Effect of word embedding vector dimensionality on sentiment analysis through short and long texts

Mohamed Chiny[1], Marouane Chihab[1], Abdelkarim Ait Lahcen[1], Omar Bencharef[2], Younes Chihab[1]
[1]Laboratory of Computer Sciences, Ibn Tofail University, Kenitra, Morocco
[2]Department of Computer Sciences, Cadi Ayyad University, Marrakesh, Morocco

Article Info

Article history:

Received May 8, 2022
Revised Nov 29, 2022
Accepted Dec 12, 2022

Keywords:

Deep learning
Gated recurrent unit
Global vectors
Sentiment analysis
Word embedding

ABSTRACT

Word embedding has become the most popular method of lexical description in a given context in the natural language processing domain, especially through the word to vector (Word2Vec) and global vectors (GloVe) implementations. Since GloVe is a pre-trained model that provides access to word mapping vectors on many dimensionalities, a large number of applications rely on its prowess, especially in the field of sentiment analysis. However, in the literature, we found that in many cases, GloVe is implemented with arbitrary dimensionalities (often 300d) regardless of the length of the text to be analyzed. In this work, we conducted a study that identifies the effect of the dimensionality of word embedding mapping vectors on short and long texts in a sentiment analysis context. The results suggest that as the dimensionality of the vectors increases, the performance metrics of the model also increase for long texts. In contrast, for short texts, we recorded a threshold at which dimensionality does not matter.

Corresponding Author:

Mohamed Chiny
Laboratory of Computer Sciences, Ibn Tofail University
Kenitra, Morocco
Email: mohamed.chiny@uit.ac.ma

1. INTRODUCTION

Research fields related to natural language processing, e.g. information retrieval [1], [2], document classification [3], named entity recognition [4], machine translation [5], sentiment analysis [6], [7], recommendation systems [8] or audience segmentation [9], [10] have in common that they are problems of perception related to our senses. Thus, they have always represented a great challenge for researchers because it is particularly difficult to describe a text using algorithms and mathematical formulas. Therefore, the first models deployed in this field were based on a certain expertise such as the passage through grammatical and syntactic rules. Several years have been devoted to research on the exploitation and transformation of this unstructured data in order to give it meaning. One of the most successful techniques is word embedding.

The foundations of word embedding were set by the linguistic theory of Zelling Harris, also known as distributional semantics [11], [12]. This theory states that a word is characterized by its context formed by the words around it. Therefore, words that share similar contexts also share the same meanings.

Word embedding is a numerical representation of text where words that share the same meaning also share a similar representation. Word embedding consists of representing each word in the dictionary as real-valued vectors in a defined vector space. These vectors are often generated using neural network-based models. As a result, the word embedding technique is often grouped into the deep learning domain. Indeed, the principle of using neural networks to model high-dimensional discrete distributions has already been supported for learning the joint probability of a set of random variables where each is likely different in nature [13]. Thus, the idea of word embedding is to use a dense distributed representation for each word, which results in vectors composed of dozens or hundreds of dimensions, contrasting with the thousands or millions of dimensions required for sparse word representations, such as one-hot encoding. Indeed, when applying one-hot encoding to words, we end up with high-dimensional sparse vectors containing a large number of zeros. On large datasets, this can lead to performance problems. Moreover, one-hot encoding does not take into account the semantics of words [14].

The word embedding approach seeks to associate each vocabulary word with a distributed word feature vector. The feature vector represents various aspects of the word that is associated with a point in a vector space. The number of features on which words are mapped is significantly smaller than the vocabulary size. In addition, the semantic relationships between words are reflected in the distance and direction of the vectors [15].

The idea of identifying similarities between words to generalize training sequences to new sequences dates back to the early 1990s. For example, it is used in approaches based on learning a grouping of words. Each word is deterministically or probabilistically associated to a class and words of the same class have a certain silimarity [16], [17].

