can be a primary driver in determining the classification. So, the mixing of diverse data challenges the model and allow space for better insights.

Loading Data: We load the classified dataset from Extremesentilex containing six files. We use all of these files for our experiment except the

[1] Multilayer Perceptron.

Sentiment140 training file because of missing the extreme negative class in polarity. The files are in text format that is then loaded in pandas data frames. The primary data we need is in two columns, "message" and "polarity". We drop other columns as they are not needed.

Preprocessing: In preprocessing, we use one-hot encoding for our polarity categorical data. Another extra step is necessary in the case of dataset file T4SA. This file is encoded and contains a byte string. We decode the file and clean the text containing special characters.

Text split for longer text: BERT has a limitation of 512 max length for input characters. It means we cannot input data longer than 512 for training, while our dataset files contain longer text. Extreme *Ansar1* contains 15325 characters long input feature, and extreme *TurntoIslam* has a maximum text length of 10034 characters. This data cannot be used directly as input for the training BERT model. The simple and rough way is to truncate directly, take the initial part up to 512 characters max and discard the rest. Although, this simple naive method is effective in many cases. However, a complex method tackles this issue based on the HIERARCHICAL (cascade) idea, which divides the longer text into smaller chunks and feeds them into the base model [33]. We divide any text larger than 450 characters into 500 characters with an overlap length of 50 words. These split data keeps the same polarity class as its source. It is then given as input for training the BERT model.

Sampling: Our data is highly unbalanced, with high inconclusive and positive extreme polarity outnumbering the small class of extreme negative terms. Training without balancing the data will cause inaccurate results, i.e., predicting well for inconclusive and positive extreme while poorly with negative extreme. Negative extreme happens to be the essential kind of terms which we cannot ignore. So, we use undersampling and a mix of under and oversampling techniques to get the balanced dataset with an equal number of polarity classes.

Train-Test split: After sampling and balancing the dataset, we use 80% of data for training while keeping 10 % for validating the training and 10% for testing the model.

Training and Validation: We use the "bert-base-cased" pre-trained model as most of our data is in English. We run six epochs and a different number of batches for training the model with each dataset. The batch number varies as the higher number for batch size will cause GPU memory constraints. To overcome this limitation, we keep batch size around 6, 8 or 10 depending on the training data size. We use validation data to analyse the training accuracy and loss.

Testing: After training is complete, we run the test data on the trained model to check how well the model performs. The minimum difference between validation and test accuracy reflects the overall accuracy of our model. The results of each dataset are discussed in Section 5.

Performance Metrics: We will use common performance metrics for evaluating our performing model. It includes accuracy, confusion matrix, F1

score, Precision and Recall. As usual, *Accuracy* is the fraction of correct predictions, the number of hits divided by total number of predictions. The *confusion matrix*, also known as *error matrix*, assesses the classification accuracy by calculating the confusion matrix with each row corresponding to the true class. It is displayed in table layout that allows visualization of the efficacy of a classification algorithm.*Precision* is the ability of the classifier to not predict the false label or value. *Recall* is the ability of the classifier to find all the positive samples. It can be also referred as the fraction of the relevant label that are successfully predicted.*F1 score*, also referred as *balanced F-score* or *F-measure*, is the weighted average of the precision and recall. It achieves its best value at 1 and the worst at 0. The F-score is also used for calculating classification problems with more than two classes which is also called as Multi-class classification. These two classes are called micro-averaging and macro-averaging. The final score is obtained by micro-averaging which is biased by class frequency whereas macro-averaging takes all classes as equally important. Another type of F1 score is the weighted average.

There are three types of averages namely micro, macro and weighted. Micro average evaluate metrics globally by computing the total true positives, false negatives and false positives. Macro average tally metrics for each label, and find their unweighted mean. The imbalanced labels are not taken into account. Weighted average determines metrics for each label. It find their average weighted by support. This changes macro average to reckon unbalanced label which can lead to F-score that is distinct from precision and recall.

