ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/353607125

LSTM, VADER and TF-IDF based Hybrid Sentiment Analysis Model

Article *in* International Journal of Advanced Computer Science and Applications · July 2021

DOI: 10.14569/IJACSA.2021.0120730

CITATIONS
52

READS
2,195

4 authors, including:

Mohamed Chiny
Ibn Tofaïl University
13 PUBLICATIONS **120** CITATIONS
SEE PROFILE

Chihab Younes
Ibn Toufail University
46 PUBLICATIONS **209** CITATIONS
SEE PROFILE

Omar Bencharef
Cadi Ayyad University
85 PUBLICATIONS **591** CITATIONS
SEE PROFILE

All content following this page was uploaded by Chihab Younes on 31 July 2021.

The user has requested enhancement of the downloaded file.

LSTM, VADER and TF-IDF based Hybrid Sentiment Analysis Model

Mohamed Chiny[1], Marouane Chihab[2], Younes Chihab[4]
Laboratory of Computer Sciences
Ibn Tofail University
Kenitra, Morocco

Omar Bencharef[3]
Department of Computer Sciences
Cadi Ayyad University
Marrakesh, Morocco

***Abstract*—Most sentiment analysis models that use supervised learning algorithms consume a lot of labeled data in the training phase in order to give satisfactory results. This is usually expensive and leads to high labor costs in real-world applications. This work consists in proposing a hybrid sentiment analysis model based on a Long Short-Term Memory network, a rule-based sentiment analysis lexicon and the Term Frequency-Inverse Document Frequency weighting method. These three (input) models are combined in a binary classification model. In the latter, each of these algorithms has been implemented: Logistic Regression, k-Nearest Neighbors, Random Forest, Support Vector Machine and Naive Bayes. Then, the model has been trained on a limited amount of data from the IMDB dataset. The results of the evaluation on the IMDB data show a significant improvement in the Accuracy and F1 score compared to the best scores recorded by the three input models separately. On the other hand, the proposed model was able to transfer the knowledge gained on the IMDB dataset to better handle a new data from Twitter US Airlines Sentiments dataset.**

***Keywords*—Sentiment analysis; hybrid model; long short-term memory (LSTM); Valence Aware Dictionary and sEntiment Reasoner (VADER); term frequency-inverse document frequency (TF-IDF); classification algorithm**

I. Introduction

With the massive use of social networks such as Facebook, Twitter and Instagram, and dedicated platforms for sharing reviews and comments such as IMDB and Airbnb; it has become extremely difficult to track down published information, let alone extract relevant information such as reviews about a product or service, on the one hand, because of the abundance and variety of published data [1], and on the other hand because of the unstructured nature of the published texts, which makes it almost impossible to analyze them by classical computer methods [2].

The content produced by the social media community reflects one of the richest sources of data in terms of opinions and knowledge, and offers greater opportunities for businesses, governments, and society to extract valuable, expressive, and diverse knowledge, both in terms of the content itself and context-related knowledge [3]. Indeed, decision makers need to perceive how people feel about their services in order to improve the aspects that customers find unsatisfactory. Therefore, mining and analyzing the data left on these platforms with automated tools is crucial.

Sentiment analysis is a field of analysis that aims to determine the opinion and subjectivity of people's criticisms and attitudes towards entities and its attributes from unstructured written text [4]. A multitude of sentiment vocabulary analysis methods have been proposed over the past decades. As an example, based on the emotional attributes of words, Turny [5] used a simple unsupervised classification learning algorithm to compute pointwise mutual information to measure sentence sentiment polarity.

Wang et al. [6] proposed a topic-specific sentiment analysis method based on LSTM with attention mechanism, which focused on the features of different parts of the sentence through the attention mechanism, and achieved good performance on the task of classifying topic-specific sentiments. This work was conducted to address the problem that sentiment vocabulary generally changes with context information [7]. In [8] Pang et al. advocated for the first time the supervised learning model in sentiment classification, which performed significantly better than the traditional sentiment vocabulary-based parsing algorithms [9]. In addition, this study also pointed out that sentiment classification is more challenging than general classification tasks.

Although the models analyzed in the existing literature, which are characterized by the diversity of different features, improve performance that can be evaluated by metrics such as accuracy, Recall and F1-score, these supervised models have been trained on a large volume of data and, therefore, require a lot of labeled data, which is usually costly and leads to high labor cost in real-world applications [10,11].

On the other hand, the use of an intuitive lexicon-based classification does not work well, unlike a simple text classification. The reason is that among the overwhelming number of reviews, there are reviews that do not contain any intuitively subjective words and yet express a strong opinion. Other reviews contain very pejorative words and express a positive opinion (and vice versa) [12].

The idea of our work is to propose a sentiment analysis model that uses a low volume of labeled training data, while obtaining satisfactory results. Our approach is to combine three sentiment analysis models; the Long Short-Term Memory (LSTM) model, the Valence Aware Dictionary and sEntiment Reasoner (VADER) which is a rule-based sentiment analysis lexicon built on the wisdom of the crowd

and the Term Frequency-Inverse Document Frequency (TF-IDF) weighting based sentiment analysis model. Each of these three input models returns a sentiment positivity score in the text to be analyzed. Then we included a classification model where each of the following five algorithms has been implemented: Logistic Regression, k-Nearest Neighbors, Random Forest, Support Vector Machine and Naive Bayes. This classification model returns a binary result that indicates the sentiment experienced in the input text.

Our model improved the Accuracy, Recall, F-Score obtained by the three input models used individually (LSTM, VADER and TF-IDF). In addition, its evaluation on data from a different field than the one that provided the training data indicates that it was able to transfer the knowledge gained on an IMDB dataset to better handle a new Twitter US Airlines Sentiments dataset.

II. Literature Review

A. Recurrent Neural Network and Long Short-Term Memory

Recurrent neural networks (RNNs) are artificial neural networks that model the behaviors of dynamic systems using hidden states [13,14]. They have been the answer to most sequential data and natural language processing (NLP) problems for many years. This is because traditional neural networks take in a fixed amount of input data at a time and produce a fixed amount of output each time. In contrast, RNN do not consume all the inputs at once. Instead, they take them one at a time and in a sequence. At each step, the RNN performs a series of calculations before producing an output. The output, called a hidden state, is then combined with the next input in the sequence to produce another output. This process continues until the model is scheduled to terminate or the input sequence ends.

However, a major shortcoming that affects the typical RNN is the problem of gradient disappearance/explosion. This problem arises during backpropagation through the RNN during formation, especially for networks with deeper layers. For this reason, the LSTM was proposed by Sepp Hochreiter and Jürgen Schmidhuber in 1997 [15].

Long Short-Term Memory (LSTM) is a type of RNN architecture implementation that is faster and more accurate than standard RNN. Indeed, LSTM leads to many more successful executions and learns much faster. It also solves complex tasks that have never been solved by previous recurrent network algorithms and shows better performance for long range sequences than conventional RNN architectures [16,15].