In our study, we sought to identify the existence of a correlation between the number of dimensions of a word embedding vector and the performance of a sentiment analysis model according to the size of the text to be analyzed. We used a recurrent neural network gated recurrent unit (GRU), whose input was coupled to the word embedding representation vector using the global vectors (GloVe) model with dimensionality of 50, 100, 200 and 300, respectively. We computed performance metrics, including accuracy and F1 score, on short texts from the Twitter Airlines Sentiment dataset and relatively long texts from the Internet Movie Database dataset. The results of this study show that the dimensions of the word embedding vectors have a positive impact on the performance metrics for long texts, while these dimensions do not matter for short texts above a certain threshold.

2. LITERATURE REVIEW

2.1. Word embedding

Among the goals of statistical language modeling is learning the joint probability function of word sequences in a language. This task is intrinsically difficult because of the high dimensionality. Therefore, a word sequence on which the model will be evaluated is most likely to be different from all word sequences seen in training phase. Traditional n-gram based approaches succeed in generalizing by clustering very short overlapping sequences in the training set. However, the resulting models contain millions of parameters and thus learning them in a reasonable time is a complex task [15]. From a historical point of view, the encoding of words according to certain characteristics of their meaning began in the 1950s and 1960s [15]. In particular, the vector model in information retrieval makes it possible to represent a complete document by a mathematical object that aims to capture its meaning.

There are two approaches to encoding the context of a word: frequency-based approaches that count the words co-occurring with a given word in order to create dense vectors of small dimensions [18] and lexical embeddings that seek to predict a given word using its context or vice versa. This is the case for example of the word to vector (Word2Vec) algorithm [19]. This last approach relies on artificial neural networks to build these vectors. These models are trained on very large corpora (up to 1.6 billion words per day) in order to predict a word from its context or vice-versa. The Word2Vec model has two different architectures for creating word embeddings; the continuous bag-of-words (CBOW) model which attempts to predict a word from its neighboring words and the skip-gram model which attempts to predict context words from a central word [19]. It has been shown that distributed representations based on neural networks significantly outperform n-gram models [20]–[22]. Furthermore, since it is difficult to determine the exact number of meanings for each word, as the meaning of the word depends on the context, models such as adaptive probabilistic word embedding (APWE), where the polysemy of the words is defined on a latent interpretable semantic space [23] or word sense disambiguation [24] have been proposed.

2.2. GloVe

Recently, methods for learning the vector space of word representations have succeeded each other in capturing the fine-grained semantics of syntactic regularities using vector arithmetics. Nevertheless, the origin of these regularities has remained unclear. In order to bring out these regularities in word vectors, researchers at Stanford University combine the advantages of the two major families of models in the literature, namely the global matrix factorization and the local contextual window. The result is a pre-trained model named GloVe [25].

The approach used by the GloVe method for word integration is different. Indeed, it is an unsupervised learning algorithm that computes vector representations for words. The model is trained on aggregate word-word co-occurrence statistics of a given corpus. The resulting representations present interesting linear substructures of the word vector space. In fact, unlike Word2Vec, GloVe does not rely only on local statistics (information about the local context of words), but integrates global statistics (word co-occurrence) to generate word vectors [25].

The 50-, 100-, 200-, and 300-dimensional GloVe word vectors were trained on the Wikipedia dump and the gigaword 5 data corpus. They encode 400,000 tokens as single vectors and all tokens outside the vocabulary were encoded as a vector of zeros. The richness and robustness of GloVe vectors have allowed it to be at the heart of many works related to natural language processing as in [26], where the authors introduced an innovative MapReduce enhanced decision tree classification approach. They used several feature extractors, including the GloVe model, to efficiently detect and capture relevant data from given tweets. Alexandridis *et al.* [27] used various language models to represent social media texts and Greek language text classifiers, using word embedding implemented by the GloVe model, to detect the polarity of opinions expressed on social media. The GloVe model has also been used in sentiment analysis models, often associated with a recurrent neural network module like long sort-term memory (LSTM) or GRU [6], [28], [29].