Support is the number of actual occurrences of the class in the specified dataset. Unbalanced support in the training data may indicate structural weaknesses in the scores of the reported classifiers and could indicate the need for stratified sampling or re-balancing. The *support* does not variate between the models, but rather diagnoses the evaluation process.

5 Results and Discussion

In this section, we present the results obtained for the detection of extreme sentiments by using BERT. There are five datasets in total and one extra dataset combined using the five datasets referred to as comb_all to view the overall performance of our model. Hence, six experimental results and findings are outlined for each dataset.

The datasets used in this experiment are acquired from our previous work, which we refer to as the extended lexicon [1] . The comprehensive lexicon data is highly unbalanced, as shown in Table 2. The number of negative extremes is very low compared to inconclusive and positive extremes. The split of this data into train, validation, and test sets cause further isolation, which results in higher accuracy of dominant classes while ignoring the minor class, which is negative extreme. To tackle this issue, we keep the same number from each class and use sampling techniques to increase the number of records to feed the model.

	Datasets				
	RT-polarity	Sentiment 140	T4SA	TurntoIslam	Ansar1
Total of Extreme	2518 (≈24%)	63 (≈13%)	423689 (≈36%)	120644 (≈36%)	12002 (≈41%)
Extreme Positive	2518 (≈18%)	2518 (≈10%)	2518 (≈32%)	2518 (≈33%)	2518 (≈36%)
Extreme Negative	2518 (≈6%)	2518 (≈3%)	2518 (≈4%)	2518 (≈3%)	2518 (≈5%)
Total	2518 (≈100%)	2518 (≈100%)	2518 (≈100%)	2518 (≈100%)	2518 (≈100%)

Table 2: Extreme posts detected from datasets using the extended lexicon.

RT-polarity: this is a dataset of classified movies, containing the polarity of tweets calculated in the first phase (Section 4). It is highly skewed towards the *Inconclusive* and the *Positive Extreme* classes, with a tiny percentage of *Negative Extreme*. In order to train the model, we use the undersampling technique and take equal numbers from each class of inconclusive, positive extreme and negative. After running three epochs of our model, the validation accuracy of our model becomes flat, around 67%, and our test accuracy is the same, which shows the model is performing well.

The model gives the following classification report for RT-polarity. The overall accuracy of the model is 67%. Table 3 also highlights the precision, recall and F1-score for our target classes.

RT-polarity				
	precision	recall	f1-score	support
Inconclusive	0.63	0.71	0.67	31
Positive Extreme	0.64	0.48	0.55	29
Negative Extreme	0.74	0.84	0.79	25
accuracy			0.67	85
macro avg	0.67	0.68	0.67	85
weighted avg	0.67	0.67	0.66	85

Table 3: Classification report of RT-polarity dataset.

The use of a *confusion matrix* allows us to have a better view of the efficacy of our model, revealing specifically the type of errors being committed. It shows the values of true positive, false positive and vice versa for each class.

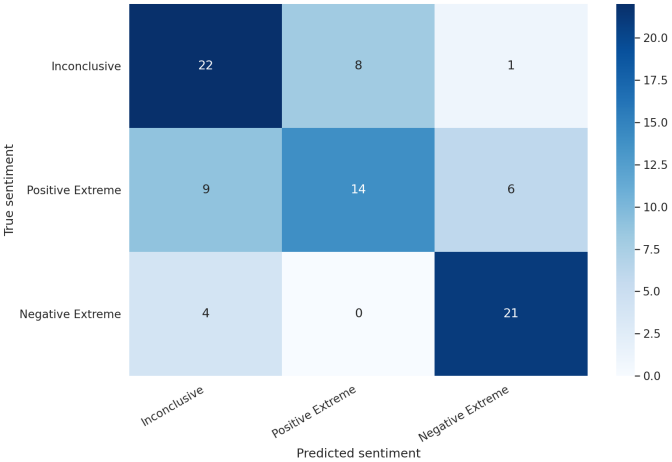

Fig. 2: Confusion matrix of RT-polarity dataset.