LSTM has found its application in many fields that require sequential models, in this case NLP and especially in sentiment analysis. For example, Thomas et al. [17] modeled the LSTM neural network to find the sentiment of transliterated text that has become the language of social media websites such as WhatsApp, Facebook and Twitter. A transliterated dataset is collected using scrapping of different websites. A sample of 10,000 datasets was prepared. Two of the layers were created for training and testing the data. Their model was trained with 65 units and a learning rate of 0.01. This work was able to achieve an average accuracy of 0.8151.

In order to solve sentiment analysis problems and improve the execution time, Zhixing et al. [9] proposed a fast sentiment analysis algorithm, called FAST-BiLSTM. The algorithm is realized by merging FastText and Bi-LSTM models. First, FastText has a fast speed for linear fitting and can generate pre-trained word vectors as a by-product. Second, Bi-LSTM uses the generated word vectors for training and then merges with FastText to perform full sentiment analysis. The results show that the temporal efficiency of the algorithm is improved by more than 30% and that FAST-BiLSTM can sufficiently extract contextual semantic information from texts.

In a similar context, a new architecture is proposed by Soubraylu et al. [18] by combining long-term memory (LSTM) with word embedding to extract the semantic relationship between neighboring words, and weighted self-attention is also applied to extract key terms from reviews. Based on the experimental analysis of the IMDB dataset, the authors showed that the proposed word-embedded self-attention LSTM architecture achieved an F1 score of 88.67%, while the LSTM and word-embedding based LSTM models resulted in an F1 score of 84.42% and 85.69%, respectively. In [6], Wang et al. propose an LSTM that provides an attention mechanism to focus on different parts of the opinion sentence, given several aspects. Embedding of the aspect expression is taken into account with word sequence folding to assign attention weights with respect to a given aspect to each word.

In order to propose software to extract Business Intelligence from SA using a modified LSTM algorithm by having a different activation function. Sreesurya et al. [20] analyzed the data using LSTM machine learning approach, evaluating the sentiments on a scale from -100 to 100. A new proposed activation function is used for LSTM giving the best results compared to the existing artificial neural network (ANN) techniques. In [21], Dhanalakshmi et al. propose an analytics system that collects employee comments from open forums and performs sentiment analysis using the RNN-LSTM algorithm. In the sentiment analysis, the employee comments are classified as positive or negative so that the organization can identify the social sentiments of its brand and can take corrective actions to retain the employees. This paper also captures the performance of various models in training and predicting the employee feedback dataset and the models evaluated are logistic regression, support vector machine, random forest classifier, AdaBoost classifier, gradient amplification classifier, decision tree classifier and Gaussian Naive Bayes. The classification ratio and accuracy of each model are captured. When training the RNN-LSTM algorithm with a dataset of size 30k, the accuracy was 88%.

LSTM networks have also shown good performances in various domains such as meteorology [22], finance [23,24], medicine [25,26], image description generation [27,28], motion prediction in video sequences [29,30] and machine translation [31,32].

B. Valence Aware Dictionary and Sentiment Reasoner Lexicon and Rule-based Sentiment Analysis

The specific nature of social media content poses serious challenges to applications of sentiment analysis due to its huge bias and big data nature [33, 34]. Indeed, traditional methods

of textual sentiment analysis are mainly devoted to the study of extended texts, such as news stories and full documents. Microblogs are considered short texts that are often characterized by large noises, new words, and abbreviations. Previous emotion classification methods generally fail to extract meaningful features and produce a poor classification effect when applied to the processing of short texts or microtexts [35].

Valence Aware Dictionary and sEntiment Reasoner (VADER) is a rule-based lexicon and sentiment analysis tool that is specifically adapted to sentiments expressed in social media. VADER uses a sentiment lexicon which is a list of lexical features that are generally labeled based on their semantic orientation as positive or negative.

VADER is based on a wisdom of crowds (WotC) approach [36] to acquire a valid point estimate of the sentiment valence (intensity) of each lexical feature. The VADER evaluation was conducted by ten independent human raters (for a total of over 90,000 ratings), leading to the adoption of 7,500 lexical features with valence scores that indicate the polarity and intensity of sentiment on a scale of -4 (Extremely negative) to +4 (Extremely positive) [34]. This work has shown that VADER's performance exceeds even individual human raters.

VADER is sensitive to both the polarity and intensity (how positive or negative the sentiment is) of emotions, and it is adapted to the content of social networks that generally use informal writing (several punctuation marks, acronyms, emoticon, slang...). Indeed, some of the heuristics used by VADER to incorporate the impact of each subtext on the perceived intensity of the sentiment in the text are part of the writing style on social networks, in this case punctuation (such as the exclamation mark that increases the magnitude of the perceived intensity) and capitalization that emphasizes an important word for the sentiment in the presence of other non-capitalized words [34].

The fact that VADER is a pre-trained model gives it an advantage with respect to users. For example, Borg et al. [37] examine sentiment analysis among customers of a large Swedish telecommunications company. The dataset consists of 168010 emails with no sentiment information available. Therefore, the VADER model is used together with a Swedish sentiment lexicon to provide an initial labeling of the emails. It is after the labeling provided by VADER that the content is used to train two Support Vector Machine models in extracting and classifying the sentiment of the e-mails. In another work, Valdez et al. [38] analyzed the average daily sentiment of 86,581,237 U.S. time-series tweets with the VADER tool to understand what themes emerge from a corpus of U.S. tweets about COVID-19 and whether the sentiment changes in response to the pandemic. In [39], Al Mansoori et al. attempted to assess criminal behavior on Facebook and Twitter, and effectively classify the collected data as negative, positive, or neutral in order to identify a suspect by performing sentiment analysis using the VADER model. The VADER model was also used by Scholz et al [40] to perform an integrated semantic analysis to provide the sentiment of tweets retrieved between 2008 and August 2018 for the purpose of detecting tourism flows in the province of Styria in Austria.

C. Term Frequency-Inverse Document Frequency

Statistical approaches such as machine learning and deep learning work well with numerical data. However, natural language consists of words and sentences. Therefore, before a sentiment analysis model can be created, text must often be converted into numbers. For this purpose, several approaches have been developed, such as Bag of Words, N-grams, Word2Vec and TF-IDF.

The Term Frequency-Inverse Document Frequency (TF-IDF) algorithm [41, 42, 43] is used to evaluate the importance of words in a textual corpus. The importance is proportional to the number of times the words appear in the document and inversely proportional to the frequency of words appearing in the corpus. Indeed, in a simple Bag of Words, each word has the same importance. The idea behind TF-IDF is that words that appear more frequently in one document and less frequently in other documents should have more importance because they are more useful for classification.