3. ARCHITECTURE AND METHODOLOGY

The objective of our study is to evaluate the effect of the dimensionality of the word embedding vectors, implemented by the GloVe model, on the performance metrics related to sentiment analysis within short and long texts. The training and test data come from the Twitter US Airlines Sentiments [30] and internet movie database (IMDb) [31] datasets, with an overall average sentence length of 17 and 231 words respectively. For each dataset, we used the GloVe model implementing 50, 100, 200 and 300 dimensions. The sentiment analysis is performed using a GRU recurrent neural network. At the output, we retrieved the binary score of positivity or negativity of the sentiment experienced in the input instances as shown in Figure 1.

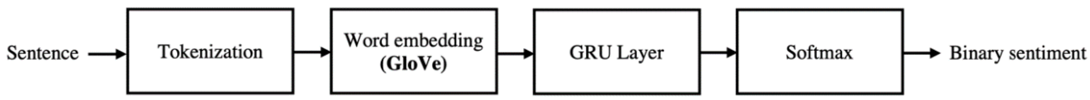


Figure 1. Proposed GRU-based sentiment analysis model for the study

3.1. Preprocessing

After cleaning and filtering the data from the two datasets, we proceeded to tokenization which consists in dividing the text into single occurrences or combinations of several successive occurrences of the same length. This operation also allows us to map the vocabulary of all the words of the dataset in a dictionary in order to train our model. We selected 10,000 and 2,000 tweets respectively for the train set and the test set to represent the short texts and 5,000 and 1,000 IMDb comments respectively for the long texts. In this study, we used word embedding by implementing the GloVe model which uses vectors of single words. Therefore, we segmented our sentences into single-word tokens. For each dataset used in this study, the entries do not have the same length. However, in order for our GRU cell-based model to work properly, we have defined the same sequence length which corresponds to the number of time steps for the GRU layer which is the maximum length calculated for a training text (36 tokens in the case of Twitter US Airlines Sentiments and 2,470 tokens in the case of IMDb).

3.2. Word embedding using GloVe

Word embedding improves text classification by solving the problem of sparse matrix (such as that resulting from the coding scheme of the bag-of-words model) and word semantics. In our study, we implemented word embedding using the GloVe model. Learning is performed on global word-word co-occurrence statistics aggregated from a text. The resulting representations thus show linear substructures of the word vector space [25].

In order to identify the effect of dimensionality of word embedding vectors. We implemented the GloVe model with dimensions 50, 100, 200, and 300 for both short and long texts. The dimension used will have a direct impact on the "Input vocab size" hyperparameter of the GRU model used for sentiment analysis as shown in Table 1.

3.3. GRU layer
Tokenization GRU, introduced by Cho *et al*. [32], is a class of recurrent neural networks that aims to solve the vanishing gradient problem that naturally accompanies traditional recurrent neural networks. This problem arises when backpropagating through the recurrent neural network during training, especially for networks with deeper layers [33], [34]. GRU is similar to the LSTM networks of the recurrent neural network class with a forgetting gate [35], but without the exit gate that characterizes LSTMs [33]. The performance of GRU on some tasks, in this case natural language processing, is similar to that of LSTMs [36], [37]. In addition, the implementation of GRU is simpler and less expensive in terms of computing power. For our study, we implemented the GRU layer with the hyerparameters shown in Table 1, in order to handle short texts (Twitter Airlines Sentiment) and long texts (IMDb).