T4SA: The next dataset we present is T4SA (Twitter sentiment analysis). Its text is in byte string format and we have applied a special functionality to clean and decode the text in order to finalise it for training as it contains emojis. Sentiment analysis is widely used on twitter datasets because of its usefulness. Our model gives 99% training and 98% validation accuracy after running six epochs, supposedly because BERT is pre-trained on a large corpus of unlabeled text, we have obtained 98% test accuracy. The classification report on the T4SA dataset is given in Table 4:

T4SA				
	precision	recall	f1-score	support
Inconclusive	0.98	0.98	0.98	2381
Positive Extreme	0.99	0.99	0.99	2484
Negative Extreme	0.99	0.99	0.99	2370
accuracy			0.99	7235
macro avg	0.99	0.99	0.99	7235
weighted avg	0.99	0.99	0.99	7235

Table 4: Classification report of T4SA dataset.

The performance of the classification model onset of test data for which the actual values are known is shown in the confusion matrix.

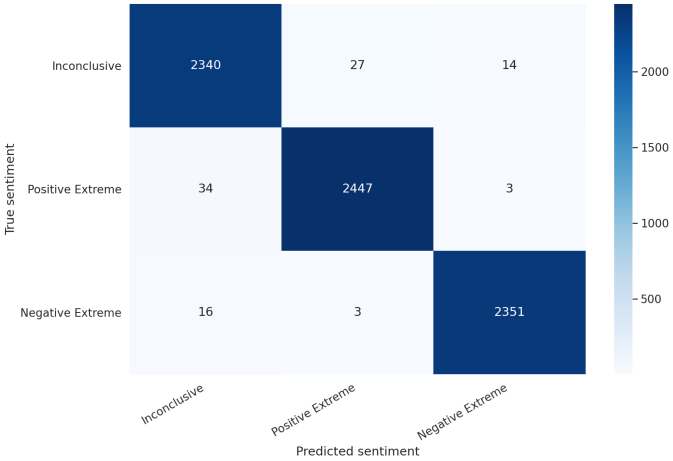

Fig. 3: Confusion matrix of T4SA dataset.

Sentiment140: The sentiment140 data contains two files. We chose sentiment140_test for the experiment and discard the sentiment140_train file because of the absence of the "Extreme Negative" class in polarity. The text, which is the training variable, is the message that needs preprocessing. We clean the text and feed it to our classification model. We run six epochs of our training data and obtain 92% training accuracy and 88% of validation accuracy. After the third epoch, the training accuracy becomes flat at 92% while the validation accuracy remains between 86% and 88%. The accuracy we get on the test set is 88% which is the same as the validation. The classification report for Sentiment140_test given in Table 5 and Figure 4 shows the confusion matrix for Sentiment140_test dataset.

Sentiment140_test				
	precision	recall	f1-score	support
Inconclusive	1.00	0.88	0.93	24
Positive Extreme	0.80	0.80	0.80	15
Negative Extreme	0.80	1.00	0.89	12
accuracy			0.88	51
macro avg	0.87	0.89	0.87	51
weighted avg	0.89	0.88	0.88	51

Table 5: Classification report of Sentiment140_test dataset.

