SINAI at IEST 2018: Neural Encoding of Emotional External Knowledge for Emotion Classification

Flor Miriam Plaza-del-Arco[†] **Eugenio Martínez-Cámara**[♣]
M. Teresa Martín-Valdivia[†] **L. Alfonso Ureña- López**[†]
[†]Department of Computer Science,
Advanced Studies Center in ICT (CEATIC)
Universidad de Jaén, Campus Las Lagunillas, 23071, Jaén, Spain
[♣]Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI)
University of Granada, Spain
{fmplaza, maite, laurena}@ujaen.es, emcamara@decsai.ugr.es

Abstract

In this paper, we describe our participation in WASSA 2018 Implicit Emotion Shared Task (IEST 2018). We claim that the use of emotional external knowledge may enhance the performance and the capacity of generalization of an emotion classification system based on neural networks. Accordingly, we submitted four deep learning systems grounded in a sequence encoding layer. They mainly differ in the feature vector space and the recurrent neural network used in the sequence encoding layer. The official results show that the systems that used emotional external knowledge have a higher capacity of generalization, hence our claim holds.

1 Introduction

Emotions play an important role in human beings due to what we notice and remember is not the mundane but events that evoke feelings like joy, sadness, surprise, and disgust. Emotions relate us to others as a form of interpersonal communication, they introduce us to the world as well as the motivational force for what is best and worst in human behavior.

Emotion mining is part of the Sentiment Analysis (SA) and consists of recognizing emotions mainly from text. According to Ekman (1992), the basic emotions expressed by humans are: joy, sadness, surprise, fear, disgust and anger. Emotion recognition is still in its infancy and still has a ling way to proceed (Yadollahi et al., 2017). The high rate at which users share their opinions on news articles, blogs, microblogs and social networking sites, make this king of media even more attractive to measure specific emotions towards current affairs. Recognizing emotion is extremely important for some text-based communication tools (Wu et al., 2006), e.g., the dialog system is a kind of human machine communication system that uses only text input and output. Recognizing the users emotional states enables the dialog system to change the response and answer types (Lee et al., 2002). Text is still the main communication tool on the Internet. In online chat, the users emotional states can be used to control the dialog strategy.

In this paper, we describe the four systems submitted to the IEST shared-task of the WASSA Workshop (Klinger et al., 2018). The shared task consists of predicting the implicit emotion expressed in a tweet, and the labels of emotion are: sadness, joy, disgust, surprise, anger, or fear. We tackled the challenge as a multi-classification task, and we claim that the use of emotional external knowledge may enhance the performance and the capacity of generalization of the classification systems. We submitted four systems based on deep learning. Two of them do not use emotional external knowledge, and the other two do. The official results show that those systems with emotional external knowledge have a higher capacity of generalization as we hypothesized.

The rest of the paper is organized as follows. Section 2 describes the dataset used by our systems. Section 3 presents the details of the proposed systems. Section 4 displays the results and analyses them. We conclude in Section 5 with remarks and future work.

2 Dataset

The evaluation dataset (Klinger et al., 2018) is annotated on a scale of six emotions, namely: sadness, joy, disgust, surprise, anger, or fear. To run our experiments, we used this dataset as follows. During pre-evaluation period, we trained our models on the *train* set, and evaluated our different approaches on the *dev* set. During evaluation period, we trained our models on the *train* and *dev* sets,

Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 195–200
Brussels, Belgium, October 31, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

Dataset	Tweets
Train	153383
Dev	9591
Test	28757

Table 1: Number of tweets for each dataset.

and tested the model on the *test* set. The size of the datasets is in Table 1.

3 System description

The aim of the shared task is the classification of the implicit emotion of an input tweet. However, the word that explicitly expresses the emotion was remove from the input tweets. Accordingly, two specific features may be incorporated in the classification system: (1) the position of the removed word with emotion meaning; and (2) emotional external knowledge. Since our claim is that the use of emotional external knowledge can enhance the classification of emotions, we only considered the second specific feature, namely the incorporation of emotional external knowledge.