Table 1. Hyperparameters applied to the GRU model deployed for sentiment analysis

Hyperparameter	Short text (Twitter)	Long text (IMDb)
Input vocabulary size	36	2,470
Output embedding dim.	50, 100, 200, 300	50, 100, 200, 300
GRU layer internal units	256	256
Optimizer	Adam	Adam
Loss	Categ. Crossentropy	Categ. Crossentropy
Activation function	Softmax	Softmax

3.4. Softmax layer
Softmax constitutes a generalization of logistic regression. It can be used in the case of multi-class classification. The softmax function transforms a K real values vector into a vector of the same dimension whose values add up to 1 (1). Thus, the softmax transforms the input values into values between 0 and 1. The latter can thus be interpreted as a normalized probability distribution.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad for\ j \in \{1, \dots, K\} \tag{1}$$

In practice, most multilayer neural networks end with a softmax layer that produces scaled real-valued scores that are easier to manipulate in further processing [6]. Indeed, in our work, we used the softmax layer that delivers two probability scores that represent the positivity and negativity of the input model text. The higher probability characterizes the overall binary sentiment experienced in the input sentence.

4. RESULTS
4.1. Evaluation of the dimensionality of word embedding on short texts
In practice, To the input of our model as shown in Figure 1, we applied texts from the Twitter US Airlines Sentiments dataset with an average length of 17 words. The longest tweet is 36 words. In the word embedding layer, we applied the GloVe model with the vectors of 50, 100, 200 and 300 dimensions respectively.

We can see that the accuracy of the model is 0.904 if we apply the word embedding by implementing the GloVe model whose words are mapped on 50 dimensions. This accuracy is 0.943, 0.944 and 0.946 if we increase the dimensionality of the word embedding to 100, 200 and 300 respectively as shown in Table 2. As for the F1 score (4), which summarizes the values of accuracy (2) and recall (3) as a harmonic mean, it is worth 0.721, 0.754, 0.747 and 0.773 with the respective dimensionalities of 50, 100, 200 and 300.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{2}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

We can subtly see that both performance metrics increase from dimensionality 50 to 100 as shown in Figure 2. However, accuracy remains almost constant from dimensionality 100 onwards. As for the F1 score, it climbs slightly beyond this threshold (after having recorded a slight drop for dimensionality 200). As for the accuracy, its maximum value (0.805) was recorded for dimensionality 50.

Table 2. Performance metrics for short texts according to the different dimensionalities of the GloVe model

Dimentionality	Accuracy	Precision	Recall	F1 Score
50d	0.904	0.805	0.652	0.721
100d	0.943	0.744	0.764	0.754
200d	0.944	0.672	0.840	0.747
300d	0.946	0.789	0.758	0.773

4.2. Evaluation of the dimensionality of word embedding on long texts

Concerning the long texts which come from the IMDb dataset, the average length of the comments is 231 words. The longest comment is 2,470 words. As for the short texts, we applied the same dimensions in the word embedding layer and kept the same hyperparameters as shown in Table 1. The performance metrics for this dataset have been reported in Table 3.

As can be seen, all performance metrics increase as the dimensionality of the word embedding increases as shown in Figure 3. Indeed, the accuracy increases from 0.686 for dimensionality 50 to 0.854 for dimensionality 300. The F1 score increases from 0.711 to 0.854. Precision and recall also increase significantly from 0.830 and 0.622 to 0.918 and 0.8 respectively. Therefore, it seems justified that one could extrapolate the observed results and deduce that the performance metrics concerning sentiment analysis in long texts will continue to evolve positively as the dimensionality of the vectors used for word embedding increases, at least up to a certain threshold that is greater than or equal to the 300d dimensionality.

Table 3. Performance metrics for long texts according to the different dimensionalities of the GloVe model

Dimentionality	Accuracy	Precision	Recall	F1 Score
50d	0.686	0.830	0.622	0.711
100d	0.784	0.830	0.739	0.782
200d	0.825	0.891	0.770	0.826
300d	0.854	0.918	0.800	0.854

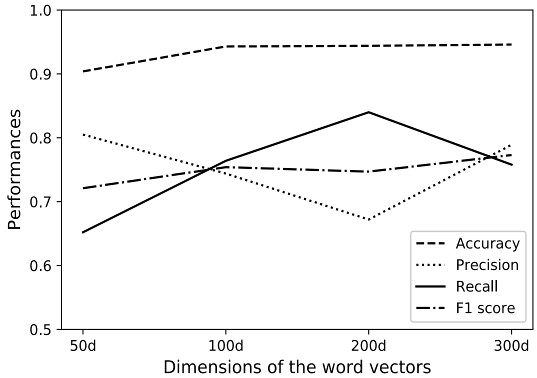


Figure 2. Graphical representation of the evolution of performance metrics for short texts