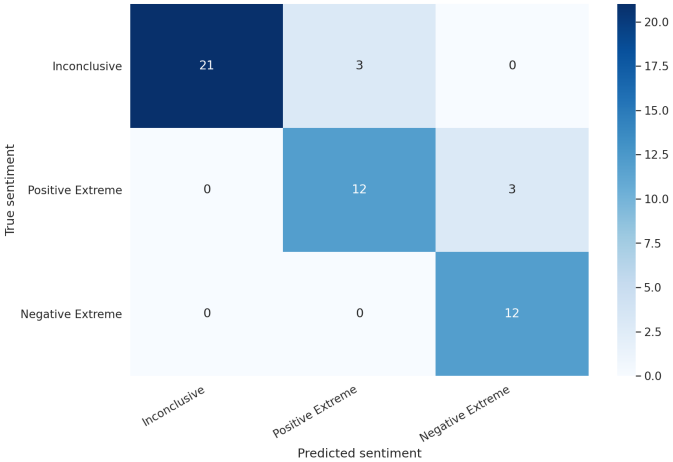

Fig. 4: Confusion matrix of Sentiment140_test dataset.

TurntoIslam: The TurntoIslam dataset contains two extra classes of our target variable, "polarity," similar to Ansar1. We regard the Negative Non-Extreme and Positive Non-Extreme as inconclusive. The main goal of our work is to find Extreme positive and Extreme negative terms. Therefore we regard other terms as inconclusive or neutral.

The text we will use to train the model is long, so we make chunks of 500 words if the text is more significant than 450 characters, and we maintain the polarity values for each chunk. We then use this data for training our model. Although our training accuracy increases by training epochs, the validation accuracy remains somewhat consistent, around 77%. The accuracy we get on the test set is 79% which exhibit little difference from the validation accuracy. The classification report of the dataset TurntoIslam shows the overall accuracy we obtained is 79% as given in Table 6

TurntoIslam				
	precision	recall	f1-score	support
Inconclusive	0.96	0.94	0.95	282
Positive Extreme	0.78	0.71	0.75	276
Negative Extreme	0.94	0.85	0.89	329
Positive Non-Extreme	0.62	0.74	0.67	276
Negative Non-Extreme	0.68	0.70	0.69	286
accuracy			0.79	1449
macro avg	0.80	0.79	0.79	1449
weighted avg	0.80	0.79	0.79	1449

Table 6: Classification report of TurntoIslam dataset.

The confusion matrix of TurntoIslam given in Figure 5 indicates that the model works well for the Negative Extreme, which is the most important class of our dataset.

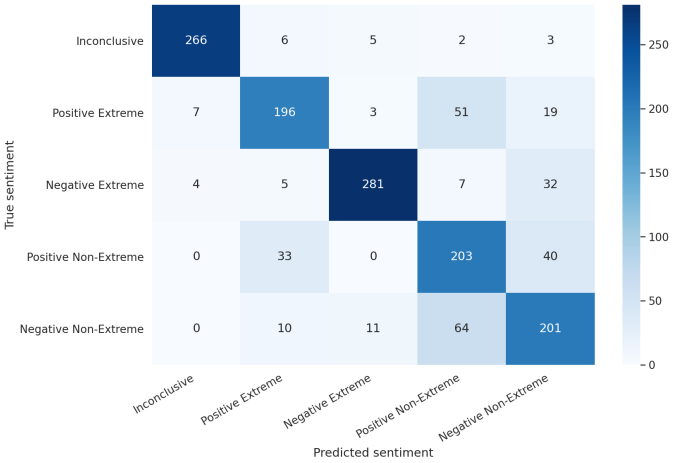

Fig. 5: Confusion matrix of TurntoIslam dataset.

Ansar1: Ansar1 is a dark web forum which contains mixed language discussions on forum. The dataset consist of these discussions from the forum. Ansar1 dataset yields lower accuracy compared to other datasets. This can be attributed to the presence of foreign languages other than the English language. While the central part consists of English, The mixture of another language, especially Arabic, can influence the model, since we are using the BERT pre-trained model in English. Although multilingual pre-trained models are available for BERT, they do not provide better results with the data we are using. The text data it contains is gathered from forum posts. We split the more extensive texts into smaller chunks to feed the BERT model. Although the training accuracy exceeds 80%, the validation accuracy remains between 56% to 58% while running epochs of our model. The accuracy we get on the test set is 57% which is an approximate value of our validation accuracy. The classification report for Ansar1 data set is shown in Table 7. Figure 6 gives the comparison of results for each class in the form of confusion matrix.