We designed a neural architecture built upon a sequence encoding approach, which is able to perform the classification with or without emotional external knowledge. We submitted four systems, which share a common structure composed of three modules: (1) language representation or features lookup module; (2) sequence encoding module; and (3) non linear classification module. The four systems differ in the first and second modules. The details of the modules and the differencies of the four systems are described in the following subsections.

3.1 Features lookup module

Regarding our claim, we defined a feature vector space for the training and the evaluation that is composed of: (1) unsupervised vectors of word embeddings; and (2) one-hot vector representation of emotional features.

Vectors of word embeddings A set of vectors of word embeddings is the representation of the ideal semantic space of words in a real-valued continuous vector space, hence the relationships between vectors of words mirror the linguistic relationships of the words. Vectors of word embeddings are a dense representation of the meaning of a word, thus each word is linked to a real-valued continuous vector of dimension d_{emb}.

There are freely available several pre-trained sets of vectors of word embeddings grounded in different approaches to represent the context of a word, such as C&W (Collobert et al., 2011), word2vect (Mikolov et al., 2013) and Glove (Pennington et al., 2014). Since the genre of the input documents is social media, Twitter, the use of a set of embeddings trained on tweets is advisable. Therefore, we specifically used the set of pre-trained vectors of word embeddings of Glove[1] that is trained on tweets. The most relevant characteristics of that set are: (1) the size of the vocabulary is 1.2 million of words; (2) all the words are lowercase.

Emotional external knowledge Two of the submitted systems used emotional external knowledge. We encoded the external emotional knowledge with a one-hot encoding approach, hence the emotional categories considered were represented as a one-hot vector. Accordingly, the feature vector space is enlarged with the size of the additional components or dimensions corresponding to the emotional categories (d=d_{emb}+d_{emo}).

To obtain the emotional external knowledge we use the following emotional lexicons:

NRC Word-Emotion Association Lexicon (EmoLex) (Mohammad and Turney, 2010). This lexicon has a list of English words associated to one or more of the following emotions: anger, fear, anticipation, trust, surprise, sadness, disgust, joy. Since the emotional external knowledge is encoded as one-hot vector, the corresponding emotion is set to 1 of those words that are in the lexicon. The results is a vector of eight emotion values. In case the word belongs to one or more emotions, all the emotions to which it belongs are taken into account.

Emoji lexicon We use this lexicon to identify the emojis present in text using some faces of an emoji lexicon.[2] This lexicon contains a list of emojis but it is not labeled with emotion. Thus, we manually annotated some emojis to one of the Ekman emotions: joy, anger, fear, disgust, surprise, sadness (Ekman, 1992). After this process, we obtained

[1] https://nlp.stanford.edu/projects/glove/

[2] https://github.com/erunion/emoji-lexicon

a lexicon with 72 emojis labelled by the Ekman emotions. The distribution of emojis by Ekman emotions is shown in Table 2.

Emotion	Number of Emojis
Joy	39
Sad	15
Anger	8
Fear	6
Surprise	3
Disgust	1

Table 2: Number of emojis for each Ekman emotion.

We tokenized the input tweets with the Twitter-aware tokenizer of NLTK[3] in order to project them in the feature vector space defined by the vector of word embeddings and emotional features. Consequently, each tweet (*t*) is transformed in a sequence of *n* words ($w_{1:n} = \{w_1, \ldots, w_n\}$). The size of the input sequence (*n*) was defined by the mode of the lengths of the inputs in the training data, hence sequences shorter than *n* were truncated. After the tokenization, the first layer of our architecture model is an feature lookup layer, which makes the projection of the sequence of tokens into the feature vector space. Therefore, the output of the features lookup layer is the matrix $\mathbf{WE} \in \mathbb{R}^{\mathrm{d \times n}}$, $\mathbf{WE^T_{1:n}} = (\mathbf{we_1}, \ldots, \mathbf{we_n})$, where $\mathbf{we_i} \in \mathbb{R}^{\mathrm{d}}$. The parameters of the embedding lookup layer are not updated during the training.