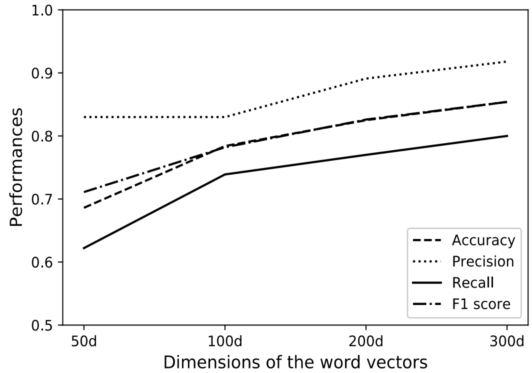


Figure 3. Graphical representation of the evolution of performance metrics for long texts

5. DISCUSSION

The results of our study clearly indicate that for long texts, such as IMDb comments, the performance metrics evolve as the dimensionality of the word embedding increases. On the other hand, for short texts such as tweets, we found that the performance metrics, in this case accuracy and F1 score which combines precision and recall, increase up to the 100d dimensionality threshold and then stabilize. Indeed, even if the dimensionality increases after reaching the 100d threshold, we notice that the model is almost insensitive to it. This behavior suggests to us that there are probably dimensions in the word mapping vector whose utility is minimized. We attribute this to the existence of dimensions that likely have anomalies in the GloVe model due to some parameters not being set to optimized global values [38]. The effect of these anomalies is revealed in the case of short texts like tweets. This is likely due to the difficulty encountered when disambiguating words in such texts [39].

Therefore, it would be wise to adopt the minimum dimensionality that ensures the best performance in order to optimize the use of computational resources. Indeed, word embedding with a small dimensionality is generally not expressive enough to capture all possible word relations, while a very large dimensionality is subject to over-fitting. In addition, the number of parameters for a word embedding or a model that relies on word embeddings, such as recurrent neural networks, is usually a linear or quadratic function of dimensionality, which affects training time and computational costs [40]. Our study showed that this dimensionality is 100d for short texts, such as tweets or comments related to blog posts and 300d for long texts, such as IMDb comments or reviews left on Airbnb [41]. Indeed, mapping each word of the corpus on such a large number of dimensions, and even more so if the corpus is large, could increase the complexity of the model, slow down the training speed and add inferential latency, which has as a direct consequence, the impossibility to deploy the model on real tasks.

6. CONCLUSION

The content generated by users of microblogs, such as social networks or opinion sharing sites, is a rich and abundant source of opinions and information. If carefully studied, it offers great potential for extracting useful and precious knowledge, in this case in terms of sentiment analysis, which aims to identify the opinion and subjectivity of people's feedback from unstructured text written in natural language. The machine learning models involved in performing sentiment analysis very often require mapping the input text into vectors that contain real values. This statistical modeling of language involves learning the joint probability function of sequences of words in a language, but which is marked by high dimensionality. There are solutions, such as n-grams, which give the possibility to obtain a generalization of overlapping word sequences, but they result in models that contain an excessively large number of parameters, which makes it impossible to train them in a reasonable time. The solution to such a problem consists in word embedding which is a vector model in information retrieval and which allows to represent a complete document by a mathematical object which aims at elucidating its meaning. Although the Word2vec model is a reference in terms of word embedding, the GloVe model, which is an unsupervised learning algorithm allowing to obtain vector representations of words, also seems to be very popular in some natural language processing-related domains such as sentiment analysis. GloVe allows to map the words of the dictionary on vectors of several dimensions, the most frequent being 50d, 100d, 200d and 300d. In our study, we investigated whether the dimensionality of the vector implementing the GloVe model can have an impact on performance metrics in relation to sentiment analysis for short and long texts. We therefore integrated GloVe into a sentiment analysis model based on GRU recurrent neural networks. Then, we trained it on corpora coming from the Twitter US Airlines Sentiments dataset which contains short texts and the IMDb dataset which contains relatively long texts. Each time, we applied a word mapping through vectors that implement the word embedding using the Glove model with a different dimensionality, in this case 50d, 100d, 200d and 300d. The results suggest that for short texts, the performance metrics (i.e., accuracy and F1 score) increase up to the 100d threshold and then stabilize. Thus, the use of word embedding through higher dimensionality vectors has almost no impact on the performance of our sentiment analysis model. On the other hand, for long text, we found that performance metrics increase the more we use word embedding across higher dimensionality vectors. Therefore, in order to optimize computational resources, we suggest using 100-dimensional word mapping through the GloVe model for short texts. On the other hand, it is recommended to use a word mapping with high dimensionality for long texts, within the limit that allows to find a compromise between the resources and the computational time on the one hand and the targeted performance metrics on the other hand.