TF represents the frequency of words, i.e. the number of times they appear in a corpus (Func 1). This consists in calculating the number of occurrences of the word out of the total number of words present in the corpus.

$$tf_i = \frac{n_i}{\sum_k n_k} \quad (1)$$

$$idf_i = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|} \quad (2)$$

$$tfidf_i = tf_i \, idf_i \quad (3)$$

IDF is the measure of the importance of the term in the whole corpus. It consists in calculating the logarithm of the inverse of the proportion of documents in the corpus that contain the term (Func 2). This consists in calculating the total number of documents contained in the corpus over the number of documents where the word is present. It is the logarithm of this result that constitutes the value of the IDF.

The TF-IDF weight is calculated by multiplying the two measures (Func 3). Thus, the higher the weight, the more significant the word in question is within the corpus.

The TF-IDF algorithm is often applied to texts for sentiment analysis. For example, Soumya et al. [44] performed sentiment analysis of Malayalam tweets using machine learning techniques. They used TF-IDF and Unigram with Sentiwordnet for training feature vectors of the input dataset, before classifying them using different techniques such as Naive Bayes (NB), Support Vector Machine (SVM) and Random Forest (RF). In [45], Ullah et al. proposed an algorithm and method for sentiment analysis using both text and emoticons. The two modes of data were analyzed in combination and separately with machine learning and deep learning algorithms to find sentiments from Twitter-based airline data using several features such as TF-IDF, N-gram and emoticon lexicons. On the other hand, Ayo et al. [46] adopted an approach that proposes an improved hybrid integration with a topic inference method and an improved

neural network for hate speech detection in Twitter data. The proposed method uses a hybrid nesting technique that includes TF-IDF for word-level feature extraction and LSTM long-term memory for sentence-level feature extraction.

III. Architecture of our Hybrid Sentiment Analysis Model

The objective of our study is to build a hybrid sentiment analysis model (Fig. 1) that is based on three input models:

- A model based on the use of LSTM layer and which was trained on a corpus of labeled IMDB reviews.
- The VADER lexicon which is a pre-trained model based mainly on the wisdom of the crowd.
- A TF-IDF model that takes into account the importance of words in the text to estimate the sentiment. This model was also trained on the same dataset as the LSTM model.

The scores calculated by these three models are then combined in a classification model that returns whether the sentiment of the input occurrence is positive or negative.

A. LSTM Model

LSTM is a class of powerful neural networks for modeling sequence data such as time series or natural language. An optimal use of LSTM layer requires the preparation of the text to be analyzed. This preparation consists of cleaning and filtering, followed by tokenization, then word embedding. The vector representation of the words in the sentence is the input to our LSTM model which uses Softmax as an activation function to produce a multi-class categorical probability distribution and the Cross Entropy loss function.

1) Cleaning and filtering: Once the sentence to be evaluated is available at the input of our model, it is first cleaned in order to eliminate all occurrences that may bias the subsequent processing, such as multiple spaces or spurious characters like excessive successions of punctuation marks.

The filtering operation was also carried out on the data used to train and test our model. This is an IMDB dataset containing 50,000 movie reviews for natural language processing, text analysis or binary sentiment classification [47].

2) Tokenization: Tokenization is a process used to divide text into single words (unigram) or combinations of successive words (n-gram). This operation also creates an index mapping dictionary using the vocabulary of all the words in the model training text.

The N-gram model is widely used in computational linguistics to predict the next element in such a contiguous sequence of n elements from a particular sample of text. However, in our case, and in order to use the GloVe model, the text has been divided into one-word tokens.

The resulting sequences have different lengths, and in order to handle both short and long criticisms, it is preferable that all entries have the same length. This length has been defined as the sequence length. This sequence length is identical to the number of time steps for the LSTM layer and is the maximum length calculated for a comment in the training corpus (1744 tokens).

3) Word Embedding with GloVe: Word embedding is a class of approaches for representing words using a dense vector representation. It is an improvement over traditional bag-of-words model coding schemes which consist in marking each word in a vector to represent an entire vocabulary. Since the latter is vast, then a given word will be represented by a large vector consisting mostly of null values.

Semantic vector space models of the language represent each word with a real-valued vector. Vectors can be used as features in various applications, such as document classification [48] or named entity recognition [49]. Indeed, Word embedding improves text classification by solving the sparse matrix and word semantics problem.

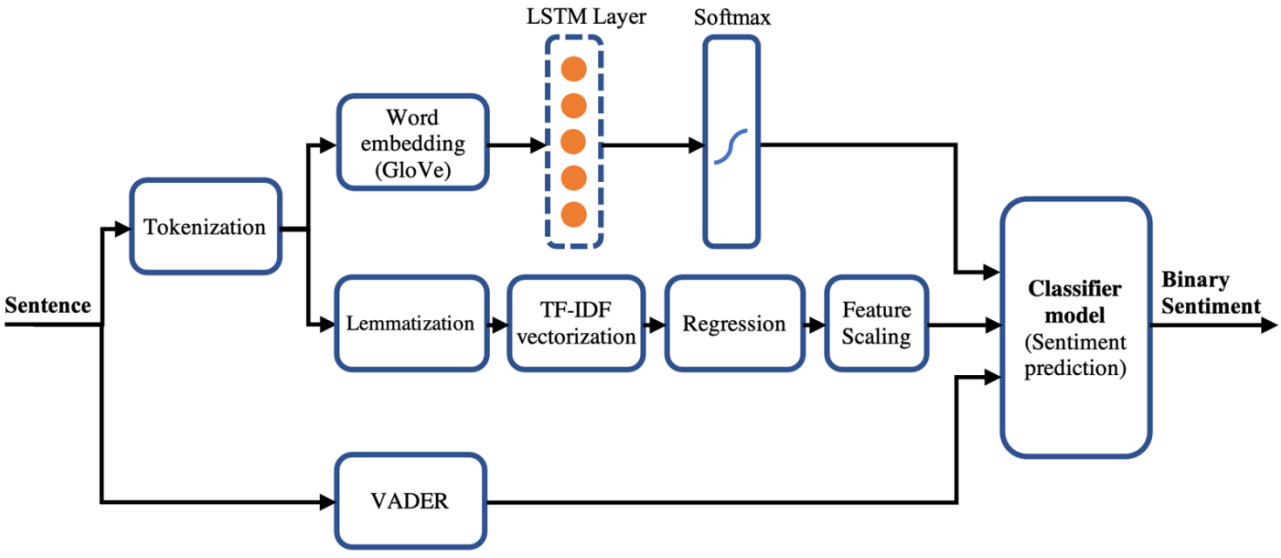

Fig. 1. Proposed LSTM, VADER and TF-IDF based Hybrid Sentiment Analysis Model.

The two most common word integrations are: Word2Vec and GloVe.However, GloVe (Global Vectors for Word Representation), as its name suggests, is better at preserving global contexts because it creates a global co-occurrence matrix by estimating the probability that a given word co-occurs with other words.

GloVe is an unsupervised learning algorithm for obtaining vector representations of words. Training is performed on global word-word co-occurrence statistics aggregated from a corpus, and the resulting representations have linear substructures of the word vector space [50]. 100-dimensional GloVe integrations of 400,000 calculated words were used.

