A comparison of Word-Embeddings in Emotion Detection from Text using BiLSTM, CNN and Self-Attention

Marco Polignano
marco.polignano@uniba.it
University of Bari "Aldo Moro", Dept. of Computer Science

Pierpaolo Basile
pierpaolo.basile@uniba.it
University of Bari "Aldo Moro", Dept. of Computer Science

Marco de Gemmis
marco.degemmis@uniba.it
University of Bari "Aldo Moro", Dept. of Computer Science

Giovanni Semeraro
giovanni.semeraro@uniba.it
University of Bari "Aldo Moro", Dept. of Computer Science

ABSTRACT

User profiling is becoming increasingly holistic by including aspects of the user that until a few years ago seemed irrelevant. The content that users produce on the Internet and is an essential source of information about their habits, preferences, and behaviors in many situations. One factor that has proved to be very important for obtaining a complete user profile that includes her psychological traits are the emotions experienced. Therefore, it is of great interest to the research community to develop approaches for identifying emotions from the text that are accurate and robust in situations of everyday writing. In this work, we propose a classification approach based on deep neural networks, Bi-LSTM, CNN, and self-attention demonstrating its effectiveness on different datasets. Moreover, we compare three pre-trained word-embeddings for words encoding. The encouraging results obtained on state-of-the-art datasets allow us to confirm the validity of the model and to discuss what are the best word embeddings to adopt for the task of emotion detection. As a consequence of the great importance of deep learning in the research community, we promote our model as a starting point for further investigations in the domain.

CCS CONCEPTS

• **Computing methodologies** → **Information extraction**; *Neural networks*; • **Information systems** → **Personalization**.

KEYWORDS

emotion detection, sentiment analysis, deep learning, text analysis, natural language processing, word embeddings

ACM Reference Format:
Marco Polignano, Pierpaolo Basile, Marco de Gemmis, and Giovanni Semeraro. 2019. A comparison of Word-Embeddings in Emotion Detection from Text using BiLSTM, CNN and Self-Attention. In *27th Conference on User Modeling, Adaptation and Personalization Adjunct (UMAP'19 Adjunct), June 9–12, 2019, Larnaca, Cyprus.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3314183.3324983

UMAP'19 Adjunct, June 9–12, 2019, Larnaca, Cyprus

ACM ISBN 978-1-4503-6711-0/19/06…$15.00
https://doi.org/10.1145/3314183.3324983

1 INTRODUCTION

Sentiment analysis is a field of Natural Language Processing (NLP) [22] that has established itself thanks to the importance of the results it produces for user profiling. In particular sentiment analysis is commonly associated with the task of opinion mining where the goal is to identify for each relevant aspect of the sentence a polarity (positive, negative, neutral). Very often, in real applications, there is, instead, the need to go beyond and to identify a finer granularity for the state of mind expressed by the user. In literature, there are different emotional models, each with its level of granularity and peculiarity of the domain of application [12, 27]. Nevertheless, the identification of different emotions from a short sentence is still a challenging task. Each user has his or her behavioral model which, in some cases, can deviate from the standard model, but the use of emotions in personalized systems is a well-established practice, and its importance has been confirmed by various works [24, 30]. For example, a Recommender System able to know the emotions of the end users can adapt its suggestions to make communication more effective, the results more consistent and the recommendations more accurate. Sentiment analysis and emotion detection is, therefore, an essential aspect for user profiling, which in recent years has received significant interest from the machine learning community [3, 11]. In this work, we propose a deep neural network model that can identify the emotion expressed by a text accurately using the latest approaches of supervised learning.

2 RELATED WORK

The proliferation of Social Media and user-generated contents has fed the acquisition of large dataset of text annotated with hashtags and descriptive labels [20]. This new wave of data allows machine learning for natural language processing (NLP) to increase its diffusion in particular in approaches for sentiment analysis [21]. Emotion detection from text emerged from sentiment analysis as an important area of research that works on the whole sentence with the purpose to detect a more fine-grain sentiment defined at emotional level [35]. The current techniques of emotion detection from text can be broadly classified in the following categories: Keyword-based Method, Lexicon-based Method, Machine learning Method and Hybrid Method [28].

