

# Towards a user-oriented adaptive system based on sentiment analysis from text:

Adil Baqach<sup>1</sup>, and Amal Battou<sup>2</sup>.

<sup>1</sup>IRF-SIC, Department of Computer Science, Faculty of Science, Ibn Zohr university, Agadir, Morocco.

<sup>2</sup>IRF-SIC, Department of Computer Science, Faculty of Science, Ibn Zohr university, Agadir, Morocco.

**Abstract.** Sentiment analysis has known a big interest over recent years due to the expansion of data. It has many applications in different fields such as marketing, psychology, human-computer interaction, eLearning, etc. There are many forms of sentiment analysis, namely facial expressions, speech, and text. This article is more interested in sentiment analysis from the text as it is a relatively new field and still needs more effort and research. Sentiment analysis from text is very important for different fields, for eLearning it can be critical in determining the emotional state of students and therefore, putting in place the necessary interactions to motivate students to engage and complete their courses. In this article, we present different methods of sentiment analysis from the text that exist in the literature, beginning from the selection of features or text representation, until the training of the prediction model using either supervised or unsupervised learning algorithms and although there has been so much work done in this domain, there is still effort that can be done to improve the performance and to do that we first need to review the recent methods and approaches put in place on this field and then try to discuss improvements in certain approaches or even proposing new approaches.

## 1 Introduction:

Predicting individuals' emotional states based on their written texts and feedback is important, but also challenging due to language ambiguity [1]. Most of the time textual expressions are not only direct using emotional words such as "happy" or "angry", but emotions can also be extracted from the interpretation of the meanings and contexts. However, the need for emotion detection is increasing while both structured and unstructured data are getting larger because of social media [2], however, it is still a research area that needs a lot of effort before reaching the success of sentiment analysis.

Emotion detection is critical in the human-machine relation. Emotions may be detected from speech, facial expression, and written text. Compared to text-based emotion recognition, a sufficient amount of work was done regarding facial and speech emotion detection.

Sentiment analysis can bring a tremendous improvement in different applications. In education, predicting students' sentiments can help solve problems such as confusion and boredom which affect their engagement and performance [1, 3]. The use of these sentiments can go further and be the input of other applications such as recommendation systems for students in e-learning environments, which can recommend different pedagogical paths based on the student's sentiments [4, 5]. In marketing, emotion detection could be used to predict customers' emotions regarding products and services, which can help to change

aspects of the product to apply with the customers' needs, so the relationship with customers gets better.

This paper aims to give a review of different methods of detecting emotions from the text that exists in literature, doing such review could give a global sight of the field, and therefore avoid redundancies, and also get an idea of the different existing methods and analysing the opportunity of improvement and innovation. From the literature, it can be observed that there are two main sections to predict emotions from text, the first relies on the text representation using different Natural Language Processing (NLP) approaches, the second considers the algorithm which receives the represented text and builds a model of sentiments prediction, it could be either supervised or unsupervised learning algorithms.

The rest of the paper will be divided as follows, the first section will contain a review of the existent text representation methods, in the second section we will state the well-known and performant supervised and unsupervised learning algorithms used in emotion detection from text, and finally we will conclude and give our predicted future work.

## 2 Text representation methods

### 2.1 One-Hot encoding:

This method is based on representing each word in the text into a 2D vector of zeros, except for the position of its corresponding index which has a value of 1. The output of this method is a matrix of (n x m) with n is the

number of tokens, and m is the number of vocabularies in the text, this matrix is fed to a classifier to predict the text's corresponding sentiment. We can get a better understanding with an example:

I like sentiment analysis

I: [1 0 0 0]

Like: [0 1 0 0]

sentiment: [0 0 1 0]

analysis: [0 0 0 1]

One-hot encoding is used in many studies of text classification, [6] used this approach after pre-processing the text, and then fed the resulted matrix into different deep learning methods to compare them.

## 2.2 Count vectorizer – Basic bag of words:

Bag of words representation puts words into a “bag” and computes the frequency of occurrence of each word. Count vectorizer can squeeze an entire sentence into a vector instead of giving a vector for each token, each element of the vector is a number of occurrences of a word in the sentence.