Ansar1				
	precision	recall	f1-score	support
Inconclusive	0.82	0.82	0.82	74
Positive Extreme	0.56	0.58	0.57	52
Negative Extreme	0.47	0.46	0.47	61
Positive Non-Extreme	0.44	0.50	0.47	50
Negative Non-Extreme	0.51	0.45	0.48	62
accuracy			0.58	299
macro avg	0.56	0.56	0.56	299
weighted avg	0.58	0.58	0.58	299

Table 7: Classification report of Ansar1 dataset.

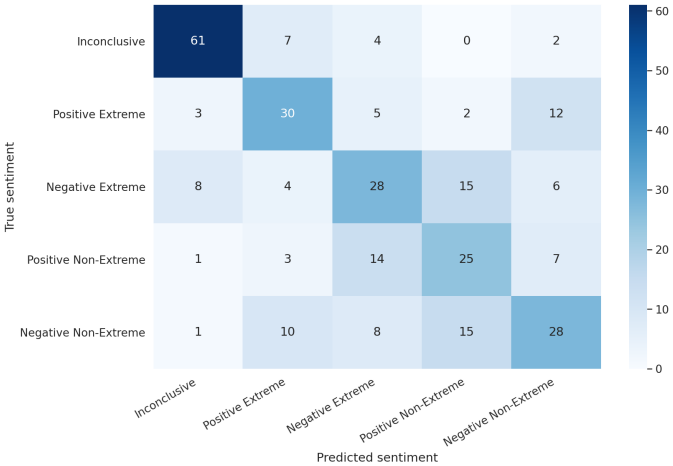

Fig. 6: Confusion matrix of Ansar1 dataset.

comb_all: Our last dataset for the experiment is made by combining all the previous five datasets we tested on. We take data from all datasets and create a new dataset to try with our model. We call this dataset comb_all. To preserve the data consistency, we keep the main classes of polarity, which are "Inconclusive, Positive Extreme and Negative Extreme". This combination of data into a new dataset provides a good insight about the efficacy of using BERT, which seems to perform well. It proves the reliability of BERT for our task and related ones. Also, it verifies our approach to classify into extreme positive and extreme negative classes. Our training accuracy reaches around 98%, while the validation accuracy of our model is around 95%. The accuracy we get on the test set is also 95%, which confirms the validation accuracy. The classification report of comb_all is shown in Table 8. Figure 7 shows the confusion matrix of our model which achieves good results for extreme values especially for *Negative Extreme*.

comb_all				
	precision	recall	f1-score	support
Inconclusive	0.94	0.94	0.94	2904
Positive Extreme	0.95	0.95	0.95	3110
Negative Extreme	0.96	0.97	0.96	3044
accuracy			0.95	9058
macro avg	0.95	0.95	0.95	9058
weighted avg	0.95	0.95	0.95	9058

Table 8: Classification report of comb_all dataset.

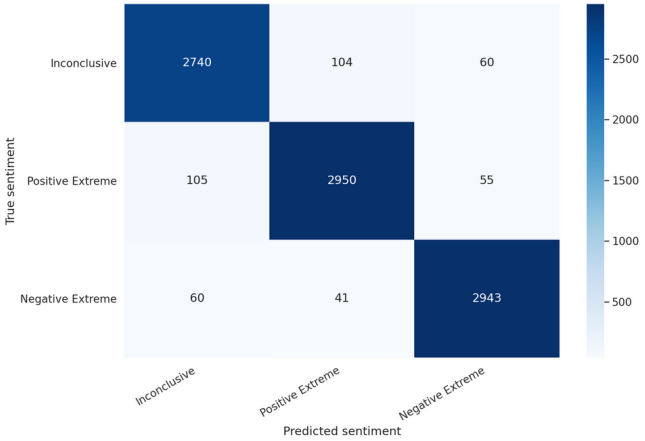

Fig. 7: Confusion matrix of comb_all dataset.