Machine learning approaches. The keyword-based approaches for emotion detection have the limit that same word can elicit multiple emotions depending on the context in which they are used. To

overcome this issue, machine learning approaches have been used to identify the relationships between emotions and the input [33].

Recently deep neural networks, have been demonstrated to be the best option when approaching classification tasks of contents in natural language [10]. Recurrent neural networks (RNNs) [26] have proven to be very useful in natural language processing tasks. In particular, their variants Long-Short Term Memory (LSTM) [14] and Gated Recurrent Neural Net (GRUs) [8]. Convolutional networks (CNN) [17] are a further deep approach often used for image classification but with numerous applications also in the NLP field, including sentiment detection from text.

Features engineering and transfer learning. Typical emotion detection systems work mostly with features directly extracted from text [16]. This step of feature engineering is often computationally expensive, and it requires long steps of analysis and correlation studies. Feature engineering falls short of extracting and organizing all the discriminative information from data. Moreover, a simple vector-space strategy can often be sufficient for resolving easier tasks, but it suffers from sparsity and lack of generalization. Recently, a more effective strategy has been commonly adopted for automatically discover a representation of words in a semantic space using neural networks. In [4] the author exposes the concept of word embedding that can be summarized as a "learned distributed feature vector to represent similarity between words". This concept has been exploited by Mikolov [18] through word2vec, a tool for implementing work embeddings through two standard approaches: skip-gram and CBOW. The model trained in a specific domain of application (e.g., newspapers) can be easily used for representing words which are used a different context, for example, Twitter. This singularity has made very famous transfer learning in NLP where pre-trained word embedding models are reused in a multitude of different domains. An alternative word embedding representation is described in [23]. GloVe, differently from word2vec, produces a word vector space trained on global word-word co-occurrence counts and it uses statistics for producing a final model which performs quite better than word2vec in some NLP tasks such as the analogy. Moreover, if we consider the efficiency in learning time GloVe consistently outperforms word2vec. Bojanowski proposes an extension of the skip-gram approach in 2017 as FastText [5]. He suggested to learn representations for character n-grams and to represent words as the sum of the n-gram vectors. This final strategy has been demonstrated to be computationally efficient and affordable for many NLP tasks.

Contribute. The contribute of our paper to the state-of-the-art includes two aspects. First, we propose an innovative classification model of emotions based on the combination of Bi-LSTM, CNN deep neural networks mediated by a level of self-attention [32]. Lately, we evaluated the model on three datasets by varying the word-embeddings used for formalizing sentences. These experiments support our claims about the high accuracy of the model proposed and the differences in performances found by choice of a specific word-embedding as word encoder.

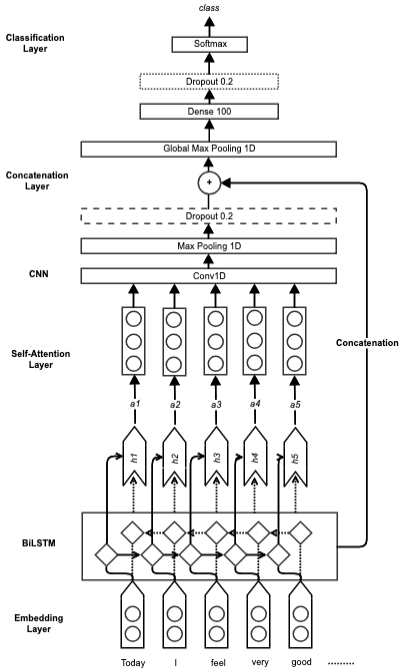

Figure 1: The architecture of the classification model based on Bi-LSTM, CNN and Self-Attention

3 EMOTION DETECTION MODEL

The model of emotion detection applied in this study is based on the synergy between two deep learning classification approaches the long-short-term memory networks (LSTM) [14] in their bi-directional variation and the convolutional neural networks [17] (CNN) mediated by a max pooling approach. Moreover, we decided to include, after the Bi-LSTM, a self-attention layer to allow the system to capture distant relationships among words with a different weight depending theirs contribute for the classification. Fig. 1 shows the complete stack of the proposed model for emotion detection, while in Fig.2 It is shown its implementation through Keras deep learning framework [9] .