Count vectorizer can also work with different n-grams, instead of using a single word as a token. [7] implements this method in his study, and he forwarded the resultant vector into a Deep Learning algorithm in a text classification task. And [8] used WordNet affect and Bag of words in his text representation, then he used the lexicon classification method to predict emotions from text.

## 2.3 Word Embedding (Word2Vec):

Word embedding is a featured word-level representation capable of detecting the semantic meaning of words. Each word is represented with a fixed-size vector, with values for each feature of the semantic meaning, which are obtained by random initialization and then updated during training. For this task, Word2Vec is a well-known algorithm, it constructs these vectors via two methods:

- CBOW (Continuous Bag Of Words): Predicts the target word based on the context of its surrounding words.
- Skip-gram: Predicts the surrounding words based on the target word (opposite of CBOW).

After obtaining each word's vector, we can measure the similarity between them depending on if they are normalized or not. If they are normalized, a simple dot product between vectors is calculated to measure similarity. If they are not normalized, cosine similarity is used:

$$\text{cosine similarity} = 1 - \text{cosinedistance} = \frac{u \cdot v}{\|u\| \|v\|} \quad (1)$$

[9] used this method to introduce his proposed model, which is based on Word2Vec. After getting the word vectors using Word2Vec algorithm, [9] used TF-IDF weighting method to give weights to the vectors, then the new approach proposed by the authors is to determine

whether or not a word has sentiment information, so the weight calculation method for each word vector is:

$$w_i = tfidf \cdot e \quad (2)$$

$$e = \begin{cases} \alpha, & \text{if } t_i \text{ is sentiment word} \\ 1, & \text{if } t_i \text{ is not a sentiment word} \end{cases} \quad (3)$$

Then let a be the distributed vector obtained by Word2Vec, then the vector obtained by the proposed method is:

$$v = w_i \cdot a \quad (4)$$

[10] also used this approach in his article, to compare the Word2Vec, TF-IDF and the association of the two methods, and the results showed that merging TF-IDF with Word2Vec gave the best performance.

## 2.4 TF-IDF (Term Frequency – Inverse Document Frequency) Weighting:

It's a method used to re-weight vectors obtained by all other methods based on the frequency of each token, it's a statistical method that evaluates how relevant a word is to a document. It is calculated by multiplying two metrics, Term Frequency (TF) of a word in a document and the Inverse Document frequency (IDF) of a word in a set of documents:

$$tf \ idf(t, d, D) = tf(t, d) \cdot idf(t, D) \quad (5)$$

Where:

$$tf(t, d) = \log(1 + freq(t, d)) \quad (6)$$

$$idf(t, D) = \log\left(\frac{N}{count(d \in D : t \in d)}\right) \quad (7)$$

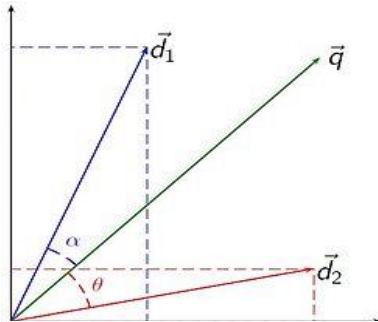
Where:

- N: total number of documents (sentences).
- $d \in D : t \in d$ : number of documents where the term t appears.

This approach is used at a large number of articles, [7] used it in his study to represent words before passing them into a machine-learning algorithm to predict opinion and sentiment from text.

## 2.5 VSM (Vector Space Model):

VSM is an algebraic model for representing a text as a vector of identifiers. This can allow us to identify the similarity between different texts although they do not share the same words. Each term is represented as a multi-dimensional vector, and each dimension represents a different term, if a term occurs in a document, its value would be non-zero, there are many ways to calculate these values (TD-IDF, Co-occurrence ...), After obtaining these vectors, we can easily calculate the similarity between them using different approaches:



**Fig. 1.** Representing word by doc in a vector space

- Euclidian distance: length of a straight line between two vectors:

$$Euc(A, B) = \sqrt{\sum_{i=1}^n (A_i - B_i)^2} \quad (8)$$

- Cosine similarity: cosine of angle between two vectors

$$\cos\theta = \frac{d_j \cdot q}{\|d_j\| \|q\|} = \frac{\sum_{i=1}^N w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^N w_{i,j}^2} \sqrt{\sum_{i=1}^N w_{i,q}^2}} \quad (9)$$