4) LSTM layer: When defining the LSTM layer, 256 hidden units have been fixed. This layer is linked to a Softmax activation function. The Adam optimizer, which is one of the methods that compute the learning rate, known to work well in practice, and compares favorably with other adaptive learning algorithms has been used (Table I).

5) Softmax layer: The softmax function is a function that transforms a vector of K real values into a vector of K real values that sum to 1. Whatever the values of the input, the softmax transforms them into values between 0 and 1, so that they can be interpreted as probabilities.

Softmax is a generalization of logistic regression that can be used for multi-class classification. Many multi-layer neural networks end with a penultimate layer that produces real-valued scores that are not properly scaled and can be difficult to work with. Here, the softmax is very useful because it converts the scores to a normalized probability distribution.

In our case, the softmax layer outputs two probability scores that correspond to the positivity and negativity of the input sequence.

Training and evaluation of the LSTM model

From the 50,000 reviews available in the dataset, 5,000 reviews were selected from the train set and 2,000 from the test set of our LSTM model. We checked that the number of positive reviews and the number of negative reviews in the dataset were balanced. Most of these reviews consist of several hundred words, and some reviews exceed a thousand words. The average number of words used in the reviews in the dataset is 1309.

TABLE I. HYPER-PARAMETERS OF THE LSTM MODEL

Hyper-parameter	Value
Input vocab size	1744
Output embedding dimension	100
LSTM layer internal units	256
Optimizer	Adam
Loss	Categorical Crossentropy
Activation function	Softmax

B. VADER Lexicon

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a rule-based lexicon and sentiment analysis tool that is specifically adapted to sentiments expressed in social media. VADER uses a combination of a sentiment lexicon and a list of lexical features (e.g., words) that are generally labeled according to their semantic orientation as positive or negative.

VADER produces four measures of sentiment from these word ratings. The first three, positive, neutral, and negative, represent the proportion of text that falls into these categories. The final metric, the composite score, is the sum of all lexicon scores that have been normalized between -1 and 1.

It should be recalled that the VADER model is sensitive to punctuation and capitalization [34]. Therefore, special characters have not been filtered out and text has not been converted to lowercase in order to capture the full sentiment.

Evaluation of the VADER model

The VADER model was evaluated on the same test dataset used for the evaluation of the LSTM model (2000 opinion). The composite score was retained after normalizing it between 0 and 1.

C. TF-IDF Model

The TF-IDF approach is used to create numerical feature vectors from text. It is a method very often used in text classification that gives information about the occurrence of words.

1) Data pre-processing: As for the LSTM model, the targeted sentence is first cleaned to eliminate all unnecessary occurrences such as multiple spaces, strings of numbers or URLs... Then all the text is converted to lowercase so as not to have 2 different dimensions for the same word at the time of vectorization

Stopwords have also been removed from the text. These are very common words in the studied language that do not bring any informative value for the understanding of the meaning of a document and corpus. In addition, they are very frequent and are part of the common vocabulary, which has the effect of significantly impacting the speed of the processing that follows.

2) Tokenization and lemmatization: The same tokenization module used for the LSTM model is used for the TF-IDF model, i.e. the input texts have been segmented into tokens of one word each (unigram) before being lemmatized.

Lemmatization refers to a lexical treatment of a text in order to analyze it. Stemming and lemmatization refer to text normalization in the field of natural language processing and are widely used in text mining.

The difference between stemming and lemmatization is that stemming simply removes the last characters, which often leads to incorrect meanings and sometimes even misspellings, whereas lemmatization considers the context and converts the word into its canonical form recorded in the dictionaries of the relevant base language.

For our model, WordNet lemmatizer which uses the WordNet repository database to search for word lemmas has been used. Indeed, Wordnet [51] is a large lexical database, freely and publicly available for the English language, aiming at establishing structured semantic relations between words. Nouns, verbs, adjectives and adverbs are grouped into cognitive synonym sets (synsets), each expressing a distinct concept. The majority of WordNet's relationships connect words from the same Part Of Speech (POS). Among the features it offers is one of the oldest and most commonly used lemmatizers.

3) TF-IDF vectorization: Unlike a Bag of Words (BoW) which converts text into a feature vector by counting the occurrence of words in a document without considering their importance, TF-IDF is based on the Bag of Words (BoW) model, which contains information about the most important and least important words in a document.

In order to convert a collection of raw documents into a TF-IDF feature matrix, a vocabulary which only considers the first 500 terms classified by term frequency in the corpus has been built, and then removed terms that appear too frequently (in more than 50% of documents) or infrequently (in less than 7 documents) (Table II). This allows us to ignore words that have very few occurrences to be considered significant, or conversely, too frequent in the corpus.

4) Linear regression: Regression is a method of modeling a variable (called target) as a function of independent predictors (called features), where the algorithm involved tries to find causal relationships between the variables [52].

Since the TF-IDF feature matrix contains 500 dimensions, and each of these dimensions represents a relevance score of each word ($tfidf_i$), our goal is to establish a regression model (Func 4) that will allow us to compute the relative weights (β_i) to the 500 most significant words in the corpus with respect to the sentiment score ($Score_i$).

$$Score_i = \sum_{i=1}^{500} \beta_i \, tfidf_i \quad (4)$$

In this kind of application (sentiment analysis), it is rather classification models that are used and not regression models. However, the objective is not to calculate a binary score, but a continuous value (like the scores calculated with the LSTM and VADER models). These three scores will constitute the inputs of the final classification model (Fig. 1).

In order to train the regression model, the same dataset as the one used for training the LSTM model (5000 reviews) was used.

TABLE II. HYPER-PARAMETERS OF THE TF-IDF MODEL

Hyper-paramter	Value
max_features	500
min_df	7 documents
max_df	50% of documents

5) Feature scaling: Many machine learning algorithms work better or converge faster when the features are relatively similar in scale and close to the normal distribution, including linear regression. However, the output of our regression model gives prediction values outside the interval [0,1] (unlike the LSTM and VADER models). In order to help the features arrive in a more suitable form to the Classifier model, normalization of the prediction values of this output was performed.

Evaluation of the TF-IDF model

The TF-IDF model was evaluated on the same test set used to evaluate the LSTM and VADER models (2000 reviews). The evaluation scores of the three models (LSTM, VADER and TF-IDF) are used as reference values to compare them to the scores of our proposed architecture in this study.

D. CLASSIFIER Model

We recall that our objective is to combine the 3 models of sentiment analysis of the input with a classification model in order to improve the performance of predictions on the sentiment conveyed through the input text. Indeed, the LSTM model, which is part of the RNN, is distinguished by its ability to adapt to sequential data. The VADER model has proven its efficiency in the microblogging domain. Finally, the TF-IDF model is characterized by its ability to handle the most significant words in a document. A higher Accuracy and F1 scores than those obtained by the three models used separately on the same data is expected. We also recall that the LSTM and TF-IDF models have been trained on IMDB review texts, while VADER is a pre-trained model.