3.1 Embedding Layer

The first layer of the model has the purpose to transform the sentences provided in input as a vector of word embeddings so that they could be computable by the neural network. Since each sentence is of different length, it is necessary to define a maximum number of terms to consider for the classification. This means that if the number of terms is smaller than the number of max_terms, a padding with zeros operation will be applied in order to reach the designated dimension. On the contrary, for each sentence, only the terms up to a maximum number equal to max_terms are considered. We set the value of max_terms fixed at 80 words. Word embeddings could be trained directly on "training data" of the domain of application, but this strategy can lack generalization. When a new sentence to classify is provided as input many words in it could be not possible to be translated making impossible the correct

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 80, 300)	0
bidirectional_1 (Bidirectional)	(None, 80, 400)	801600
seq_self_attention_1 (SeqSelfAtt	(None, 80, 400)	25665
conv1d_1 (Conv1D)	(None, 76, 400)	800400
max_pooling1d_1 (MaxPooling1D)	(None, 38, 400)	0
dropout_1 (Dropout)	(None, 38, 400)	0
concatenate_1 (Concatenate)	(None, 118, 400)	0
global_max_pooling1d_1 (GlobalMa	(None, 400)	0
dense_1 (Dense)	(None, 100)	40100
dropout_2 (Dropout)	(None, 100)	0
dense_2 (Dense)	(None, 7)	707

Total params: 1,668,472
Trainable params: 1,668,472
Non-trainable params: 0

Figure 2: Implementation of the proposed emotion detection model in Keras

classification. For this reason, we decided to use a common practice of transfer learning in NLP tasks i.e. the use of vector spaces word embeddings already pre-calculated on different domains. This allows us to cover an extensive variety of terms by reducing the computational cost of the model and including information about terms that are independent of their domain of use. We decided to compare the results of the model obtained varying three different pre-trained word embeddings:

- **Google word embeddings (GoogleEmb)** [1]: 300 dimensionality word2vec vectors, case sensitive, composed by a vocabulary of 3 million words and phrases that are obtained from roughly 100 billion of tokens extracted by a huge dataset of Google News;
- **GloVe (GloVeEmb)**[2]: 300 dimensionality vectors, composed by a vocabulary of 2.2 million words case sensitive obtained from 840 billion of tokens and trained on data crawled from generic Internet web pages;
- **FastText (FastTextEmb)**[3]: 300 dimensionality vectors, composed by a vocabulary of 2 million words and n-grams of the words, case sensitive and obtained from 600 billion of tokens trained on data crawled from generic Internet web pages by Common Crawl nonprofit organization;

In order to transform sentences into word embeddings, it is necessary to apply a strategy of data pre-processing. Each sentence is divided into tokens through the TweetTokenizer class of the NLTK library. After that, we used ekphrasis preprocessor library to annotate mentions, URL, email, numbers, dates, amount of money, and make word spelling correction and hashtag unpacking if necessary. Words not found in the vector space are transformed into word embedding through a random vector selected from the entire collection, as proposed by Zhang [34].

[1]https://goo.gl/zQFRx3
[2]https://nlp.stanford.edu/projects/glove/
[3]https://fasttext.cc/docs/en/english-vectors.html

3.2 Bi-LSTM Layer

Considering the intrinsic sequential relationship between the terms of a sentence, i.e. the next term depends on the context of previous terms, the contribution made by a recurrent neural network in order to grasp this relationship is evident. LSTM uses the forget gate (hidden neuron) for dynamically scale the weights of its internal "self loop" depending by the weights learned by the network for previous words provided as input [14]. This step provides to the layer a "memory" for considering the relations with the past elements in input. The bi-directional variant considers the relations among inputs by both the directions, finally provided as output the concatenation of the relations from both the sides as in Eq. 1.

$$x_i = \overleftarrow{x_i}||\overrightarrow{x_i} \quad x_i \in R^{2d} \tag{1}$$

where $||$ is the operator of concatenation and d is the dimension of the LSTM in terms of hidden units.