3.2 Sequence encoding module

The aim of the sequence encoding layer is the generation of high level features, which condense the semantic meaning of the entire sentence. We used an RNN layer because RNNs can represent sequential input in a fixed-size vector and paying attention to the structured properties of the input (Goldberg, 2017). RNN is defined as a recursive *R* function applied to a input sequence. The input of the function R is an state vector $\mathbf{s_{i-1}}$ and an element of the input sequence, in our case a word vector ($\mathbf{we_i}$). The output of *R* is a new state vector ($\mathbf{s_i}$), which is transformed to the output vector $\mathbf{y_i}$ by a deterministic function *O*. Equation 1 summarizes the former definition.

$$\begin{aligned} \mathrm{RNN}(\mathbf{we_{1:n}}, \mathbf{s_0}) &= \mathbf{y_{1:n}} \\ \mathbf{y_i} &= O(\mathbf{s_i}) \\ \mathbf{s_i} &= R(\mathbf{we_i}, \mathbf{s_{i-1}}); \end{aligned} \tag{1}$$

$$\mathbf{we_i} \in \mathbb{R}^{\mathrm{d_{in}}}, \mathbf{s_i} \in \mathbb{R}^{\mathrm{f(d_{out})}}, \mathbf{y_i} \in \mathbb{R}^{\mathrm{d_{out}}}$$

From a linguistic point of view, each vector ($\mathbf{y_i}$) of the output sequence of an RNN condenses the semantic information of the word w_i and the previous words ($\{w_1, \ldots, w_{i-1}\}$). However, according to the distributional hypothesis of language (Harris, 1954), semantically similar words tend to have similar contextual distributions, roughly speaking, the meaning of a word is defined by its contexts. An RNN can only encode the previous context of a word when the input of the RNN is the sequence $\mathbf{we_{1:n}}$. However, the input of the RNN can be also the reverse of the previous sequence ($we_{n:1}$). Consequently, we can elaborate a composition of two RNNs, the first one encode the sequence from the beginning to the end (*forward*, *f*), and a second one from the end to the beginning (*backward*, *b*), therefore the previous and the following context of a word is encoded. This elaboration is known as *bidirectional* RNN (biRNN), whose definition is in Equation 2.

$$\begin{aligned} \mathrm{biRNN}(\mathbf{we_{1:n}}) = [&\mathrm{RNN}^f(\mathbf{we_{1:n}}, s_0^f); \\ &\mathrm{RNN}^b(\mathbf{we_an : 1}, s_0^b)] \end{aligned} \tag{2}$$

The four systems submitted are based on the use of a specific gated-architecture of RNN, namely Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). Figure 1 shows the architecture of these models. The specific details of the sequence encoding layer of each submitted system are described as what follows.

NoEMoLSTM The sequence encoding layer is composed of one LSTM RNN. The input is the feature space only defined by the matrix of vectors of word embeddings ($\mathbf{WE} \in \mathbb{R}^{\mathrm{d_{emb}}}$). The output is all the output vectors of all the words of the sequence, hence the output is the sequence $\mathbf{y_{1:n}}$, $y_i \in \mathbb{R}^{\mathrm{d_{out}}}$.

EmoLSTM This sequence encoding layer is similar to the previous one, however the input is a feature space composed of vectors of word embeddings and emtotional features, mathematically

[3] https://www.nltk.org/api/nltk.tokenize.html#nltk.tokenize.casual.TweetTokenizer

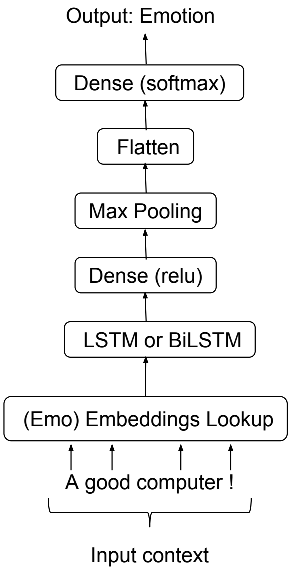

Figure 1: Neural model, where Emo represents the EmoLSTM and EmoBiLSTM models.

$\mathbf{WE} \in \mathbb{R}^{\mathrm{d_{emb}+d_{emo}}}$. As NoEMoLSTM, the output is all the output vectors of all the words of the sequence, hence the output is the sequence $\mathbf{y_{1:n}}$, $y_i \in \mathbb{R}^{\mathrm{d_{out}}}$.