Our classification model contains three inputs that are directly related to the outputs of the LSTM, VADER and TF-IDF models. The values of these inputs are continuous in a range of [0,1] and the output of the classification model returns a binary result (positive or negative) which is the prediction of the sentiment of the text of the full model input (Fig. 1).

5000 random reviews have been selected from the dataset that are different from the training set and test set data used for the LSTM and TF-IDF models. We ran them through the input of our global model to obtain the predictions computed by the LSTM, VADER and TF-IDF models. Then we divided these results into two batches (75% for the train set and 25% for the test set), in order to train and evaluate our binary classification model, implementing each of the following five classification algorithms: Logistic Regression (LR), k Nearest Neighbors (k-NN), Random Forest (RF), Support Vector Machine (SVM) and Naive Bayes (NB).

The hyper-parameters of each of the classification algorithms used were manipulated to have the best possible evaluations for our data set. Table III gives an overview of the most important hyperparameters that were applied to our classification models.

TABLE III. MOST IMPORTANT HYPER-PARAMETERS APPLIED TO THE ALGORITHMS IMPLEMENTED IN THE CLASSIFICATION MODEL

Algorithme	Hyper-parametre	Value
Logistic Regression	Inverse of regularization strength	1
k Nearest Neighbors	Number of neighbors	13
Random Forest	Number of trees in the forest Maximum depth of the tree	19 4
Support Vector Machine	Regularization parameter Kernel type	1 Linear
Naive Bayes	Var smoothing	1e-09

IV. RESULTS

A. Evaluation of the Binary Classification Model

The binary classification model is the block that returns the final result of the sentiment experienced in the input text of the full model. Its three inputs come from the three input models (LSTM, VADER and TF-IDF). Table IV lists the Accuracy of each classification model following its evaluation on the test data.

TABLE IV. ACCURACY EVALUATED FOR THE ALGORITHMS IMPLEMENTED IN THE CLASSIFICATION MODEL

Model	Accuracy
Logistic Regression	0.888
k Nearest Neighbors	0.868
Random Forest	0.876
Support Vector Machine	0.884
Naive Bayes	0.868

B. Evaluation of our Model with IMDB Dataset Data

In the following, the results obtained using the proposed architecture will be exposed. In order to better identify the performance improvement that it has allowed, the complete model was evaluated on the same test set that evaluated the LSTM, VADER and TF-IDF models separately.

Fig. 2 shows mean micro-averaged for the model by implementing the five different algorithms in the model Classifier. Table I shows that after training the LSTM model, its evaluation on the test set gave an accuracy of 0.829 and an F1 score of 0.835. As for the VADER model (which is a pre-trained model), its evaluation on the same testset gave an accuracy of 0.723 and an F1 Score of 0.766. With the TF-IDF model, an accuracy of 0.789 and an F1 score of 0.792 have been obtained. Between these three basic models, it turns out that the LSTM model shows higher scores in terms of accuracy, Recall and F1 score.

Table V displays the performance metrics (Accuracy, Recall and F1 score) of the 3 input models (LSTM, VADER and TF-IDF) on the IMDB test data.

After training and evaluating our model using the five proposed classification algorithms, the performance metrics shown in Table VI have been obtained.

The evaluation scores obtained using our model is different depending on the classification algorithm used. However, whatever the algorithm, the scores are better than those obtained using the three models LSTM, VADER and TF-IDF separately, except for the F1 scores obtained using Random Forest (0.83) which is slightly lower than the F1 scores obtained using the LSTM model (0.835), but higher than the F1 scores obtained using VADER and TF-IDF (respectively 0.766 and 0.792).

The average Accuracy obtained using our model using the 5 classification algorithms separately (0.854) is 9.517% higher than the average Accuracy obtained using the three models LSTM, VADER and TF-IDF (0.780%), and the average F1 score obtained using the full model (0.856) is 7.363% higher than that obtained using the three models separately (0.797) (Table VI).

It should be noted that the Logistic Regression model offers a better Accuracy (0.878) compared to the accuracy obtained with the three models LSTM, VADER and TFIDF (respectively 0.829, 0.723 and 0.789), i.e. 5.91% higher than the accuracy obtained with LSTM which is the best score recorded among the 3 initial models. Logistic Regression also offers a better F1 score (0.881) which is 5.51% higher than the F1 score of LSTM (0.835) (Table VII).

TABLE V. PERFORMANCE METRICS EXPECTED BY THE THREE INPUT MODELS ON IMDB DATA

Model	Accuracy	Recall	F1 Score
LSTM	0.829	0.827	0.835
VADER	0.723	0.675	0.766
TF-IDF	0.789	0.803	0.792

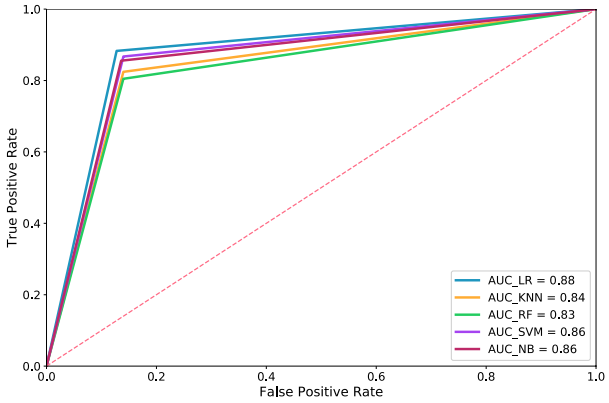

Fig. 2. AUC of the Proposed Model by implementing the different Classification Algorithms when Predicting Sentiment on IMDB Data.

TABLE VI. PERFORMANCE METRICS ACHIEVED BY OUR MODEL BY IMPLEMENTING THE 5 CLASSIFICATION ALGORITHMS ON IMDB DATA

Model	Accuracy	Recall	F1 Score
Logistic Regression	0.878	0.882	0.881
k Nearst Neighbors	0.841	0.824	0.842
Random Forest	0.831	0.804	0.830
Support Vector Machine	0.863	0.867	0.867
Naive Bayes	0.860	0.855	0.862

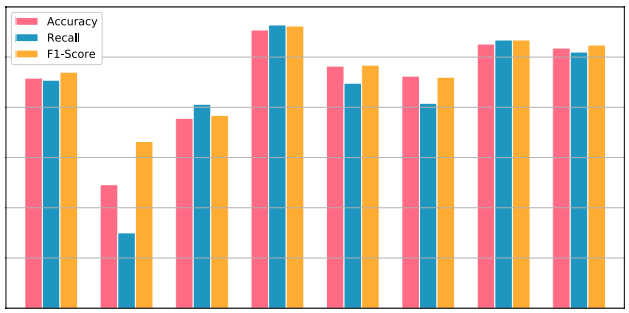

Fig. 3. Representation of the Evaluation Metrics on IMDB Data.