We have configured the LSTM network by setting the value of hidden units to 200 and the dropout value to 0.3. This choice was motivated by the need to reduce the dimensionality of the output of the network so that the operations carried out by the following layers were not computationally too expensive. Moreover, the dropout value was used to reduce, during the learning, the effect of the overfitting on the training data. We have decided to vary also the function of activation used by the net setting it to the hyperbolic tangent function (tanh). This activation function has an S-Shape and produces values among -1 and 1 making layer output more center to the 0. Moreover, it produces a gradient larger than sigmoid function helping to speed up the convergence [13].

3.3 Self-Attention Layer

A level of self-attention [7] is added following the LSTM. As well as the attention strategy proposed in [1], self-attention, also known as intra-attention, provides the model ability to weigh the vectors of single words of the sentence differently, according to the similarity of the neighboring tokens. It is possible to say that the level of attention can provide us an idea of what features the network is looking at most during learning and subsequent classification. In particular, we consider an additive self-attention context-aware equal to the whole set of words in input (Eq. 2) [36].

$$\begin{aligned} g_{t,t'} &= tanh(W_g h_t + W'_g h'_t + b_g) \\ \alpha_{t,t'} &= \sigma(W_\alpha g_{t,t'} + b_\alpha) \\ a_{t,t'} &= softmax(\alpha_{t,t'}) \\ l_t &= \sum_{t'=1}^{n} a_{t,t'} h'_t \end{aligned} \tag{2}$$

where, σ is the element-wise sigmoid function, W_g and W'_g are the weight matrices corresponding to the hidden states h_t and $h_{t'}$âĂš; W_α is the weight matrix corresponding to their non-linear combination; b_g and b_α are the bias vectors. The attention-focused hidden state representation l_t of a token at timestamp t is given by the weighted summation of the hidden state representation $h_{t'}$ of all other tokens at timesteps t. We use the last self-attention implementation for Keras available at https://github.com/CyberZHG/keras-self-attention .

3.4 CNN Layer

CNN is a robust neural network ideal for working on data with a shape of grid [15] as a consequence of the convolutional operations performed by the algorithm over adjacent cells. The result of the convolution is a grid more dense and smaller of the previous that captures the hidden relations among cells that fall in the kernel dimension. In our specific case, we applied the CNN layer on the result of the attention algorithm. Such hidden level has a matrix form as a consequence of the vectorial representation supplied by the word embeddings on the tokens in input. In detail, it has the form 80x400 which allows us to apply a 1D Convolutional network with 400 filters and 5x5 kernel. We used, as activation function, ReLu that unlike the hyperbolic tangent is faster to calculate [13].

On the top of the CNN layer, we added a Max Pooling function for subsampling the values obtained, reducing the computational load and, the number of parameters of the model. In particular, we used a small 2x2 kernel.

3.5 Emotion classification

On the output of the last max pooling layer, we applied a dropout function for reducing the number of connections inside the model and limiting the effect of overfitting [13]. Dropout is a common regularization technique that for a defined value of p sets p fraction of units to 0 at each update during training time.

The hidden model obtained until this step has been merged with the output of the previous Bi-LSTM. We apply this operation for letting the model to better conceptualize both local and long-term features. After that, we used a max pooling layer for 'flatten' the results and reduce the model parameter. An analog function of dimensionality reduction is performed by the consequent dense layer and the following dropping function. Finally, another dense layer with a soft-max activation function has been applied for estimating the probability distribution of each of the emotional classes of the dataset.