## 2.6 PMI (Pointwise Mutual Information):

For TF-IDF, we can calculate the importance of a word inside a corpus, but what if we want to measure the score of words with respect to a specified category. For this task, another group of feature scoring is used, called “association measures”. The most common association measure is PMI, for example, we can determine the sentiment of a text as “positive” or “negative” by calculating the PMI score between each word in the document with respect to “negative” or “positive” categories. Another use for it is to normalize a vector-space matrix by determining the importance of a word “w” in a category “c”. The mathematical formula for calculating PMI is as follows:

$$PMI(x, y) = \log \left( \frac{p(x,y)}{p(x)p(y)} \right) = \log \left( \frac{p(x|y)}{p(x)} \right) = \log \left( \frac{p(y|x)}{p(y)} \right) \quad (10)$$

Where:

- $P(x|y)$  is the number of documents (sentences) in category  $y$  that contains  $x$ .
- $P(x)$  is the number of documents containing  $x$ .

This approach is used by [11] in his study, after extracting all NAVA words (Noun, Adverb, Verb, Adjective), he calculates PMI of each of the NAVA words with respect to different words that represent different emotions, then they used this PMI score to calculate an emotion vector that will be used in an unsupervised machine learning problem to detect sentiments from the text.

## 2.7 Co-occurrence matrix

A co-occurrence matrix is a matrix representing the number of times words appear together in a fixed context window. Let us understand by an example, let’s say our corpus contains these two sentences:

- I like NLP
- I enjoy text classification

If the window context was 1 for examples, it means that context words for each word are one word to the left and one to the right, therefore:

- I: like (1), enjoy (1)
- Like: I (2), NLP (1)
- NLP: like (1)
- Enjoy: I (1), text (1)
- Text: enjoy (1), classification (1)
- Classification: text (1)

So, the co-occurrence matrix for this corpus is:

$$Co - occurrence = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

## 3 Text Classification

### 3.1 Supervised learning methods

The main difficulty with the supervised learning algorithms is the lack of labeled data, it can be time-consuming to manually label data, as the labeled corpora that works well for a certain problem is not guaranteed to work as well for different classification problems. However, good results have been reached using different supervised learning algorithms.

In a text-to-speech task, [12] worked on a sentiment analysis from children’s fairy tales using the Naïve Bayes algorithm. The goal was to detect emotions from text, to modify prosody, pitch, intensity, and duration cues of the speech signal. The data used in this study were annotated manually, and each sentence was labeled by one of the basic Ekman’s emotions. Then, different features were extracted from the corpus (Conjunctions of selected features, WordNet emotion words, positive/negative word count ...), there were 30 features in total. Two different parameter tuning approaches were used, they are both based on 10-fold-cross validation to choose the best combination of features, the first called sep-tune-eval which used only 50% of the dataset for tuning, the second is same-tune-eval, which uses all data to tune the features. A comparison between the Naïve baseline and the Bag of Words approach, and the extracted 30 features showed that the last gave better performance, and the same-tune-eval was slightly better than the sep-tune-eval.

[13] used Support Vector Machines (SVMs) to recognize emotions from text using a four emotions model (neutral, happiness, anger, and sadness). After extracting the keywords which are the NAVA words (noun, adverb, verb, adjective), they were transformed into a vector-space using VSM, and in order to increase accuracy, the

authors implemented the Rough Set Theory to find a minimal subset from the attributes. After testing this model using a 3-fold-cross validation, Rough Set Theory combined with SVM actually resulted better than using only SVM, in terms of prediction accuracy, the model was the best for all emotions.

In education platforms, usually feedback is collected at the end of the course, but it is more beneficial if it was taken in real-time, [14] used students' real-time feedback to predict their emotions. Student Response Systems (SRS) were used in this task, and then data was labeled manually. They used four machine learning algorithms to classify the data which are Naïve Bayes (NB), Compliment Naïve Bayes (CNB), Maximum Entropy (ME), and SVMs, and unigrams were implemented as features. After using a 10-fold-cross validation, results showed that the best classifiers were SVM with 94% accuracy followed by CNB with 84%.