For a comprehensive analysis and visualization of the difference between the original lexicon and the expanded one, using BERT, we can look at the results presented in Table 9, for each dataset. We further plotted graphically these results in Figures 8, 9, and 10.

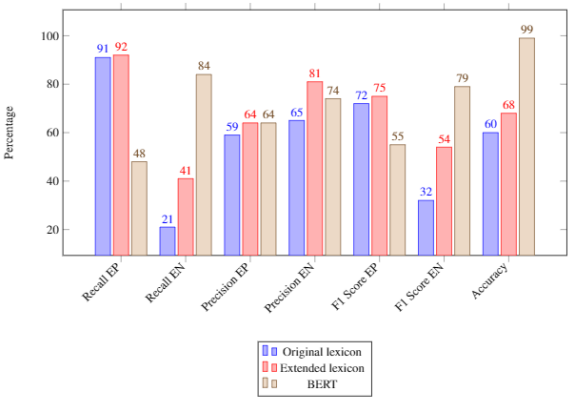

Fig. 8: Comparison between results of RT-polarity

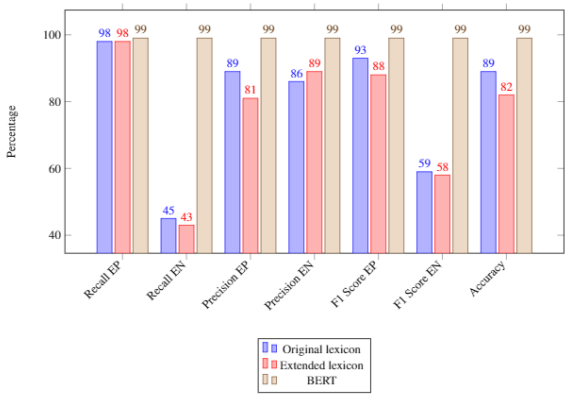

Fig. 9: Comparison between results of T4SA.

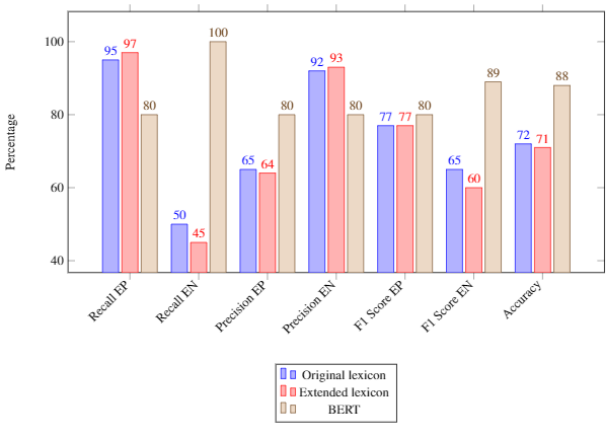

Fig. 10: Comparison between results of Sentiment 140.

	Datasets		
	RT-polarity	Sentiment140	T4SA
Recall $_{EP}$	48%	**9**9%	80%
Recall $_{EN}$	**8**4%	**9**9%	**1**00%
Precision $_{EP}$	64%	**9**9%	80%
Precision $_{EN}$	74%	**9**9%	80%
F1 Score $_{EP}$	**5**5%	**9**9%	**8**0%
F1 Score $_{EN}$	**7**9%	**9**9%	**8**9%
Accuracy	67%	**9**9%	**8**8%

Table 9: Results obtained using BERT.

In terms of overall accuracy, BERT shows promising results for the datasets RT polarity, Sentiment140 and T4SA. Mainly, it successfully classifies the Extreme Negative (EN) after fine-tuning the base model of BERT as seen in Table 9.