The model has been trained using the categorical cross entropy loss function [13] and Adam optimizer for 100 epochs and best models have been used for the classification phase.

4 MODEL EVALUATION

The evaluation of the proposed model of emotion detection has been carried out in order to be able to answer two different research questions. First, we want to investigate whether the model produces results of accuracy that are comparable with the state-of-the-art approaches. Secondly, we want to understand if the choice of word embedding vector space influences the final performance of the model with statistical validity. Such a result may produce interesting considerations to be used in future work on the subject by providing a detailed starting point.

4.1 Datasets, Baselines and Metrics

In the literature, there are not many datasets annotated with emotional traits, and a high percentage of them have a number of examples within them too small to generate statistical significance. We decided to use one very common dataset with a number of examples enough for training our deep neural model: ISEAR [29]; and two released more recently by SemEval 2018 task 1 [19] and SemEval 2019 task 3 [6].

ISEAR. The ISEAR dataset has been collected by Scherer et al. [29] providing a questionnaire to 1096 participants with different cultural background. In total it is composed by 7666 instances annotated by seven main emotions: joy (14.27%), fear (14.28%), anger (14.3%), sadness (14.3%), disgust (14.3%), shame (14.3%), and guilt (14.25%).

SemEval 2018 task 1. Task 1 of SemEval 2018 was defined to identify the intensity of emotions in the text. For our experiment, we used only the phrases present in the training and test set with an emotional intensity higher than 0.5. In total, we obtained a dataset of 2761 records for the training set noted with emotions: anger (33.8%), fear (16%), joy (25.6%), sadness (24.6%). The test set consists of 1580 tweets divided into anger (24.6%), fear (14.5%), joy (36.6%), sadness (24.3%).

SemEval 2019 task 3. The EmoContext task at SemEval 2019 [6] [4] aims to understand the emotion of the last turn of a short dialog composed of three discussion turns extracted from social media. The *training set* is composed of 30k records annotated with three main emotions: Happy, Sad, Angry and the 'other' class that includes all other not annotated emotions. The dataset follows a data distribution of respectively 15k, 5k, 5k, 5k. The *test set* is composed by 5509 records where 'Happy' records compose the 2,95% of the total, the 2,68% is about records in the 'Sad' class, and 3,15% is the amount of 'Angry' records. The papers about systems submitted to the challenge have not been published yet.

4.2 Methodology

We run the model proposed in Sec. 3 three times per dataset, using one word-embedding space at a time among Google, GloVe, FastText previously detailed.
ISEAR dataset has been subdivided into 60% training and 40% using a normal distribution of examples among classes for making our results comparable with the system proposed by Razek et al. [25] and Balahur et al. [2]. The other two datasets are already provided subdivided into training and test set. The results for each dataset are labels coherent with the annotation provided in their test set.

As a baseline for each configuration, we used the results obtained by an SVM configured with "rbf" kernel, C = 1.5 and Gamma = 0.2, a Naïve Bayes algorithm and a Random Forest configured with 500 trees. Sentence representations are obtained by the averaging the word-embedding (FastText) of each word that composes the record.

Evaluation metrics. In the evaluation of the model, we used as metrics precision, recall and F1 measure for each emotional class and the Area Under the Curve applied to ROC curve (AUC) for the two SemEval datasets. In the results for the SemEval 2018 task 1, we included its respective values of F1 micro-averaged measure [31]. Whereas for SemEval 2019 task 3, we calculated the F1 micro-averaged measure excluding the "other" emotional class for making the results compatible with the scores reported on the global challenge leaderboard [5].

