SINAI-DL at SemEval-2019 Task 7: Data Augmentation and Temporal Expressions

Miguel Ángel García-Cumbreras[1], Salud María Jiménez-Zafra[1], Arturo Montejo-Ráez[1], Manuel Carlos Díaz-Galiano[1], Estela Saquete[2]
[1]CEATIC / Universidad de Jaén
[1]{magc, sjzafra, amontejo, mcdiaz}@ujaen.es
[2]DLSI / Universidad de Alicante
[2]stela@dlsi.ua.es

Abstract

This paper describes the participation of the SINAI-DL team at RumourEval (Task 7 in SemEval 2019, subtask A: SDQC). SDQC addresses the challenge of rumour stance classification as an indirect way of identifying potential rumours. Given a tweet with several replies, our system classifies each reply into either supporting, denying, questioning or commenting on the underlying rumours. We have applied data augmentation, temporal expressions labeling and transfer learning with a four-layer neural classifier. We achieve an accuracy of 0.715 with the official run over reply tweets.

1 Introduction

Fake news has been identified as "news stories that have no factual basis but are presented as news" (Allcott and Gentzkow, 2017). On a conceptual level, we can define hoaxes or rumours as false information spread across social media with the intention to be picked up by traditional news websites (Rubin et al., 2015). Rumours have been around for millennia, as attested by the ancient (and modern) Greek word 'pheme', which means rumour or inaccurate information.

We have participated in RumourEval (SemEval 2019 Task 7, Subtask A), named *SDQC: Determining support for rumours*, with the complementary objective of tracking how other sources orient to the accuracy of the rumourous story by looking at the replies to the tweet that presented the rumourous statement (Gorrell et al., 2019).

These replies are extracted from Twitter and Reddit, but we have only processed the tweet replies, obtaining a good score in terms of accuracy.

The rest of the paper is organized as follows. Section 2 is a brief overview of the task. Data analysis is shown in section 3. Section 4 describes the neural network architecture. The experiments and results are analyzed in section 5. Finally, conclusions and proposals for further experimentation are provided in section 6.

2 Related Work

The previous edition of RumorEval was organized as part of the SemEval 2017 workshop. Thirteen systems were presented in that edition. Most of the systems presented face this task as a tweet classification task with four categories. Some participants use neural networks such as LSTM (Kochkina et al., 2017) and CNN (Chen et al., 2017; García Lozano et al., 2017), or SVM machine learning algorithm (Wang et al., 2017; Singh et al., 2017), using as main feature word embeddings. Most systems add lexical, syntactic and semantic features to word embeddings.

3 Data analysis

The data provided by the organization consist of a set of tweets and replies. Replies can be originated from two different sources: Reddit or Twitter. We have only worked with Twitter replies because features of Reddit replies and tweets are different, especially in regards to the length. In Table 1 we present the datasets distribution.

Dataset	Tweets	Replies	Tweets replies
train_EN	327	5,217	4,244
dev_EN	38	1,485	1,025
test_EN	56	1,746	1,010

Table 1: Datasets distribution (only tweets).

The objective of task A is to determine whether each reply supports, denies, queries or comments the rumour. The classification of tweet replies in each of the four categories established is shown in Table 2. We can conclude that although

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 1120–1124
Minneapolis, Minnesota, USA, June 6–7, 2019.

the labels show a realistic situation in terms of user comments, the classes are very unbalanced.

Category	train_EN	dev_EN	test_EN
Comment	2,897	779	771
Deny	335	70	92
Query	358	107	56
Support	634	69	91

Table 2: Tweets replies distribution.

In order to decide which window size to use in our system, we generated a cumulative histogram according to the different lengths of the tweets replies. Our objective was to select a size that could cover a high rate of tweets replies. In Table 3 we summarize the quantiles at 80 % and 90 % for the different datasets. Based on the values we decided to select a window size with 30 words because approximately 90 % of training and development tweet replies have a length of 30 words or less.

Data	Quantile 0.8	Quantile 0.9
train_EN	25	29
dev_EN	28	30

Table 3: Length of tweet replies covering 80 % and 90 % of cases.

4 System overview

Nowadays, deep neural architectures are populating the scientific playground in many scenarios: image recognition, speech recognition (Graves et al., 2013) and synthesis (Ze et al., 2013), and, of course, text classification (Zhang et al., 2015). But these supervised learning algorithms demand certain requirements that sometimes are difficult to meet. One of the most difficult to overcome in some cases is the need for a large and varied learning data set. When there is a lack of data, two main strategies can be followed: *transfer learning* and *data augmentation*.

4.1 Model description

We have implemented the proposed neural network using the Keras[1] library for Python, running on TensorFlow over an NVIDIA Titan X card. Each model took approximately 15 minutes to get trained and few seconds to classify development or test sets. The architecture of our neural network follows a sequence of layers as follows:

1. **First layer**: An embedding layer that is loaded with pre-trained weights, and converts each word into a 200-dimensional vector of real values.

2. **Second layer**: A bi-directional LSTM recurrent network with 512 activations and a dropout value of 0.5.

3. **Third layer**: A dense network with 128 activations and the *ReLU* function as activation function. A dropout of 0.5 is also applied after this network.

4. **Fourth layer**: last classification layer, with 4 activations on the final softmax function.

Figure 1 shows the neural network model with the four layers.