6 Conclusion and Future Work

In this paper, we demonstrate an unsupervised and language-independent approach for detecting people's extreme sentiments on social media platforms. Our approach is based on defining extreme polarity for terms and generating extreme sentiment lexicon by relying upon two standard lexical resources, i.e., *SentiWordNet 3.0* and *SenticNet 5*. With this work, we provide a standard

lexicon consisting of extreme positive and negative terms polarity. We implemented a prototype system with two different components *ESG* and *ESC*. We experimented with our system on five social networks and media data lexicons to analyze its accuracy, effectiveness, and efficiency. We have also used word embeddings to extend the lexicon to analyze the improvement in systems' performance. The obtained results are promising and encouraging, as the system shows excellent improvement using the extended lexicon. This standard lexicon can also be helpful for other researchers to exploit it for SA studies and anti-extremism authorities, allowing them to identify and prevent violent extremism early.

As an extension of this research, we want to improve and handle the issues and limitations identified to make our system more efficient. For this, we will apply linguistic tools in our approach, for example, to detect negation [34, 35] (*he is happy* is the opposite of *he is not happy*), to detect expressions with intensifiers [36] (e.g., *he likes a lot*). For future research, we also plan to enhance our approach using NLP techniques to detect radical elements on social media and networks. A radical event has some specific features being quite different from the identification of extremism, as, for example, a radical behaviour does not imply the manifestation of extreme sentiments.

Acknowledgments. This work was supported by National Founding from the FCT- Fundacão para a Ciência e a Tecnologia, through the MOVES Project-PTD/EEI-AUT/28918/2017 and by operation Centro-01-0145-FEDER-000019-C4 - Centro de Competências em Cloud Computing, co-financed by the European Regional Development Fund (ERDF) through the Programa Operacional Regional do Centro (Centro 2020), in the scope of the Sistema de Apoio à Investigação Científica e Tecnológica.

References

[1] Pais, S., Tanoli, I.K., Albardeiro, M., Cordeiro, J.: Unsupervised approach to detect extreme sentiments on social networks. In: 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 651–658 (2020). https://doi.org/10.1109/ASONAM49781.2020.9381420

[2] Persia, F., D'Auria, D.: A survey of online social networks: challenges and opportunities. In: 2017 IEEE International Conference on Information Reuse and Integration (IRI), pp. 614–620 (2017). IEEE

[3] Becker, H., Naaman, M., Gravano, L.: Selecting quality twitter content for events. In: Fifth International AAAI Conference on Weblogs and Social Media (2011)

[4] Krumm, J.S.: Influence of social media on crowd behavior and the operational environment. Technical report, ARMY COMMAND AND

GENERAL STAFF COLLEGE FORT LEAVENWORTH KS SCHOOL OF ... (2013)

[5] Scanlon, J.R., Gerber, M.S.: Automatic detection of cyber-recruitment by violent extremists. Security Informatics **3**(1), 5 (2014)

[6] Pais, S., Tanoli, I., Albardeiro, M., Cordeiro, J.: A lexicon based approach to detect extreme sentiments. In: Proceedings of the Fifteenth International Conference on Internet Monitoring and Protection. ICIMP '20. IARIA, ??? (2020)

[7] Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT press, ??? (2016)

[8] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)

[9] Ruder, S., Peters, M.E., Swayamdipta, S., Wolf, T.: Transfer learning in natural language processing. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials, pp. 15–18 (2019)

[10] Ahmad, S., Asghar, M.Z., Alotaibi, F.M., Awan, I.: Detection and classification of social media-based extremist affiliations using sentiment analysis techniques. Human-centric Computing and Information Sciences **9**(1), 24 (2019)

[11] Kaur, A., Saini, J.K., Bansal, D.: Detecting radical text over online media using deep learning. CoRR **abs/1907.12368** (2019) https://arxiv.org/abs/1907.12368. https://doi.org/https://arxiv.org/abs/1907.12368v2

