TASS 2018: Workshop on Semantic Analysis at SEPLN, septiembre 2018, págs. 117-123

SCI^{2}S at TASS 2018: Emotion Classification with Recurrent Neural Networks

SCI^{2}S en TASS 2018: Clasificación de Emociones con Redes Recurrentes Neuronales

Nuria Rodríguez Barroso, Eugenio Martínez-Cámara, Francisco Herrera
Instituto de Investigación Andaluz en Ciencia de Datos e Inteligencia Computacional (DaSCI)
Universidad de Granada, España
rbnuria@gmail.com, {emcamara, fherrera}@decsai.ugr.es

Abstract: In this paper, we describe the participation of the team SCI^{2}S in all the Subtasks of the Task 4 of TASS 2018. We claim that the use of external emotional knowledge is not required for the development of an emotional classification system. Accordingly, we propose three Deep Learning models that are based on a sequence encoding layer built on a Long Short-Term Memory gated-architecture of Recurrent Neural Network. The results reached by the systems are over the average in the two Subtasks, which shows that our claim holds.

Keywords: Deep Learning, Recurrent Neuronal Networks, LSTM, Emotion Classification

Resumen: En este artículo se presenta la participación del equipo SCI^{2}S en la Tarea 4 de TASS 2018. Partiendo de la asunción de que no es necesario el uso de características emocionales para el desarrollo de un sismtema de clasificación de emocoiones, se proponen tres modelos de redes neuronales basados en el uso de una capa de Red Recurrente Neuronal de tipo Long Short-Term Memory. Los sistemas han alcanzado una posición por encima de la media en las dos Subtareas en las que se ha participado, lo cual ha permitido confirmar nuestra hipótesis.

Palabras clave: Redes Neuronales, Redes Neuronales Recurrentes, LSTM, Clasificación de Emociones

1 Introduction

People usually have a look at advertisements when they read traditional newspapers. These advertisements generally fit the news that are in the same, previous or next page, because the match of the news and the ads are carefully decided during the edition time, which is before the printing of the newspaper. Nowadays, online newspapers are as read as traditional ones, hence companies also want to show their brands in online newspapers, and they invest money to buy ads in them. However, one of the differences between traditional and on-line newspapers is the moment when the correspondence between the news and the advertisements is done, which is in reading time. Thus, the news and the ads likely do not match.

The lack of correspondence between a news and a advertisement means that the topic of the news is not suitable for the advertisement, or the emotion that may elicit from the reader is not positive. If the readers are disgusted by the news, they may be revolted by the advertisement too, which is highly detrimental for the brand advertised. The advertising spots in online newspapers are fixed beforehand, and the advertisement that appears in each spot does not depend on the decision of the editor or the journalist, but it depends on a automatic broadcasting system of ads of an online marketing company. Consequently, companies are not able to control whether the reputation of its brands may be damaged, which is known by marketing experts as the brand safety issue.[1]

The Task 4 of TASS 2018 (Martínez-Cámara et al., 2018) is focused on the mentioned issue of brand safety, and it proposes the classification if a news is sure for a brand according to the emotion elicited from the readers when they read the headline of a news. The organization provided an anno-

[1] https://www.thedrum.com/opinion/2018/07/09/brand-safety-the-importance-quality-media-fake-news-and-staying-vigilant

ISSN 1613-0073 Copyright © 2018 by the paper's authors. Copying permitted for private and academic purposes.

CEUR-WS.org/Vol-2172/p14_sci2s_tass2018.pdf

tated corpus of headlines of news of Spanish written newspapers from around the world, so the corpus SANSE is a global representation of the written Spanish language. In this paper, we present the systems submitted by the SCI^{2}S team to the two Subtasks of Task 4 of TASS 2018.[2]

We claim that the emotional classification can be tackled without the use of emotional features or any other kind of handcrafted linguistic feature. We thus propose the generation of dense high quality features following a sentence encoding approach, and then the use of a non lineal classifier. We submitted three systems based on the encoding of the input headline with a Recurrent Neural Network (RNN) Long Short Term Memory (LSTM). Our submitted systems are over the average in the competition, hence this fact shows that our claim holds.

2 Architecture of the models

The organization proposed two Subtasks, the first one is defined in a monolingual context, and the second in a multilingual one. The first Subtask has two levels of evaluation, which differ in the size of the evaluation set. We designed the neural architecture without taking into account the specific characteristics of the Subtasks, because our aim was the evaluation of our claim on the SANSE corpus.

The architecture of the three systems submitted is composed of three modules: (1) language representation, for the sake of simplicity embeddings lookup module; (2) sequence encoding module, in which the three architectures differ; and (3) non lineal classification. The details of each module are explained in the following subsections.