[15], used a new proposed approach based on merging both lexicon-based classification and machine learning algorithms to detect sentiments from students' written messages on Facebook. First, they implemented a preprocessing on their data (convert to lower case, segmentation into sentences, tokenization...); then they put the lexicon-based approach to work on the preprocessed data, and after obtaining enough labeled sentences, they then used them as a training set for the machine learning classifier. The goal wasn't only to detect students' emotions based on their Facebook messages, but also detecting the change of those emotions, therefore, they focused on data that was changing dynamically (Mean of the sentiment showed by messages, Number of messages written, Number of comments, Number of likes ...), and they observed the change on this data on a week-interval, and then comparing weeks gave them the change of sentiments. Results showed an accuracy of 83% for the proposed approach.

[1] implemented a sentiment prediction system from students' feedback about English lectures, collected using Twitter. After preprocessing the data (tokenization, lower case, removing stop words, punctuation, hashtags, numbers and URLs...), they used as features a different combination of n-grams: Unigrams (UNI), Bigrams (BI), and Trigrams (TRI); and also, they used different classifiers to make a comparison between them: Naïve Bayes (NB), Multinomial NB (MNB), Compliment NB (CNB), SVM, Maximum Entropy (ME), Sequential Minimal Optimization (SMO) and Random Forest (RF). They implemented different class models, from two classes to eight classes. After 10-fold cross-validation, results showed that the best classifier was Compliment Naïve Bayes (CNB), the best performing features were the combination of n-grams, and the models with two classes gave better performance than others.

[2] made a comparison between several machine learning and deep learning approaches to predict students' sentiments from their feedback on educational content. Therefore, the data was trained using the classifiers namely, MLP (deep learning backpropagation algorithm), SMO, Decision Tree, Simple Logistics, Multi-class classifier, K-star (instance-based classifier), Bayes Net, and Random Forest, then test data is applied to the

resulting model and 10-fold cross-validation is performed, final results were evaluated in terms of Accuracy, Root Mean Square Error (RMSE), Sensitivity and Receiver Operating Characteristics (ROC) curve area. Results showed that SMO and MLP-deep learning methods stand out and perform better than all other classifiers.

Another performant approach was introduced by Z. H. Kilimci et al. in their article [10], which is to use Heterogeneous Classifier Ensembles and word embedding as features. The base learners used in this experiment were: Multivariate Benroulli (MVNB) and MNB, SVM, RF, and CNN. As For the ensemble integration strategies, there are two approaches, the first called Majority Voting, which basically classifies according to the class that got more votes from base learners, the second called Stacking, for this method, a meta-data set is generated based on decisions of the base-level classifiers on the original dataset, then a metalevel classifier is trained on the meta-dataset to make predictions. The text was represented using Word2vec word embedding with both CBOW and Skip-gram which gives an Avg-Word2vec by averaging all vectors of word embeddings, and was compared to the representation of the text using TF-IDF, and also compared to merging the two approaches Avg-Word2vec + TF-IDF. The experiment was conducted on eight different datasets, results showed that RF was the best algorithm compared to the base learners, followed by CNN, the heterogeneous ensemble with stack method performs better than individual classifiers, and the majority vote method. And finally merging Avg-Word2vec with TF-IDF gave better results.

[16] implemented sentiment analysis on self-evaluated comments to improve early prediction of academic failure, in other words, their goal was to detect which students are more likely to fail on an early basis. In order to achieve that, a machine learning algorithm and a deep learning algorithm (SVM and CNN) were trained using two types of data, structured data (attendance, homework, completion...) and unstructured data (self-evaluated comments from students). After collecting the data, it was labeled manually and then used in training the algorithms. Results showed that the proposed model which consists of using both structured and unstructured data was better in terms of effectiveness, and CNN outperformed SVM.

[9] introduced a new method to represent text, which is based on Word2vec representation, TF-IDF weighting method and the new proposed approach that consists of giving a bigger weight to words that actually carry sentiment information. After obtaining the vector representation of the text, it was fed to a BiLSTM (Bidirectional Long-Short Memory), to capture context information, and as a result, a document representation which takes into consideration context was obtained and then given as an input to a feedforward neural network to detect sentiment. In order to evaluate the proposed method, a comparison was made between Word2vec, TF-IDF, Seninfo (proposed method), and the combination between all these methods, In addition, a comparison between the feedforward neural network, Long Short-

Term Memory (LSTM), RNN, CNN, and NB. Results showed that combining Seninfo and TF-IDF on the Word2vec representation gave the best results in terms of F1 score, precision and recall, and the precision of the proposed method (BiLSTM + feedforward neural network) was 91% and recall 92% which are better values than other classifiers.