[12] Jaki, S., Smedt, T.D.: Right-wing german hate speech on twitter: Analysis and automatic detection. CoRR **abs/1910.07518** (2019) https://arxiv.org/abs/1910.07518. https://doi.org/https://arxiv.org/abs/1910.07518v1

[13] Cambria, E., Poria, S., Hazarika, D., Kwok, K.: Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)

[14] Wagh, B., Shinde, J., Kale, P.: A twitter sentiment analysis using nltk and machine learning techniques. International Journal of Emerging Research in Management and Technology **6**(12), 37–44 (2018)

[15] Mane, S.B., Sawant, Y., Kazi, S., Shinde, V.: Real time sentiment analysis

of twitter data using hadoop. IJCSIT) International Journal of Computer Science and Information Technologies **5**(3), 3098–3100 (2014)

[16] Bouazizi, M., Ohtsuki, T.: A pattern-based approach for multi-class sentiment analysis in twitter. IEEE Access **5**, 20617–20639 (2017)

[17] Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In: Lrec, vol. 10, pp. 2200–2204 (2010)

[18] Pang, B., Lee, L., *et al.*: Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval **2**(1–2), 1–135 (2008)

[19] Friedrich, N., Bowman, T.D., Stock, W.G., Haustein, S.: Adapting sentiment analysis for tweets linking to scientific papers. arXiv preprint arXiv:1507.01967 (2015)

[20] Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford **1**(12), 2009 (2009)

[21] Vadicamo, L., Carrara, F., Cimino, A., Cresci, S., Dell'Orletta, F., Falchi, F., Tesconi, M.: Cross-media learning for image sentiment analysis in the wild. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 308–317 (2017)

[22] T4SA. http://www.t4sa.it/#dataset

[23] Smeureanu, I., Bucur, C., *et al.*: Applying supervised opinion mining techniques on online user reviews. Informatica Economică **16**(2), 81–91 (2012)

[24] Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the ACL (2005)

[25] Artificial Intelligence Lab, U.o.A. Management Information Systems Department: Turn to islam forum dataset. University of Arizona Artificial Intelligence Lab, AZSecure-data, Director Hsinchun Chen, ??? (2013)

[26] Ansar1 forum dataset. University of Arizona Artificial Intelligence Lab, AZSecure-data, Director Hsinchun Chen (2013)

[27] Gonçalves, P., Araújo, M., Benevenuto, F., Cha, M.: Comparing and combining sentiment analysis methods. In: Proceedings of the First ACM Conference on Online Social Networks, pp. 27–38 (2013)

[28] Ribeiro, F.N., Araújo, M., Gonçalves, P., Gonçalves, M.A., Benevenuto,

F.: Sentibench-a benchmark comparison of state-of-the-practice sentiment analysis methods. EPJ Data Science **5**(1), 1–29 (2016)

[29] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space (2013)

[30] Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018) https://arxiv.org/abs/1810.04805

[31] Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)

[32] Zhang, A., Lipton, Z.C., Li, M., Smola, A.J.: Dive Into Deep Learning, (2020). https://d2l.ai

[33] Pappagari, R., Żelasko, P., Villalba, J., Carmiel, Y., Dehak, N.: Hierarchical transformers for long document classification. Automatic Speech Recognition and Understanding Workshop, arXiv:1910.10781v1 (2019)

[34] Blanco, E., Moldovan, D.: Some issues on detecting negation from text. In: Twenty-Fourth International FLAIRS Conference (2011)

[35] Sharif, W., Samsudin, N.A., Deris, M.M., Naseem, R.: Effect of negation in sentiment analysis. In: 2016 Sixth International Conference on Innovative Computing Technology (INTECH), pp. 718–723 (2016). IEEE

[36] Mohammad, S.M., Bravo-Marquez, F.: Emotion intensities in tweets. arXiv preprint arXiv:1708.03696 (2017)