NoEMoBiLSTM It only differs from NoEMoLSTM in the RNN layer. In this case the RNN layer is an BiLSTM layer. Since an BiLSTM is the composition of two LSTMs, the output units returned of NoEMoBiLSTM is larger than the one of NoEMOLSTM, specifically $\mathbf{y_{1:n}}$, $y_i \in \mathbb{R}^{\mathrm{d_{out}} \cdot 2}$.

EmoBiLSTM As EmoLSTM, it incorporates emotional external knowledge ($\mathbf{WE} \in \mathbb{R}^{\mathrm{d_{emb}+d_{emo}}}$), and as NoEMoBiLSTM, the encoding layer is an BiLSTM RNN, therefore the output is the sequence $\mathbf{y_{1:n}}$, $y_i \in \mathbb{R}^{\mathrm{d_{out}} \cdot 2}$.

3.3 Non linear classification module

The sequence representation of the tweets is then classified by two fully connected layers with ReLU as activation function, and additional layer activated by the softmax function. The layers activated by ReLU have different hidden units or output neurons ($dense^1$ and $dense^2$, see Table 3). With the aim of selecting the most relevant features, the output of the first full connected layer is processed by a max pooling layer. The size of the pooling layer was 2, and it was applied to every step, i.e. the strides size is 1. Since the four sequence encoding layers return an output sequence $y_{1:n} \in \mathbb{R}^{\mathrm{n \times d_{out}}}$, after the max pooling layer, the sequence is flattened to a single vector $\mathbf{y} \in \mathbb{R}^{\mathrm{n \cdot d_{out}}}$. The number of hidden units of the softmax layer matches the number of emotion categories of the task.

Hyper	NoEMo- LSTM	EMo- LSTM	NoEMo- BiLSTM	EMo- BiLSTM
n	27	27	27	27
d_{emb}+d_{emo}	200	214	200	214
d_{out}	128	128×2	128	128×2
$dense^1$	128	128	128	128
$dense^2$	64	64	64	64
dr^1	0.5	0.5	0.5	0.5
dr^2	0.5	0.5	0.5	0.5
L_2 r	0.0001	0.0001	0.0001	0.0001

Table 3: Hyperparater values of the systems submitted.

In order to avoid overfitting, we add a dropout layer (Hinton et al., 2012) after each fully connected layer with a dropout rate value *dr*. Besides, we applied an L_2 regularization function to the loss function with a regularization value (*r*). Moreover, the training is stopped in case the loss value does not improve in 3 epochs.

The training of the network was performed by the minimization of the cross entropy function, and the learning process was optimized with the Adam algorithm (Kingma and Ba, 2015) with its default learning rate. The training was performed following the minibatches approach with a batch size of 64, and the number of epochs was 30.

For the sake of the replicability of the experiments, Table 3 shows the values of the hyperparaments of the network, and the source code of our experiments is publicly available.[4]

3.4 Internal baseline

We also developed an internal baseline for evaluating the performance of the neural networks models. Our internal baseline was a linear classification system, namely Support Vector Machines (SVM). The feature space is composed of the unigrams of the training set weighted by the TF-IDF metric, and the number of unigrams of each emotional category considered. The results reached by our internal baseline are in Table 4 with the name EmoSVM.

[4] https://github.com/fmplaza/WASSA-2018

	Development			Test		
System	M. Prec	M. Recall	M. F1	M. Prec	M. Recall	M. F1
Amobee^1	-	-	-	-	-	71.4%
EmoSVM	49.99%	50.1%	50.02%	49.77%	49.84%	49.69%
EmoLSTM^{21}	56.11%	56.13%	55.96%	58.41%	58.3%	58.30%
NoEMoLSTM^{21}	56.60%	56.46%	56.43%	58.25%	58.17%	58.11%
EmoBiLSTM^{22}	56.42%	56.3%	56.17%	57.92%	57.97%	57.94%
NoEMoBiLSTM^{26}	55.17%	55.24%	55.12%	56.41%	54.7%	54.68%

Table 4: Dev. and Test results reached by our systems and by the best system in the competition. The superscripts specify the rank of the systems in the competition.