2.1 Embeddings lookup layer

Regarding our claim, we defined a feature vector space for the training and the evaluation that is composed of unsupervised 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 different algorithms to build vectors of word embeddings in the literature, standing out C&W (Collobert et al., 2011), word2vect (Mikolov et al., 2013) and Glove (Pennington, Socher, and Manning, 2014). Likewise, several sets of pre-trained vectors of word embeddings built using the previous algorithms are freely available. However, those pre-trained sets were generated using documents written in English, thus they cannot been used for representing Spanish words.

We used the pre-trained set of word embeddings SBW[3] (Cardellino, 2016). SBW was built upon several Spanish corpora, and the most relevant characteristics of its development were: (1) the capitalization of the words were kept unchanged; (2) the word2vect algorithm used was skip-gram; (3) the minimum allowed word frequency was 5; and (4) the dimension or components of the word vectors is 300 ($d_{emb} = 300$).

We tokenized the input headlines with the default tokenizer of NLTK[4] in order to project them in the feature vector space defined by the vector of word embeddings. Consequently, each headline (h) 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 maximum length 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 embedding lookup layer, which makes the projection of the sequence of tokens into the feature vector space. Therefore, the output of the embeddings lookup layer is the matrix $\mathbf{WE} \in \mathrm{I\!R}^{\mathrm{d,n}}$, $\mathbf{WE^T_{1:n}} = (\mathbf{we_1}, \ldots, \mathbf{we_n})$, where $\mathbf{we_i} \in \mathrm{I\!R}^{\mathrm{d}}$. The parameters of the embedding lookup layer are not updated during the training.

2.2 Sequence encoding layer

The aim of the sequence encoding layer is the generation of high level features, which condense the semantic 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

[2] The details about the Task 4 of TASS 2018 are in (Martínez-Cámara et al., 2018).

[3] `https://crscardellino.github.io/SBWCE/`

[4] `https://www.nltk.org/api/nltk.tokenize.html`

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[5] summarizes the former definition.

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

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

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, or in other words, 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 encodes 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.

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

The three systems submitted are based on the use of a specific gated-architecture of RNN, namely LSTM (Hochreiter and Schmidhuber, 1997), which has reached strong results in several Natural Language Processing tasks (Tang, Qin, and Liu, 2015; Kiperwasser and Goldberg, 2016; Martínez-Cámara et al., 2017). The specific details of the sequence encoding layer of each submitted system are described as what follows.

Single LSTM (SLSTM). The layer is composed of one LSTM, whose input is the sequence $\mathbf{we_{1:n}}$, and its output is composed of a single vector, namely the last output vector ($\mathbf{y_n} \in \mathbb{R}^{\mathrm{d_{out}}}$). In this case, the semantic information of the entire headline is condensed in the last vector of the LSTM, which correspond to the last word.

Single biLSTM (SbLSTM). In order to encoded the previous and forward context of the words of the input sequence, the sequential encoding layer of this system is a biLSTM. The output is the concatenation of the last output vector of the two LSTMs of the biLSTM ($\mathbf{y_n} = [\mathbf{y_n^f}; \mathbf{y_n^b}] \in \mathbb{R}^{2\times \mathrm{d_{out}}}$).

Sequence LSTM (SeLSTM). The encoding is carried out by an LSTM, but the output is composed of all output vectors of all the words of the sequence, hence the output is not a vector, but the sequence $\mathbf{y_{1:n}}$, $y_i \in \mathbb{R}^{\mathrm{d_{out}}}$.

The semantic information returned by SeLSTM is greater than the other two layers, because it returns the output vector of each word, therefore the subsequent layers receive more semantic information from the sequence encoding layer.

2.3 Non lineal classification layer

Since RNN and specifically LSTM has the ability of encoding the semantic information of the input sequence, the output of the sequence encoding layer is a high level representation of the semantic information of the input headline.

The sequence representation of the headline is then classified by three 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 (see Table 1). The SeLSTM layer does not return an output vector, but an output sequence $y_{1:n} \in \mathbb{R}^{\mathrm{n,d_{out}}}$. Thus, after the second fully connected layer, the sequence is flattened to a single vector $\mathbf{y} \in \mathbb{R}^{\mathrm{n\times d_{out}}}$. Since the task is a binary classification task, the number of hidden units of the softmax layer is 2.

In order to avoid overfitting, we add a dropout layer after each fully connected layer with a dropout rate value (dr). Besides, we applied an $\mathrm{L_2}$ regularization function to the output of each fully connected layer with a

[5] The definition of RNN states that the dimension of $\mathbf{s_i}$ is a function of the output dimension, but some architectures as LSTM does not allow that flexibility.

regularization value (r). Moreover, the training is stopped in case the loss value does not improve in 5 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 25, and the number of epochs was 40.