To predict emotion from short texts, [17] proposed to firstly sample term groups that co-occur together to enrich the number of features, secondly, they used two supervised topic models to associate topics with emotions accurately, and finally, they accelerated the algorithm using a combination of the Alias method and the Metropolis-Hasting sampling. The two proposed supervised topics models were: weighted labeled topic model (WLTM) and -term emotion topic model (XETM). The generative process of WLTM consists of first define a one-to-many mapping of each emotion to multiple topics. Second, they used labeled documents to generate the topic probability for each feature. Finally, they employed support vector regression (SVR) to predict emotion distributions of unlabeled documents given the estimated topic probability for each feature. In the generative process of XETM, authors extracted the emotion-topic probability, then the topic-feature probability is derived to predict the emotion probabilities of unlabeled documents. They used three well-known datasets (SemEval, ISEAR, and RendCECps), and after comparing the proposed models with other models (LLDA, BTM, ETM, CSTM, SLTM, and SVR), results showed that WLTM gave the best performance of all other methods, XETM however, gave a modest performance but still better than the base models. After acceleration fWLTM was slightly lower than WLTM in terms of accuracy but still competitive with base models, less time consuming, and more efficient.

[7] made a comparison between different sentiment analysis classifiers, using three techniques: machine learning, deep learning, and an evolutionary approach called EvoMSA. Two corpora were used in this study, sentiText which is labeled as positive and negative emotions, and eduSERE which has four emotions (engaged, excited, bored, and frustrated). These two corpora were gathered using web scraping over educational platforms (Udemy, Platzi ...) and then labeled using multiple dictionaries. After preprocessing the corpora (remove punctuation and stop words, transforming abbreviations, deleting links, and URLs...), for the machine learning algorithm, each word is represented using TF-IDF and then training the model. For the deep learning algorithm, the text was represented using a bag of words to determine the frequency of each word in the corpora and then choosing the number of layers, inputs, and output neurons per layer, layer type (CNN, LSTM), activation function, loss function, and the optimization method. Finally, for the EvoMSA, which is based on combining the outputs of different text classifiers and then giving a final prediction. Fourteen classifiers were evaluated: EvoMSA, 8 ML algorithms (MNB, KNN, DT, B4MSA, BNB, SVM, LSVM, and RF) and 5 DL algorithms (LSTM, CNN with two different layers, CNN + LSTM, and BERT), results then showed

that for sentiText, best models were EvoMSA and BERT with a 93% accuracy, however, B4MSA showed good results with 92% accuracy and SVM with 90% accuracy. But for eduSERE, the performance decreases, and the best was EvoMSA with 84% accuracy, followed by B4MSA and BERT with 83% accuracy.

In another study, [6] used deep learning approaches to analyze sentiments from IMDB reviews. The classification of reviews is necessary, for researchers, it can be based on the relevance of the sentiment and ratings of the film. It is also beneficial for both, users as it could be a recommendation tool for movie selection, and film companies, which can be used for marketing decisions. In this study, the authors implemented three deep learning algorithms namely RNN, LSTM, and CNN, to compare between them and decide which one is better for review classification. The dataset used in this study is the public IMDB dataset, then it was preprocessed by removing punctuation, and then it was converted to a one-dimensional vector. After testing the three models, results showed that CNN outperformed the other models and achieved higher accuracy (around 88.22%). Besides, RNN, and LSTM are found to perform better than SVM.