[4] https://www.humanizing-ai.com/emocontext.html
[5] https://competitions.codalab.org/competitions/19790

Table 1: The table shows the results obtained for the ISEAR dataset

	joy			**fear**			**anger**			**sadness**			**disgust**			**shame**			**guilt**		
	P	R	**F1**	*P*	*R*	***F1***	*P*	*R*	***F1***	*P*	*R*	***F1***	*P*	*R*	***F1***	***P***	***R***	***F1***	*P*	R	F1
SVM	0.72	0.69	0.71	0.67	0.68	0.67	0.53	0.41	0.46	0.54	0.67	0.60	0.55	0.61	0.58	0.40	0.46	0.43	0.49	0.44	0.46
Naïve Bayes	0.47	0.63	0.54	0.49	0.57	0.53	0.55	0.23	0.33	0.32	0.50	0.39	0.36	0.40	0.38	0.09	0.31	0.13	0.34	0.28	0.30
Random Forest	0.71	0.57	0.63	0.61	0.56	0.59	0.40	0.39	0.39	0.47	0.59	0.52	0.55	0.48	0.52	0.34	0.46	0.39	0.39	0.41	0.40
Balahur et al. [2]	0.43	0.47	0.45	0.48	0.55	0.51	0.35	0.41	0.38	**0.70**	**0.76**	**0.73**	0.29	0.24	0.26	0.44	0.41	0.42	0.46	0.38	0.40
Razek et al. [25]	0.26	0.50	0.34	0.26	0.55	0.35	0.20	0.52	0.29	0.27	0.60	0.37	0.22	0.46	0.30	0.20	0.48	0.28	0.20	0.51	0.29
GoogleEmb	0.82	**0.71**	0.76	0.70	**0.78**	**0.74**	0.55	0.52	0.54	0.64	0.67	0.65	0.58	0.67	0.62	**0.53**	0.49	0.51	0.56	0.57	0.56
GloVeEmb	**0.86**	0.68	0.76	0.71	0.74	0.72	0.45	**0.60**	0.52	0.63	0.70	0.66	**0.69**	0.57	0.63	0.49	0.52	0.50	0.54	0.56	0.55
FastTextEmb	0.80	0.77	**0.78**	**0.78**	0.65	0.71	**0.58**	0.52	**0.55**	0.64	0.66	0.65	0.60	**0.74**	**0.66**	0.48	**0.56**	**0.52**	**0.56**	**0.57**	**0.57**

Table 2: The table shows the results obtained for the SemEval 2018 task 1 dataset

	anger			fear			joy			sadness					
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	**μF1**	**Acc.**	**AUC**
SVM	0.79	0.68	0.73	0.59	0.71	0.64	0.91	0.97	0.94	0.68	0.66	0.67	0.78	0.778	0.835
Naïve Bayes	0.54	0.67	0.59	0.62	0.41	0.50	0.74	0.88	0.80	0.58	0.51	0.54	0.63	0.632	0.750
Random Forest	**0.88**	0.55	0.67	0.27	**0.77**	0.40	0.89	0.93	0.91	0.53	0.64	0.58	0.71	0.710	0.773
GoogleEmb	0.77	0.77	0.77	0.73	0.73	0.73	0.94	0.96	0.95	0.72	0.69	0.70	0.81	0.813	0.939
GloVeEmb	0.77	0.80	0.78	0.75	0.69	0.72	0.97	0.94	0.96	0.68	0.72	0.70	0.82	0.818	0.940
FastTextEmb	0.77	**0.83**	**0.80**	**0.77**	0.70	**0.73**	**0.97**	**0.97**	**0.97**	**0.74**	**0.74**	**0.74**	**0.84**	**0.836**	**0.946**

Table 3: The table shows the results obtained for the SemEval 2019 task 3 dataset

	others			angry			sad			happy					
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	**μF1 - others**	**Acc.**	**AUC**
SVM	0.87	0.93	0.90	0.64	0.41	0.50	0.67	0.44	0.53	0.43	0.40	0.42	0.486	0.825	0.768
Naïve Bayes	0.44	0.92	0.59	0.66	0.08	0.14	0.35	0.28	0.31	0.44	0.26	0.33	0.198	0.445	0.640
Random Forest	0.96	0.88	0.92	0.22	0.46	0.30	0.32	0.58	0.41	0.21	0.43	0.28	0.511	0.851	0.619
GoogleEmb	0.90	0.95	0.92	0.81	0.59	0.68	0.67	0.49	0.56	0.59	0.46	0.52	0.591	0.869	0.927
GloVeEmb	0.91	0.96	0.93	0.87	0.53	0.66	0.62	0.56	0.59	0.56	0.52	0.54	0.602	0.873	0.943
FastTextEmb	**0.93**	**0.96**	**0.95**	**0.83**	**0.61**	**0.70**	**0.80**	**0.65**	**0.72**	**0.68**	**0.69**	**0.69**	**0.703**	**0.906**	**0.961**