REFERENCES

[1] C. D. Manning, H. Schütze, and P. Raghavan, *Introduction to information retrieval*. Cambridge university press, 2008.

[2] M. Chiny, O. Bencharef, and Y. Chihab, "Towards a machine learning and datamining approach to identify customer satisfaction factors on Airbnb," in *2021 7th International Conference on Optimization and Applications (ICOA)*, May 2021, pp. 1–5, doi: 10.1109/ICOA51614.2021.9442657.

[3] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, Mar. 2002, doi: 10.1145/505282.505283.

[4] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning," in *In Proceedings of ACL*, 2010, pp. 384–394.

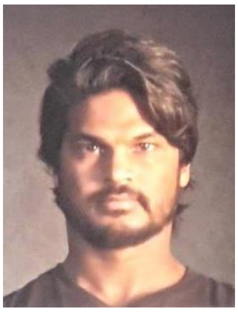
[5] Y. T. Phua, S. Navaratnam, C.-M. Kang, and W.-S. Che, "Sequence-to-sequence neural machine translation for English-Malay," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, no. 2, p. 658, Jun. 2022, doi: 10.11591/ijai.v11.i2.pp658-665.

[6] M. Chiny, M. Chihab, O. Bencharef, and Y. Chihab, "LSTM, VADER and TF-IDF based hybrid sentiment analysis model," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, 2021, doi: 10.14569/IJACSA.2021.0120730.