### 3.2 Unsupervised methods

[11] used a new approach to predict emotions from the text called UnSED, this approach is based on a calculation of an emotion vector for each affect-bearing word, using the semantic relation between different words, then fine-tuning the vector using the syntactic dependencies of each sentence. There are four main components of this framework. First, preprocessing the data, which includes sentence parsing, parts-of-speech tagging, and syntactic dependency parsing. Then, analyzing the text semantically, which is done on a word level to calculate the emotion vector of each affect-bearing word by calculating their semantic relatedness to emotion concepts. Then analyzing the text syntactically, which is done at phrase-level by using context to adjust the pre-calculated vector. Finally, sentence analysis, by aggregating the emotion vectors of all affect-bearing words and deducing the emotion label of the sentence. The affect words or NAVA words are extracted and looked up in a lexical resource to measure their emotional affinity. And the syntactic dependencies are represented as  $d(w1\downarrow, w2\uparrow)$ , where  $d$  specifies a syntactic relation (nominal subject, negation ...), and the arrows  $\downarrow$  and  $\uparrow$  represent the modified and the modifier respectively. The semantic relation is calculated using PMI:

$$PMI(x, y) = \frac{co-occurrence(x, y)}{occurrence(x).occurrence(y)} \quad (11)$$

And then to calculate the emotion vector  $\sigma_{wi}$  for NAVA words is to use PMI between the NAVA word  $w_i$  and the word representing each emotion:

$$PMI(w_i, e_j) = \sqrt[r]{\prod_{g=1}^r PMI(w_i, K_j^g)} \quad (12)$$

And:

$$\sigma_{w_i} = \langle PMI(w_i, e_1), \dots, PMI(w_i, e_m) \rangle \quad (13)$$

Where  $K_j$  is a set of  $r$  representative words for emotion category  $e_j$ . And then to adjust the emotion vector with context, for the adjectival complement and adjectival modifier:

$$\sigma'_{\beta_q} = \frac{\sigma_{\beta_q} + \sigma_{\alpha_p}}{2} \quad (14)$$

Where  $\beta_q$  is the dependent word and  $\alpha_p$  its influencing word. And for the negation relationship, the dependent word's score is set to zero. Finally, the emotion vector of the whole sentence is calculated by averaging the emotion vectors of all NAVA words included in the sentence. After obtaining the emotion vector  $\sigma_s = \langle s_1, \dots, s_m \rangle$ , if the highest score is above a certain threshold  $t$ , the sentence is labeled with that emotion. Otherwise, it is classified as neutral.

$$w_s = \begin{cases} e_k \text{ if } \max_{i=1, \dots, m} (s_i) = s_k \text{ and } s_k \geq t \\ \text{neutral otherwise} \end{cases} \quad (15)$$

Results showed that the UnSED approach performed better than the Keyword baseline, PLSA, and DIM and it is comparable to LSA. And the context-based method performs better than the context-free one.

In another study on unsupervised learning methods for emotion detection from text, [18] proposed a new method based on emotional signals. Emotional signals are any information that could be correlated with the sentiment polarity of a word, namely, emoticons, and consistency theory, i.e., words that co-occur together tend to have the same sentiment polarity. Two types of emotion signals were introduced in this study, emotion indication that strongly affects the emotional tendency of a text (emoticons), and emotion correlation which reflects the correlation between posts (emotion consistency). In order to model emotion indication, the difference between the last and the emotion polarity should be minimized using the loss function:

$$\|U(i, *) - U_0(i, *)\|_2^2 \quad (16)$$

Where  $U \in \mathbb{R}^{m \times c}$  is the post-sentiment matrix,  $U_0 \in \mathbb{R}^{m \times c}$  is the emotion indication matrix, and  $c$  is the number of sentiment classes.

For the emotion correlation, a post-post graph was implemented, the adjacent matrix of this graph is represented as:

$$W^u(i, j) = \begin{cases} 1 \text{ if } u_i \in \mathcal{N}(u_j) \text{ or } u_j \in \mathcal{N}(u_i) \\ 0 \text{ otherwise} \end{cases}$$

Where  $u_j$  is a post and  $\mathcal{N}(u_j)$  is  $k$ -nearest neighbors of the post based on textual similarity and social network information.

Then these emotional signals were exploited by the Orthogonal Non-negative Matrix Tri-Factorization (ONMTF) which is based on clustering data based on the distribution of features, and features are clustered based on their distribution of data instances. After conducting many experiments and comparing the proposed model with traditional lexicon-based methods, basic ONMTF with no signal information, and other literature methods that incorporate the emotional signals, results showed that the proposed method ESSA outperforms all baselines.