Overall, Fig. 3 shows that our model gave the best performance using the Logistic Regression, k-NN, SVM and Naive Bayes models. The Random Forest model on the other hand gave a slightly lower F1 score than the LSTM model, but still outperformed VADER and TF-IDF.

TABLE VII. RELATIVE IMPROVEMENT RECORDED BY EACH CLASSIFICATION ALGORITHM COMPARED TO THE BEST SCORES DISPLAYED BY THE INPUT MODELS ON THE IMDB DATA

Model	Accuracy improvement	F1 Score improvement
Logistic Regression	+5.91%	+5.51%
k Nearst Neighbors	+1.44%	+0.83%
Random Forest	+0.24%	-0.59%
Support Vector Machine	+4.10%	+3.83%
k-Means	+3.74%	+3.23%
Mean	+9.51%	+7.36%

C. Evaluation of our Model with Data from the Twitter Dataset

The proposed model has also been evaluated on a US Airlines Sentiments Twitter dataset available on Kaggle [53]. This is a set of labeled tweets that was posed as a binary classification problem. The dataset contains 14427 unique texts that were used as a test set for our models.

It should be noted that the structure of the data encompassed in this dataset is different from that of the IMDB movie review dataset. On the one hand, the tweets contain text that is too short (with an average of 104 words, compared to 1309 words for the IMDB reviews), and on the other hand, due to the nature of the topic being reviewed, the vocabulary used most likely contains words that our LSTM and TF-IDF models never saw during training.

Fig. 4 shows mean micro-averaged for the model by implementing the five different algorithms in the model Classifier. Obviously, the performance of the LSTM and TF-IDF models has dropped considerably. Indeed, the accuracy score and the F1 score of the LSTM model are respectively 0.66 and 0.67. For the TF-IDF model, these scores are respectively 0.667 and 0.637. On the other hand, the VADER model showed almost the same scores as for the IMDB data (Table VIII).

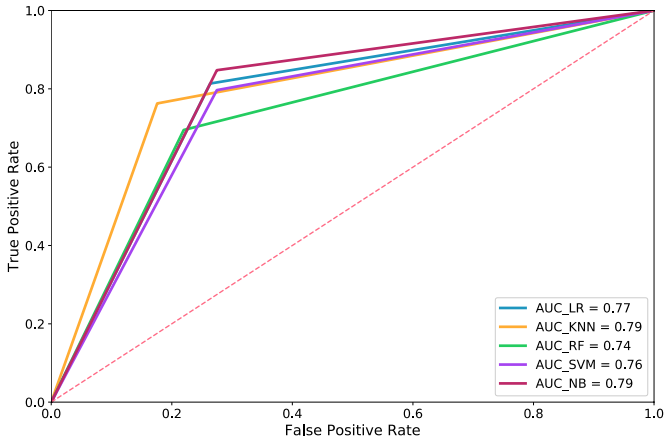

Fig. 4. AUC of the Proposed Model by Implementing the different Classification Algorithms when Predicting the Sentiment on the Twitter US Airlines Sentiments Data.

TABLE VIII. PERFORMANCE METRICS DISPLAYED BY THE 3 INPUT MODELS ON THE US AIRLINES SENTIMENTS TWITTER DATA

Model	Accuracy	Recall	F1 Score
LSTM	0.660	0.541	0.670
VADER	0.720	0.591	0.723
TF-IDF	0.667	0.556	0.637

However, the Accuracy score of our model remains higher than that of the LSTM, TF-IDF and VADER models (0.767, 0.8, 0.747, 0.754 and 0.773 for Logistic Regression, k Nearest Neighbors, Random Forest, SVM and Naive Bayes respectively). The same is true for the F1 score of the three models Logistic Regression, k-NN and Naive Bayes which are respectively 0.733, 0.75 and 0.746 (Table IX).

The average Accuracy obtained using our model using the five classification algorithms separately is 12.58% higher than the average Accuracy obtained using the three models LSTM, VADER and TF-IDF, and the average F1 score obtained using the full model is 7.26% higher than that obtained using the three models separately (Table X).

If the VADER model which displayed the best scores (Accuracy=0.72 and F1 score=0.723) is taken as a reference, then we can notice that the proposed model recorded an improvement in accuracy and F1 score (respectively 11.11% and 3.73%) using the k-NN algorithm.

TABLE IX. PERFORMANCE METRICS ACHIEVED BY OUR MODEL BY IMPLEMENTING THE FIVE CLASSFICIATION ALGORITHMS ON THE TWITTER US AIRLINES SENTIMENTS DATA

Model	Accuracy	Recall	F1 Score
Logistic Regression	0.767	0.813	0.733
k Nearst Neighbors	0.800	0.762	0.750
Random Forest	0.747	0.695	0.683
Support Vector Machine	0.754	0.797	0.717
Naive Bayes	0.773	0.847	0.746

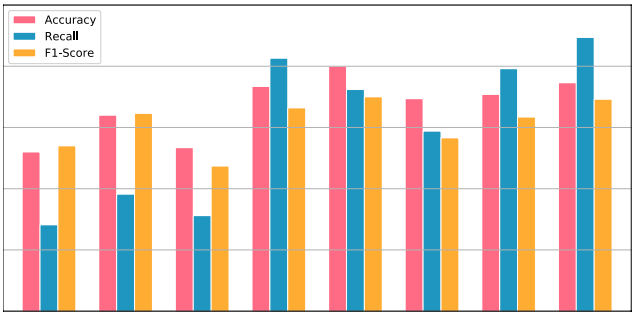

Fig. 5. Representation of Evaluation Metrics on US Airlines Sentiments Twitter Data.

TABLE X. RELATIVE IMPROVEMENT RECORDED BY EACH CLASSIFICATION ALGORITHM COMPARED TO THE BEST SCORES DISPLAYED BY THE INPUT MODELS ON THE DATA AFTER THE EVALUATION OF OUR MODEL ON THE TWITTER US AIRLINES SENTIMENTS DATA

Model	Accuracy improvement	F1 Score improvement
Logistic Regression	+6.52	+1.38
k Nearest Neighbors	+11.11	+3.73
Random Forest	+3.75	-5.53
Support Vector Machine	+4.72	-0.83
Naive Bayes	+7.36	+3.18
Mean	+12.58	+7.26

Fig. 5 clearly illustrates that our model performed well (compared to the three input models), in this case if our classifier model implements the Logistic regression, k-NN and Naive Bayes algorithms.

V. DISCUSSION

According to the results obtained, the proposed model shows better performances in terms of accuracy and F1 score, and which can exceed the performances of the best among the three input models (LSTM, VADER and TF-IDF) by 5.91% for accuracy and 5.51% for F1 Score. This peak was obtained by implementing the Logistic Regression algorithm in our Classifier model and by evaluating our model on the IMBD dataset, knowing that the training data also comes from this same dataset.