[7] D. Febrian Sengkey, A. Jacobus, and F. Johanes Manoppo, "Effects of kernels and the proportion of training data on the accuracy of SVM sentiment analysis in lecturer evaluation," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 9, no. 4, p. 734, Dec. 2020, doi: 10.11591/ijai.v9.i4.pp734-743.
[8] F. Sakketou and N. Ampazis, "A constrained optimization algorithm for learning GloVe embeddings with semantic lexicons," *Knowledge-Based Systems*, vol. 195, p. 105628, May 2020, doi: 10.1016/j.knosys.2020.105628.
[9] N. A. Rahman, S. D. Idrus, and N. L. Adam, "Classification of customer feedbacks using sentiment analysis towards mobile banking applications," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, no. 4, p. 1579, Dec. 2022, doi: 10.11591/ijai.v11.i4.pp1579-1587.
[10] M. Chiny, M. Chihab, E. M. Juiher, K. Jabari, O. Bencharef, and Y. Chihab, "The impact of influencers on the companies reputation in developing countries: Case of Morocco," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 1, p. 410, Oct. 2021, doi: 10.11591/ijeecs.v24.i1.pp410-419.
[11] Z. Harris, "Language and information," Columbia University Press, New York, 1988.
[12] Z. Harris, *A theory of language and information (a mathematical approach)*. Oxford: Clarendon Press, 1991.
[13] S. Bengio and Y. Bengio, "Taking on the curse of dimensionality in joint distributions using neural networks," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 550–557, May 2000, doi: 10.1109/72.846725.
[14] P. Cerda and G. Varoquaux, "Encoding high-cardinality string categorical variables," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 3, pp. 1164–1176, Mar. 2022, doi: 10.1109/TKDE.2020.2992529.
[15] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.
[16] P. F. Brown, V. J. Della Pietra, P V DeSouza, J. C. Lai, and R. L. Mercer, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.
[17] F. Pereira, N. Tishby, and L. Lee, "Distributional clustering of English words," in *Proceedings of the 31st annual meeting on Association for Computational Linguistics -*, 1993, pp. 183–190, doi: 10.3115/981574.981598.
[18] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, Sep. 1990, doi: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.
[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, Jan. 2013, doi: 10.48550/arXiv.1301.3781.
[20] T. Mikolov, A. Deoras, S. Kombrink, L. Burget, and J. Černocký, *Empirical evaluation and combination of advanced language modeling techniques*. INTERSPEECH, 2011.
[21] A. Mnih and G. Hinton, "Three new graphical models for statistical language modelling," in *Proceedings of the 24th international conference on Machine learning - ICML '07*, 2007, pp. 641–648, doi: 10.1145/1273496.1273577.
[22] Q. Luo, W. Xu, and J. Guo, "A study on the CBOW model's overfitting and stability," in *Proceedings of the 5th International Workshop on Web-scale Knowledge Representation Retrieval & Reasoning - Web-KR '14*, 2014, pp. 9–12, doi: 10.1145/2663792.2663793.
[23] S. Li, Y. Zhang, R. Pan, and K. Mo, "Adaptive probabilistic word embedding," in *Proceedings of The Web Conference 2020*, Apr. 2020, pp. 651–661, doi: 10.1145/3366423.3380147.
[24] M. Sasaki, "Word embeddings of monosemous words in dictionary for word sense disambiguation," *SEMAPRO 2018, The Twelfth International Conference on Advances in Semantic Processing*. 2018.
[25] J. Pennington, R. Socher, and C. Manning, "Glove: global vectors for word representation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2014, doi: 10.3115/v1/d14-1162.
[26] F. Es-Sabery *et al.*, "A mapreduce opinion mining for COVID-19-related tweets classification using enhanced ID3 decision tree classifier," *IEEE Access*, vol. 9, pp. 58706–58739, 2021, doi: 10.1109/ACCESS.2021.3073215.
[27] G. Alexandridis, I. Varlamis, K. Korovesis, G. Caridakis, and P. Tsantilas, "A survey on sentiment analysis and opinion mining in greek social media," *Information*, vol. 12, no. 8, p. 331, Aug. 2021, doi: 10.3390/info12080331.
[28] R. Ni and H. Cao, "Sentiment analysis based on gloVe and LSTM-GRU," in *2020 39th Chinese Control Conference (CCC)*, Jul. 2020, pp. 7492–7497, doi: 10.23919/CCC50068.2020.9188578.
[29] B. Subba and S. Kumari, "A heterogeneous stacking ensemble based sentiment analysis framework using multiple word embeddings," *Computational Intelligence*, vol. 38, no. 2, pp. 530–559, Apr. 2022, doi: 10.1111/coin.12478.
[30] Kaggle, "Twitter US Airline Sentiment." 2015, Accessed: Jun. 01, 2021. [Online]. Available: https://www.kaggle.com/crowdflower/twitter-airline-sentiment.
[31] Kaggle, "IMDb dataset of 50K movie reviews." 2019, Accessed: Jun. 01, 2021. [Online]. Available: https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews.
[32] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: encoder–decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014, pp. 103–111, doi: 10.3115/v1/W14-4012.
[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
[34] R. Siddalingappa and K. Sekar, "Bi-directional long short term memory using recurrent neural network for biological entity recognition," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, no. 1, p. 89, Mar. 2022, doi: 10.11591/ijai.v11.i1.pp89-101.
[35] F. A. Gers, "Learning to forget: continual prediction with LSTM," in *9th International Conference on Artificial Neural Networks: ICANN '99*, 1999, vol. 1999, pp. 850–855, doi: 10.1049/cp:19991218.
[36] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.
[37] N. Gruber and A. Jockisch, "Are GRU cells more specific and LSTM cells more sensitive in motive classification of text?," *Frontiers in Artificial Intelligence*, vol. 3, Jun. 2020, doi: 10.3389/frai.2020.00040.
[38] Y.-Y. Lee, H. Ke, H.-H. Huang, and H.-H. Chen, "Less is more," in *Proceedings of the 25th International Conference Companion on World Wide Web - WWW '16 Companion*, 2016, pp. 71–72, doi: 10.1145/2872518.2889381.
[39] A. Chihaoui, A. Bouhafs, M. Roche, and M. Teisseire, "Disambiguation of spatial entities by active learning (in [Prancis]:[Désambiguïsation des entités spatiales par apprentissage actif])," *Revue Internationale de Géomatique*, vol. 28, no. 2, pp. 163–189, Apr. 2018, doi: 10.3166/rig.2018.00053.
[40] Z. Yin and Y. Shen, "On the dimensionality of word embedding," *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*. Canada, Dec. 2018, doi: 10.48550/arXiv.1812.04224.