## 4 Conclusion and Future work:

In this paper, we reviewed the current state of sentiment analysis from textual data, the different methodologies to represent text to generate meaningful and performing features, and the different learning models of predicting the emotions. It is obvious that there are many difficulties faced in this field, from one hand, the quality of data is representing a big challenge for learning algorithms, more time is spent in gathering and labeling data and it is often not enough to have well-represented data for this problem, from another hand, the models are still insufficient, we can obtain a performing model that works for certain datasets but does not work as well for other datasets, and for this reasons, it is necessary to bring in more research, therefore discover opportunities of proposing improvements in the existent methods, and even proposing new models to face this challenge, and that is what will be our future focused goal.

## References:

1. N. Altrabsheh, M. Cocca & S. Fallahkhair Predicting learning-related emotions from students' textual classroom feedback via Twitter, In *The 8th International Conference on Educational Data Mining (EDM)* (pp. 436-440) (2015).
2. J. Sultana, N. Sultana, K. Yadav & F. AlFayez, Prediction of Sentiment Analysis on Educational Data based on Deep Learning Approach. in *2018 21st Saudi Computer Society National Computer Conference (NCC)* 1–5 (IEEE, 2018).
3. P. Rodriguez, A. Ortigosa & R. M. Carro Detecting and making use of emotions to enhance student motivation in e-learning environments. *International Journal of Continuing Engineering Education and Life Long Learning* **24**, 168–183 (2014).
4. Y. Madani, H. Ezzikouri, M. Erritali & B. Hssina Finding optimal pedagogical content in an adaptive e-learning platform using a new recommendation approach and reinforcement learning. *J Ambient Intell Human Comput* **11**, 3921–3936 (2020).
5. M. Birjali, A. Beni-Hssane & M. Erritali A novel adaptive e-learning model based on Big Data by using competence-based knowledge and social learner activities. *Applied Soft Computing* **69**, 14–32 (2018).

6. P. Cen, K. Zhang & D. Zheng Sentiment Analysis Using Deep Learning Approach. *Journal on Artificial Intelligence* **2**, 17–27 (2020).
7. M. L. B. Estrada, R. Z. Cabada, O. R. Bustillos & M. Graff. Opinion mining and emotion recognition applied to learning environments. *Expert Systems with Applications* **150**, 113265 (2020).
8. M. S. Kim, & R. A. Calvo Sentiment analysis in student experiences of learning. In *Educational Data Mining*. (2010).
9. G. Xu, Y. Meng, X. Qiu, Z. Yu & X. Wu Sentiment Analysis of Comment Texts Based on BiLSTM. *IEEE Access* **7**, 51522–51532 (2019).
10. Z. H. Kilimci & S. Akyokus Deep Learning- and Word Embedding-Based Heterogeneous Classifier Ensembles for Text Classification. *Complexity* vol. **2018** e7130146 (2018).
11. A. Agrawal & A. An Unsupervised Emotion Detection from Text Using Semantic and Syntactic Relations. in *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology* 346–353 (IEEE, 2012).
12. C. O. Alm, D. Roth, & R. Sproat, Emotions from text: machine learning for text-based emotion prediction. *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*. 1220648 (2005).
13. Z. Teng, F. Ren & S. Kuroiwa Emotion Recognition from Text based on the Rough Set Theory and the Support Vector Machines. in *2007 International Conference on Natural Language Processing and Knowledge Engineering* 36–41 (IEEE, 2007).
14. N. Altrabsheh, M. Cocea & S. Fallahkhair Learning Sentiment from Students' Feedback for Real-Time Interventions in Classrooms. in *Adaptive and Intelligent Systems* (ed. Bouchachia, A.) vol. **8779** 40–49 (Springer International Publishing, 2014).
15. A. Ortigosa, J. M. Martín & R. M. Carro Sentiment analysis in Facebook and its application to e-learning. *Computers in Human Behavior* **31**, 527–541 (2014).
16. L. C. Yu, *et al.* Improving early prediction of academic failure using sentiment analysis on self-evaluated comments. *J Comput Assist Learn* **34**, 358–365 (2018).
17. J. Panget *al.* Fast Supervised Topic Models for Short Text Emotion Detection. *IEEE Trans. Cybern.* **51**, 815–828 (2021).
18. X. Hu, J. Tang, H. Gao & H. Liu Unsupervised sentiment analysis with emotional signals. in *Proceedings of the 22nd international conference on World Wide Web* 607–618 (Association for Computing Machinery, 2013).