On the other hand, when the proposed model has been evaluated using the Twitter US Airlines Sentiments dataset, the performance obviously decreased, but it remains globally more advantageous than those obtained using the three input models. Indeed, we were able to record a higher accuracy score of 11.11% and a higher F1 score of 3.73% using the k-NN algorithm. These comparisons were made with respect to the highest Accuracy and F1 scores that were displayed by the VADER model (0.72 and 0.723, respectively).

It would be useful to recall that the structure of the US Airlines Sentiments Twitter dataset is different from the IMDB movie review dataset in terms of text size and vocabulary used. However, our model managed to display better scores compared to the three input models (LSTM, VADER and TF-IDF).

This improvement could be explained by the combination of the different techniques used in the three input models. Indeed, LSTM is more adapted to sequential data such as time series, speech and text [16]. VADER is a pre-trained lexicon focused on the wisdom of the crowd and mainly adapted to microblog data [34]. TF-IDF, on the other hand, takes into account the presence of the most significant words in a textual corpus [41]. We can therefore conclude that the combination of these three basic models through a classification model has allowed this performance improvement by capturing each of the different features of the input text according to their operating mode.

On the other hand, considering that most machine learning algorithms are based on the assumption that the training dataset and the test dataset belong to the same descriptor space and follow the same probability distribution [19], our model was able to transfer the knowledge gained on an IMDB dataset to better process a new US Airlines Sentiments Twitter dataset. Although the scores obtained are not very high, they are still much better than those obtained by the LSTM, VADER and TF-IDF models separately.

VI. CONCLUSION

The content created by users of social media (such as Twitter, Facebook or Instagram) and dedicated platforms (such as IMDB or Airbnb) reflects one of the richest sources of data in terms of opinions and knowledge. The data they encompass offers great opportunities for companies to extract valuable and expressive knowledge. For this reason, a field like sentiment analysis, which seeks to determine the opinion and subjectivity of people's reviews from unstructured written text, is growing rapidly.

Although for more than a decade, many sentiment analysis models have been proposed, they are generally data-intensive and computationally expensive. Indeed, most of these models generally require a huge amount of training data to achieve satisfactory performance metrics, namely, accuracy and F1 score.

The objective of our study is to propose a hybrid sentiment analysis model based on three basic models, namely, LSTM, VADER and TF-IDF. Each of these models captures different specifications of the same text. These models are then combined in a classification model where each of the following five algorithms has been implemented: Logistic Regression, k-Nearest Neighbors, Random Forest, Support Vector Machine and Naive Bayes. The output of our model delivers a binary score that reflects the sentiment of the input text. The proposed model was trained on 5000 IMDB movie reviews and then evaluated on other reviews from the same dataset, then it was evaluated on Twitter US Airlines Sentiments which has a different structure in terms of text size and vocabulary used.

The results suggest that, depending on the classification algorithm implemented, our model displays higher Accuracy and F1 scores than those achieved by the three basic models. Indeed, with Logistic Regression, an improvement of 5.91% for the Accuracy and 5.51% for the F1 Score on the evaluation data of the IMBD dataset has been noted. These scores were

calculated with respect to the best performances achieved by the three basic models.

After evaluating our model on the US Airlines Sentiment Twitter data, an overall decrease in performance has been noted. However, the performance of the model is still much higher than those recorded by the three basic models. Indeed, we were able to record a higher accuracy score of 11.11% and a higher F1 score of 3.73% using the k-NN algorithm, which indicates that our model was able to transfer the knowledge acquired on an IMDB dataset to better process a new US Airlines Sentiments Twitter dataset.

As a perspective, it would be interesting to improve the proposed model by implementing a BiLSTM model based on self-attention in order to capture the polarity of a whole sentence that may contain several term-aspects. Such an improvement would have a significant impact on the evaluation metrics, namely the accuracy and the F1 Score.

REFERENCES

[1] Sepideh Bazzaz Abkenar, Mostafa Haghi Kashani, Ebrahim Mahdipour, Seyed Mahdi Jameii. Big data analytics meets social media: A systematic review of techniques, open issues, and future directions. Telematics and Informatics, 101517, 2020. doi:10.1016/j.tele.2020.101517.

[2] Soubraylu Sivakumar, Ratnavel Rajalakshmi. Analysis of Sentiment on Movie Reviews Using Word Embedding Self-Attentive LSTM, International Journal of Ambient Computing and Intelligence (IJACI)12(2), 2021.

[3] M. Baumgarten, Maurice Mulvenna, N. Rooney, J. Reid. Keyword-Based Sentiment Mining using Twitter, International Journal of Ambient Computing and Intelligence, 2015.

[4] Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: Capturing favorability using natural language processing. In Proceedings of the 2nd international conference on Knowledge capture, pages 70–77. ACM, 2003.

[5] P. D. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews, 40th Annual Meeting on Association for Computational Linguistics, Stroudsburg, USA, 2002.

[6] WANG Y., HUANG M., ZHU X. & ZHAO L. Attention-based LSTM for aspect-level sentiment classification. In J. SU, X. CARRERAS & K. DUH, Eds., Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 2016, p. 606–615 : The Association for Computational Linguistics.

[7] T. Mike, B. Kevan, P. Georgios, C. Di and K. Arvid. Sentiment in short strength detection informal text. Journal of the American Society for Information Science & Technology, vol. 61(12), 2010.

[8] B. Pang, L. Lee and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques, Conf. on Empirical Methods in Natural Language Processing, Brussels, Belgium, 2018.

[9] Zhixing Lin, Like Wang, Xiaoli Cui, Yongxiang Gu. Fast Sentiment Analysis Algorithm Based on Double Model Fusion, Computer Systems Science & Engineering, 2020.

[10] F. Koto and M. Adriani. HBE: Hashtag-based emotion lexicons for twitter sentiment analysis, 7th Forum for Information Retrieval Evaluation, New York, USA, 2015.

[11] X. C. Huang, Y. H. Rao, H. R. Xie, T. L. Wong and F. L. Wang. Cross-domain sentiment classification via topic- related TrAdaBoost, Association for the Advancement of Artificial Intelligence Conf, San Francisco, USA, 2017.

[12] Anaïs Collomb, Crina Costea, Damien Joyeux, Omar Hasan, Lionel Brunie. A Study and Comparison of Sentiment Analysis Methods for Reputation Evaluation, Laboratoire d'InfoRmatique en Image et Systèmes d'information, 2013.

[13] K.-i. Funahashi and Y. Nakamura. Approximation of dynamical systems by continuous time recurrent neural networks, Neural networks, vol. 6, no. 6, pp. 801–806, 1993.

[14] A. Delgado, C. Kambhampati, K. Warwick. Dynamic recurrent neural-network for system-identification and control. IEE Proceedings – Contro, 1995.

[15] S. Hochreiter, J. Schmidhuber. Long Short-Term Memory. Neural Computation, 9(8), 1735–1780, 1997. doi:10.1162/neco.1997.9.8.1735.

[16] Y. J. Kim, S. Choi, S. Briceno, D. Mavris. A deep learning approach to flight delay prediction. 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), 2016. doi:10.1109/dasc.2016.7778092.