5 RESULTS AND ANALYSIS

The results showed in Tab. 1 describe the efficacy of our approach comparing it with state-of-the-art systems on a common dataset online available. It is important to note that it overcomes the others for all the metrics and all the emotional classes except for "sadness" reported in the paper of Balahur et al.[2]. On this class, all the reported systems perform worse, but globally it is clear how the approaches proposed by Balahur et al.[2], and by Razek et al. [25] are less performant. Similar behavior is observable in Tab. 2 and Tab. 3 where the model proposed in this work obtains the best results compared with all the baselines for every emotional class. In particular, for the dataset about SemEval 2018 task 1, our approach obtained a max value of μF1 score of 0.84 versus the 0.78 of the SVM baseline. For the SemEval 2019 task 3 dataset, we obtain a max value of μF1 score of 0.703 versus the 0.44 of the Random Forest classifier, and if we compare it with the global challenge leaderboard, we can affirm that our results are in line with them of systems in top part of it. These results are confirmed by both the tables by the AUC score and they allow us to confirm the efficacy of the model for the emotion detection task.

Focusing the analysis of the results obtained by our system in all the three datasets it is possible to compare the performances obtained by each configuration varying the word embedding vectorial space used as the first layer of the network. In Tab. 1 it is possible to observe that differences among the configurations are not very sharp, but surely, considering the F1 score for each emotional class, the one which uses FastText overcome the other results. In Tab. 2 and Tab. 3 the model scores a similar trend and, in particular, the scores obtained on SemEval 2019 task 3 dataset by the configuration which uses FastText is the best for all the metrics calculated. The value of the AUC score obtained among these configurations is very significant for highlighting the better performances obtained using FastText on all the three datasets evaluated. Consequently, we suggest using this vector space for encoding the input in tasks of emotion detection from the text considering that we obtained on average an increase of the 2% in final results.

In order to support our claims of the robust accuracy of the model configured for the use of FastText word-embeddings, we performed the McNemar's Test for Classifiers. We observed that the differences in classification accuracy are statistically significant at p-value < 0.05 considering the pairs results obtained evaluating

the configuration FastText with the others for the datasets SemEval 2019 task 3 and SemEval 2018 task 1.

6 CONCLUSION

The task of identifying emotions from text plays an important role in customization systems. In this respect, tools are needed to obtain accurate annotations with a significant level of granularity. To meet this need, we have proposed a text identification model based on the use of deep neural networks LSTM and CNN mediated through the use of a level of attention. The results demonstrated the effectiveness of the approach on three different state-of-the-art datasets. Moreover, it was possible to measure the influence that the choice of the technique of encoding the textual input through word embedding influences the final result of the whole system. It has been demonstrated that for this architecture the FastText vector space allows obtaining the best results for the identification of the emotion. Considering the solidity of the model, it is proposed as a starting point for future work in the field. The results obtained allow to consider the analysis of the various textual data produced online as a technique close to consolidation. This makes it suitable for use in personalized systems capable of showing emotional intelligence. It is evident that the answers, suggestions, and way of approaching the system could be strongly influenced by the emotion identified in the user also using contents produced in text format. This process of humanizing artificial intelligence will be increasingly present with the massive spread of social robots and therefore is one of the possible future ways of application of this work.