For the sake of the replicability of the experiments, Table 1 shows the values of the hyperparaments of the network, and the source code of our experiments is publicly available.[6]

Hyper. value	SLSTM	biLSTM	SeLSTM
n	20	20	20
d_{emb}	300	300	300
d_{out}	512	256×2	512
dr^1	0.35	0.35	0.35
dr^2	0.35	0.35	0.5
dr^3	0.5	0.5	0.5
L_2 r^1	0.0001	0.0001	0.0001
L_2 r^2	0.001	0.001	0.001
L_2 r^3	0.01	0.01	0.01

Table 1: Hyperparameter values of the systems submitted

3 Results and Analysis

The organization provided a development set of the SANSE corpus with the aim that the teams would use the same data to tune the classification models. We participated in the two levels of Subtasks 1 and in the Subtask 2, and we present in Tables 2, 3 and 4 the results reached with the development set (development time) and the official results with the test set of SANSE (evaluation time).

The main differences among the submitted systems are: (1) The semantic information encoded; and (2) the number of parameters. SLSTM is the model with less semantic information encoded, because the LSTM is only run in one direction, and the last output vector of the LSTM is only processed by the subsequent layers. Although SbLSTM encodes more semantic information than SLSTM, they have the same number of parameters, because SbLSTM only processes the last output vector of the sequence encoding layer as the SLSTM model. In contrast, the SeLSTM is the model that uses more parameters, because it processes the output vectors of the sequence encoding layer of each input word.

We expected that models with a higher number of parameters and capacity of encoding semantic information, they will reach higher results in the competition, or in other words, they will have a higher capacity of generalization. However, the comparison of the results reached on the development and test set shows a non expected performance. Regarding the two main differences among the models, we highlight the following two facts:

Generalization capacity. The model that reached a higher results in the two levels of the Subtask 1 is SLSTM. The performance of SLSTM stands out in the second level of Subtask 1, because it is the second higher ranked system. Since the test set of the second level is larger than the level one, it demands a higher generalization capacity from the systems, thus the good performance of SLSTM is more relevant. In contrast, SbLSTM and SeLSTM are in the fifth and sixth position respectively in the second level, and the sixth and seventh position in the first level of Subtask 1, which was not expected due to they have more parameters and condense more semantic information.

Concerning the Subtask 2, the results reached were the expected ones, because SeLSTM, which has more parameters and condense more semantic information, reached the best results among our three systems. The generalization demand in this task is high too, because the language or the domain of the training and the test sets and different, because the training set is composed of headlines written in the Spanish language used in America, and the test set is written in the Spanish language used in Spain.

Although the generalization capacity of our systems is high, the different performance in Subtask 1 and Subtask 2 allow us to conclude that to reach a good generalization capacity, a balance between the number of parameters and the complexity or depth of the neural network is required as it is also asserted in (Conneau et al., 2017).

Differences among datasets. SLSTM and SbLSTM reached a value of Macro Recall

[6] https://github.com/rbnuria/TASS-2018

	Development				Test (official)			
System	M. Prec.	M. Recall	M. F1	Acc.	M. Prec.	M. Recall	M. F1	Acc.
SLSTM[1]	73.89	74.74	74.10	74.80	78.40	76.40	77.40	78.60
SbLSTM[3]	75.24	75.15	75.19	76.40	77.40	75.20	76.30	77.60
SeLSTM[2]	76.08	76.35	76.21	77.20	76.30	76.50	76.40	77.20

Table 2: The macro-average and accuracy values in % reached by the three systems on the development and test sets in the Subtask 1, level 1. The superscript is the official rank (ranked by the M. F1 value) among the three submitted systems in the official results

	Development				Test (official)			
System	M. Prec.	M. Recall	M. F1	Acc.	M. Prec.	M. Recall	M. F1	Acc.
SLSTM[1]	73.89	74.74	74.10	74.80	88.80	86.70	87.30	88.80
SbLSTM[2]	75.24	75.15	75.19	76.40	86.80	85.70	86.30	87.80
SeLSTM[3]	76.08	76.35	76.21	77.20	83.80	87.00	85.30	85.30

Table 3: The macro-average and accuracy values in % reached by the three systems on the development and test sets in the Subtask 1, level 2. The superscript is the official rank (ranked by the M. F1 value) among the three submitted systems in the official results