[17] M. Thomas, C. A. Latha. Sentimental analysis of transliterated text in Malayalam using recurrent neural networks. Journal of Ambient Intelligence and Humanized Computing, 2020. doi:10.1007/s12652-020-02305-3.

[18] Soubraylu Sivakumar, Ratnavel Rajalakshmi, Analysis of Sentiment on Movie Reviews Using Word Embedding Self-Attentive LSTM, International Journal of Ambient Computing and Intelligence (IJACI)12(2), 2021.

[19] Matthew E. Taylor, Peter Stone. Transfer Learning for Reinforcement Learning Domains: A Survey, Journal of Machine Learning Research 10, 2009.

[20] I. Sreesurya, H. Rathi, P. Jain, T. K. Jain. Hypex: A Tool for Extracting Business Intelligence from Sentiment Analysis using Enhanced LSTM. Multimedia Tools and Applications, 2020. doi:10.1007/s11042-020-08930-6.

[21] R. Dhanalakshmi and T. Sri Devi. Adaptive cognitive intelligence in analyzing employee feedback using LSTM, Journal of Intelligent & Fuzzy Systems, 2020.

[22] S. X. Jian, C. Z. Rong, W. Hao, Y. D. Yan, W. W. Kin. Convolutional LSTM network: A machine learning approach for precipitation nowcasting, in Proc. Advances in Neural Information Processing Systems, Montreal, Canada, pp. 802–810, 2015.

[23] Xuan Zhang, Xun Liang, Aakas Zhiyuli, Shusen Zhang, Rui Xu and Bo Wu. AT-LSTM: An Attention-based LSTM Model for Financial Time Series Prediction, 2019 IOP Conf. Ser.: Mater. Sci. Eng. 569 052037.

[24] Jungsik Hwang. Modeling Financial Time Series using LSTM with Trainable Initial Hidden States, arXiv:2007.06848 [q-fin.ST].

[25] Trang Pham, Truyen Tran, Dinh Phung, Svetha Venkatesh. Predicting healthcare trajectories from medical records: A deep learning approach, Journal of Biomedical Informatics,Volume 69, May 2017.

[26] Jing Xia, Su Pan, Min Zhu, Guolong Cai, Molei Yan, Qun Su, Jing Yan, Gangmin Ning. A Long Short-Term Memory Ensemble Approach for Improving the Outcome Prediction in Intensive Care Unit, Computational and Mathematical Methods in Medicine, 2019.

[27] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In CVPR, 2015.

[28] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In ICML, 2015.

[29] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. arXiv preprint arXiv:1412.6604, 2014.

[30] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. In ICML, 2015.

[31] Sutskever, I., Vinyals, O., and Le, Q. (2014). Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems (NIPS 2014).

[32] Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua. Neural machine translation by jointly learning to align and translate. arXiv:1409.0473, September 2014.

[33] Parul Pandey, Simplifying Sentiment Analysis using VADER in Python (on Social Media Text), https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f (2018).

[34] C.J. Hutto, Eric Gilbert. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text, Conference: Proceedings

of the Eighth International AAAI Conference on Weblogs and Social Media At: Ann Arbor, MI, 2015.

[35] D. Xu, Z. Tian, R. Lai, X. Kong, Z. Tan, W. Shi. Deep Learning Based Emotion Analysis of Microblog Texts. Information Fusion, 2020. doi:10.1016/j.inffus.2020.06.002.

[36] James Surowiecki. The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations. Random House Large Print, 2004.

[37] A. Borg, M. Boldt. Using VADER Sentiment and SVM for Predicting Customer Response Sentiment. Expert Systems with Applications, 2020.

[38] Valdez Danny, Ten Thij Marijn, Bathina Krishna, Rutter Lauren A., Bollen Johan. Social Media Insights Into US Mental Health During the COVID-19 Pandemic: Longitudinal Analysis of Twitter Data, JOURNAL OF MEDICAL INTERNET RESEARCH, 2020.

[39] Al Mansoori Saeed, Almansoori Afrah, Alshamsi Mohammed, SalloumSaid A. Suspicious Activity Detection of Twitter and Facebook using Sentimental Analysis. TEM JOURNAL-TECHNOLOGY EDUCATION MANAGEMENT INFORMATICS, 2020.

[40] J. Scholz, J. Jeznik. Evaluating Geo-Tagged Twitter Data to Analyze Tourist Flows in Styria, Austria. ISPRS International Journal of Geo-Information, 9(11), 681, 2020. doi:10.3390/ijgi9110681.

[41] G. Kang, M. Tang, J. Liu, X. Liu, and B. Cao "Diversifying web service recommendation results via exploring service usage history", IEEE Transactions on Services Computing, vol. 9, 2016.

[42] A. Guo and T. Yang. Research and improvement of feature words weight based on TF-IDF algorithm. IEEE Information Technology, Networking, Electronic and Automation Control Conference, pp. 415–419, Chongqing, China, 2016.

[43] Shengqi Wu, Huaizhen Kou, Chao Lv, Wanli Huang, Lianyong Qi, Hao Wan. Service Recommendation with High Accuracy and Diversity, Wireless Communications and Mobile Computing Volume 2020.

[44] Soumya S, Pramod K. V. Sentiment analysis of malayalam tweets using machine learning techniques. ICT Express, 2020. doi:10.1016/j.icte.2020.04.003.

[45] M. A. Ullah, S. M. Marium, S. A. Begum,N. S. Dipa. An algorithm and method for sentiment analysis using the text and emoticon. ICT Express, 2020. doi:10.1016/j.icte.2020.07.003.

[46] F. E. Ayo, O. Folorunso, F. T. Ibharalu, I. A. Osinuga. Hate speech detection in Twitter using hybrid embeddings and improved cuckoo search-based neural networks. International Journal of Intelligent Computing and Cybernetics, 2020. doi:10.1108/ijicc-06-2020-0061.

[47] Kaggle. https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews, 2019.

[48] Fabrizio Sebastiani. Machine learning in automated text categorization. ACM Computing Surveys, 34, 2002.

[49] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In Proceedings of ACL, 2010.

[50] Jeffrey Pennington, Richard Socher, Christopher D. Manning. GloVe: Global Vectors for Word Representation, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.

[51] Fellbaum, Christiane. WordNet and wordnets. In: Brown, Keith et al. (eds.), Encyclopedia of Language and Linguistics, Second Edition, Oxford: Elsevier, 665-670. 2005.

[52] Mohamed Chiny, Omar Bencharef, Moulay Youssef Hadi, Younes Chihab. A Client-Centric Evaluation System to Evaluate Guest's Satisfaction on Airbnb Using Machine Learning and NLP, Applied Computational Intelligence and Soft Computing, 2021.

[53] Twitter US Airline Sentiment. https://www.kaggle.com/crowdflower/twitter-airline-sentiment, February 2015.

View publication stats