[41] M. Chiny, O. Bencharef, M. Y. Hadi, and Y. Chihab, "A client-centric evaluation system to evaluate guest's satisfaction on airbnb using machine learning and NLP," *Applied Computational Intelligence and Soft Computing*, vol. 2021, pp. 1–14, Feb. 2021, doi: 10.1155/2021/6675790.

BIOGRAPHIES OF AUTHORS



Mohamed Chiny is an engineer at Cadi Ayyad University in Marrakech, Morocco. He obtained a Master in networks and systems. He is a specialist in web development and application security. He is conducting research in the field of artificial intelligence, including deep learning and natural language processing. He can be contacted at email: mohamed.chiny@uit.ac.ma.

Marouane Chihab obtained a Master degree in in jully 2019 from the Faculty of Sciences, University of Mohammed V, Rabat, Morocco. Currently he is a PhD student at the Computer Sciences Research Laboratory (Lari), Faculty of Sciences, Ibn Tofail University, Kenetra, Morocco. His research concerns artificial intelligence, and its applications. He can be contacted at email: marouane.chihab2@gmail.com.

Abdelkarim Ait Lahcen holds an Engineering Diploma in Computer Science, Networking & Telecommunications, besides several professional certificates and skills. He is currently the Head of Information Systems and Digital Resources Department at Cadi Ayyad University, Marrakesh, Morocco. He is a PhD Student and His research areas of interest includes artificial intelligence, machine learning, global optimization and data mining. He can be contacted at email: a.aitlahcen@uca.ac.ma.

Dr. Omar Bencharef received the D.Sc. degree (Doctor Habilitatus D.Sc.) in computer science from the Faculty of Sciences and Technology, Cadi Ayyad University, Marrakech, Morocco, in April 2018. he is a Professor of Computer Science in the Faculty of Sciences and Technology, Cadi Ayyad University, since 2013. His research interests include the signal and image processing and coding, networking and artificial intelligence. He can be contacted at email: o.bencharef@uca.ac.ma.

Dr. Younes Chihab received the doctorate thesis in Network Security from the Faculty of Sciences in the Cadi Ayyad University, Marrakech, Morocco, in December 2013. His research was in artificial intelligence, signal processing and machine learning. In 2019, He received the PHD degree in Computer Sciences from the Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco. His research focuses on signal and image processing and coding, networks and artificial intelligence. He is currently a professor and member of the Computer Sciences Research Laboratory (Lari), Superior School of Technology, Ibn Tofail University, Kenetra, Morocco. He can be contacted at email: younes.chihab@uit.ac.ma.