In order to make everything easily replicable, the sources are publicly accessible at the following GitHub link:
https://github.com/marcopoli/emofinder

7 ACKNOWLEDGMENT

This research has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement N. 691071.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
[2] Alexandra Balahur, Jesus M Hermida, and Andres Montoyo. 2012. Building and exploiting emotinet, a knowledge base for emotion detection based on the appraisal theory model. *IEEE Transactions on Affective Computing* 3, 1 (2012), 88–101.
[3] Pierpaolo Basile, Valerio Basile, Danilo Croce, and Marco Polignano. 2018. Overview of the EVALITA 2018 Aspect-based Sentiment Analysis task (ABSITA). *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITAâĂŹ18), Turin, Italy. CEUR. org* (2018).
[4] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3, Feb (2003), 1137–1155.
[5] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.
[6] Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. SemEval-2019 Task 3: EmoContext: Contextual Emotion Detection in Text. In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval-2019)*. Minneapolis, Minnesota.
[7] Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733* (2016).
[8] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).
[9] François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.
[10] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. ACM, 160–167.
[11] Cicero Dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. 69–78.
[12] Paul Ekman and Dacher Keltner. 1997. Universal facial expressions of emotion. *Segerstrale U, P. Molnar P, eds. Nonverbal communication: Where nature meets culture* (1997), 27–46.
[13] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.
[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[15] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188* (2014).
[16] Edward Chao-Chun Kao, Chun-Chieh Liu, Ting-Hao Yang, Chang-Tai Hsieh, and Von-Wun Soo. 2009. Towards text-based emotion detection a survey and possible improvements. In *Information Management and Engineering, 2009. ICIME'09. International Conference on*. IEEE, 70–74.
[17] Yann LeCun et al. 1989. Generalization and network design strategies. *Connectionism in perspective* (1989), 143–155.
[18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[19] Saif M. Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 Task 1: Affect in Tweets. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*. New Orleans, LA, USA.
[20] Saif M Mohammad and Svetlana Kiritchenko. 2015. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence* 31, 2 (2015), 301–326.
[21] Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)*. 1–18.
[22] Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval* 2, 1–2 (2008), 1–135.
[23] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.
[24] Marco Polignano. 2015. The Infuence of User's Emotions in Recommender Systems for Decision Making Processes.. In *DC@ CHItaly*. 58–66.
[25] Mohammed Abdel Razek and Claude Frasson. 2017. Text-Based Intelligent Learning Emotion System. *Journal of Intelligent Learning Systems and Applications* 9, 01 (2017), 17.
[26] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. *Learning internal representations by error propagation*. Technical Report. California Univ San Diego La Jolla Inst for Cognitive Science.
[27] James A Russell. 1980. A circumplex model of affect. *Journal of personality and social psychology* 39, 6 (1980), 1161.
[28] Kashfia Sailunaz, Manmeet Dhaliwal, Jon Rokne, and Reda Alhajj. 2018. Emotion detection from text and speech: a survey. *Social Network Analysis and Mining* 8, 1 (2018), 28.
[29] Klaus R Scherer and Harald G Wallbott. 1994. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of personality and social psychology* 66, 2 (1994), 310.
[30] Marko Tkalcic, Andrej Kosir, and Jurij Tasic. 2011. Affective recommender systems: the role of emotions in recommender systems. In *Proc. The RecSys 2011 Workshop on Human Decision Making in Recommender Systems*. Citeseer, 9–13.
[31] Vincent Van Asch. 2013. Macro-and micro-averaged evaluation measures [[basic draft]]. *Belgium: CLiPS* (2013).
[32] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2018. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318* (2018).
[33] Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 4 (2018), e1253.
[34] Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In *European Semantic Web Conference*. Springer, 745–760.
[35] Xu Zhe and AC Boucouvalas. 2002. Text-to-emotion engine for real time internet communication. In *Proceedings of International Symposium on Communication Systems, Networks and DSPs*. Citeseer, 164–168.
[36] Guineng Zheng, Subhabrata Mukherjee, Xin Luna Dong, and Feifei Li. 2018. OpenTag: Open attribute value extraction from product profiles. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1049–1058.