4 Analysis of results

We performed our experiments in the pre-evaluation phase and evaluation phase, and we used the official competition metric macro-averaged F1-score as evaluation measure. Moreover, we computed the Macro Precision, Macro Recall and Macro F1. This results are shown in Table 4.

First, we assessed our internal baseline (EmoSVM), which set a lower bound during the development or pre-evaluation phase. Since the number of classes is 6, the performance of our internal baseline is acceptable, and we also conclude the its generalization capacity is acceptable.

The submitted systems, which are based on deep learning, outperformed EmoSVM with the development and test data. Regarding the results reached with the development subset, the models that did not use emotional external knowledge reached higher results that those ones that used external knowledge. However, those systems that used emotional external knowledge (EmoLSTM and EmoBiLSTM) outperformed the ones that did not use external knowledge on the evaluation data, which means that the emotional external knowledge enhance the capacity of generalization of the classification models, as we expected.

Regarding the performance of LSTM and BiLSTM, the use of LSTM as sequence encoding module resulted in a higher capacity of generalization, because BiLSMT reached higher results with the development data, but LSTM reached higher results with the evaluation data. Therefore, we conclude that the use of LSTM as sequence encoding module and emotional external knowledge allow to reach good results in the task of emotion classification, which allow us to confirm our claim.

Finally, the best system of the task (Amobee) has reached 71.4% of F1-score in the evaluation phase, since we do not know the evaluation measures corresponding to the pre-evaluation phase, in Table 4 they appear as "-". Our best system (EmoLSTM) hold the 22 position in the ranking with 58.3% in F1-score.

5 Conclusions

We described the participation of the SINAI lab in the IEST shared task of the WASSA Workshop. We submitted four systems based on deep learning. The systems mainly differ in the sequence encoding module, and the feature vector space. We compare the performance of LSTM and BiLSTM as sequence encoding module, and the use of emotional external knowledge. The results show that the use of LSTM and emotional external knowledge give a higher capacity of generalization to the classification model.

As future work, we will study how to improve the use of external knowledge in the task of emotion classification, as well as, how to automatically select the most relevant features. Hence, we will study the development of an Attention module in our models. Furthermore, we will research how to add relevant linguistic information to our models, as the influence of negation in the classification of emotions.

Acknowledgments

This work has been partially supported by Fondo Europeo de Desarrollo Regional (FEDER), REDES project (TIN2015-65136-C2-1-R) and SMART-DASCI project (TIN2017-89517-P) from the Spanish Government. Eugenio Martínez

Cámara was supported by the Juan de la Cierva Formación Programme (FJCI-2016-28353) from the Spanish Government.

References

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Paul Ekman. 1992. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200.

Yoav Goldberg. 2017. *Neural Network Methods for Natural Language Processing*. Morgan & Claypool Publishers.

Zellig S. Harris. 1954. Distributional structure. *WORD*, 10(2-3):146–162.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference for Learning Representations, San Diego, 2015*.

Roman Klinger, Orphée de Clercq, Saif M. Mohammad, and Alexandra Balahur. 2018. Iest: Wassa-2018 implicit emotions shared task. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Brussels, Belgium. Association for Computational Linguistics.

Chul Min Lee, Shrikanth S Narayanan, and Roberto Pieraccini. 2002. Combining acoustic and language information for emotion recognition. In *Seventh International Conference on Spoken Language Processing*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Saif M Mohammad and Peter D Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 26–34. Association for Computational Linguistics.

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)*, pages 1532–1543. Association for Computational Linguistics.

Chung-Hsien Wu, Ze-Jing Chuang, and Yu-Chung Lin. 2006. Emotion recognition from text using semantic labels and separable mixture models. *ACM transactions on Asian language information processing (TALIP)*, 5(2):165–183.

Ali Yadollahi, Ameneh Gholipour Shahraki, and Osmar R Zaiane. 2017. Current state of text sentiment analysis from opinion to emotion mining. *ACM Computing Surveys (CSUR)*, 50(2):25.