	Development				Test (official)			
System	M. Prec.	M. Recall	M. F1	Acc.	M. Prec.	M. Recall	M. F1	Acc.
SLSTM[3]	74.54	72.05	72.67	75.00	68.30	66.10	67.20	70.00
SbLSTM[2]	75.60	71.14	71.87	75.90	67.90	67.20	67.60	69.80
SeLSTM[1]	72.47	69.41	69.98	77.20	68.70	67.80	68.30	63.11

Table 4: The macro-average and accuracy values in % reached by the three systems on the development and test sets in the Subtask 2. The superscript is the official rank (ranked by the M. F1 value) among the three submitted systems in the official results

higher than the value of Macro-Precision in the development set of Subtask 1 in the two levels of evaluation. However, they reached the inverse relation on the test set of both levels of Subtask 1. In contrast, SeLSTM had the same trend in both datasets, thus the performance of SeLSTM shows a higher stability. On the other hand, the three systems had the same performance in the development and test sets in Subtask 2, that it is to say, the value of Macro-Precision was higher than the value of Macro-Recall in development and evaluation time.

Regarding the differences between the datasets, the performance of models with more parameters and with more semantic information is more stable, which means that the results in development time follows a similar trend to the results in evaluation time that is an desirable characteristic of a classification system.

Regarding the competition, the rank position of our systems are in Table 5. In Subtask 1, the systems reached a rank position over the average, and SLSTM stands out in Level 2 of Subtask 1. In Subtask 2, the systems are on the average, and the performance is close to their competitors. Regarding our claim and the high results reached by the three systems, we conclude that our claim holds, hence we can obtain strong results in the task of emotion classification without the use of emotional features.

4 Conclusions

We described the three systems submitted to all the Subtasks of Task 4 of TASS 2018 by the team SCI^{2}S. Our proposal is based on the claim that emotional classification can be performed without the use of emotional external knowledge or handcrafted features. The three systems are three neural networks grounded in a sentence classifica-

	Rank		
System	Sub. 1, L1	Sub. 1, L2	Sub. 2
SLSTM	4/13	2/10	6/8
SbLSTM	7/13	5/10	5/8
SeLSTM	6/13	6/10	4/8

Table 5: Rank position of the submitted systems in the competition

tion approach, namely the use of an LSTM and a biLSTM. The three systems reached a rank position over the average in the two Subtasks of Task 4, thus we conclude that our claim holds.

Our future work will go in the direction defined by the analysis of the results (see Section 3), hence we are going to work in the study of the balance between the depth and the generalization capacity of our emotional classification model. Likewise, we will work in the addition of an Attention layer (Bahdanau, Cho, and Bengio, 2015) to the model, with the aim of automatically selecting the most relevant features.

Acknowledgements

This work was partially supported by the Spanish Ministry of Science and Technology under the project TIN2017-89517-P, and a grant from the Fondo Europeo de Desarrollo Regional (FEDER). Eugenio Martínez Cámara was supported by the Juan de la Cierva Formación Programme (FJCI-2016-28353) from the Spanish Government.

References

Bahdanau, D., K. Cho, and Y. Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference for Learning Representations, San Diego, 2015.*

Cardellino, C. 2016. Spanish Billion Words Corpus and Embeddings, March.

Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, November.

Conneau, A., H. Schwenk, L. Barrault, and Y. Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1107–1116. Association for Computational Linguistics.

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

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

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

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

Kiperwasser, E. and Y. Goldberg. 2016. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association of Computational Linguistics*, 4:313–327.

Martínez-Cámara, E., Y. Almeida-Cruz, M. C. Díaz-Galiano, S. Estévez-Velarde, M. A. García-Cumbreras, M. García-Vega, Y. Gutiérrez, A. Montejo Ráez, A. Montoyo, R. Muñoz, A. Piad-Morffis, and J. Villena-Román. 2018. Overview of TASS 2018: Opinions, health and emotions. In E. Martínez-Cámara, Y. Almeida-Cruz, M. C. Díaz-Galiano, S. Estévez-Velarde, M. A. García-Cumbreras, M. García-Vega, Y. Gutiérrez, A. Montejo Ráez, A. Montoyo, R. Muñoz, A. Piad-Morffis, and J. Villena-Román, editors, *Proceedings of TASS 2018: Workshop on Semantic Analysis at SEPLN (TASS 2018)*, volume 2172 of *CEUR Workshop Proceedings*, Sevilla, Spain, September. CEUR-WS.

Martínez-Cámara, E., V. Shwartz, I. Gurevych, and I. Dagan. 2017. Neural disambiguation of causal lexical markers based on context. In *IWCS 2017 – 12th International Conference on Computational Semantics – Short papers.*

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. 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*. Curran Associates, Inc., pages 3111–3119.

Pennington, J., R. Socher, and C. 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.

Tang, D., B. Qin, and T. Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432. Association for Computational Linguistics.