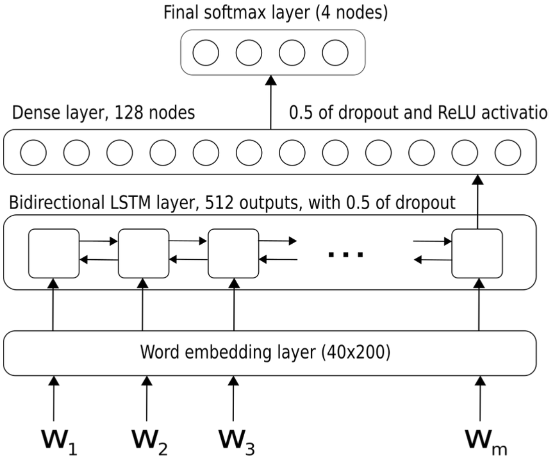
Final softmax layer (4 nodes)
Dense layer, 128 nodes
0.5 of dropout and ReLU activation
Bidirectional LSTM layer, 512 outputs, with 0.5 of dropout
...
Word embedding layer (40x200)
w_1
w_2
w_3
w_m

Figure 1: Neural network model.

The model has been trained with the hyperparameters values specified in Table 4 (ce means cross-entropy).

Parameter	value
Batch size	512
Loss function	categorical ce
Optimization algorithm	Adam
Sequence length	30 terms
No. Epochs	50

Table 4: Hyperparameters

The texts have been preprocessed as follows:

1. Lower case is applied.

2. Hashtags are split into several tokens according to a camel case approach. For example, “*#MeToo*” is converted into the terms “*<BOH>me too <EOH>*”.

[1] http://keras.io

3. Mentions are replaced by the token <*MENTION*>

4. Unknown terms (those not found in the embedding dictionary) are replaced by the token <*UNK*>.

5. A final token <*EOT*> is added at the end of the tweet.

We have taken an already trained word embedding matrix for the first layer, allowing the weights of the these learned model to get retrained during the learning process. We have used the weights from the GloVe Twitter model provided by the Stanford NLP Group, which is built over 2 billion tweets (27B tokens, 1.2M vocab, uncased, 200-dimensional vectors, 1.42 GB download).

4.2 Data augmentation

There are two important reasons for proposing data augmentation. On the one hand, deep neural models require a significant amount of training data to extract relevant features. On the other hand, as we can see in the section 3 there is a strong class imbalance between the samples (in dev and train datasets more than 75 % of the tweets are labeled as *comments*).

For each tweet, our system expands the information using paraphrasing. To express the same message with different words, we applied the online tool RewriterTools[2]. For instance, the paraphrase of the tweet *“EU’s hailed migrant plan ‘a road to Hell’ Czech Republic refuses involvement”* is *“EU’s hailed migrant layout ’a avenue to Hell’ Czech Republic refuses involvement”*.

4.3 Temporal expressions

As human beings, we tend to organize the flux in structured units known as events. Events take place at certain times, which are expressed in the text in the form of temporal expressions. However, these expressions are not always explicit dates that a computer is able to understand. For this reason, we decided to add a module that is capable of processing temporal information at the level of events and temporal expressions and annotate and resolve this information, so that it can be used in the detection of a rumor.

TimeML (Saurí et al., 2006) is the most standardized schema to annotate temporal information. They defined the event as “something that can be said to obtain or hold true, to happen or to occur”. This annotation schema annotates not only events and temporal expressions, but also temporal relations, known as links (Pustejovsky et al., 2003). Example below shows a sentence annotated with TimeML temporal expressions (TIMEX3), events (EVENT), and the links between them (TLINK).

```
John <EVENT
eid=’e1’>came</EVENT> on <TIMEX3
tid=’t1’>Monday</TIMEX3>
<TLINK eventInstanceID=’e1’
relatedToTime=’t1’
relType=’IS_INCLUDED’ />
```

In our approach, the Temporal Information Extraction and Processing was performed by TIPSem system (Temporal Information Processing using Semantics) (Llorens et al., 2013, 2012)[3]. TIPSem is able to automatically annotate all the temporal information according to TimeML standard annotation scheme (Saurí et al., 2006). In this first approach of the system, only the tags regarding temporal expressions and events have been considered and we will explore using the links as further work.

5 Experiments and results

We performed an evaluation of the proposed neural network on the development set, but training a model on two different official training sets: the official ones and those augmented by paraphrasing the given tweets. The results were discouraging when paraphrased tweets are added to the training set, as Table 5 shows. After checking the tweets generated by the paraphrasing tool, we noticed that the quality was low, with non-sense texts in some cases and few structural variations from the original tweet. Thus, we believe that the network was not even less robust, but worse as a non-realistic model was learned.

The detection of temporal expressions and the inclusion of the generated tags into the model didn’t report any improvement either. We believe that the related embeddings (randomly initialized) needed a far larger set to fit in the transferred learned embedding model for GloVe vectors.

Thus, our submission relies only on official training data which was, as we know, not enough data to ensure a good learning process. Anyhow, our system performed in 9th position out from 21 in

[2] https://www.rewritertools.com/paraphrasing-tool

[3] http://gplsi.dlsi.ua.es/demos/TIMEE/

train data	accuracy on dev data
official	0.690499
official + paraphrased	0.675808
official with temporal tags	0.684622

Table 5: Development experiments

subtask A, with an SDQC value of 0.3927 (F1-score).

Table 6 shows the results obtained with the official run over test set (only with the tweet replies).

truth label	accuracy	total	correct
all labels	**0.7148**	**1,066**	**762**
support	0.0219	91	2
deny	0.1413	92	13
query	0.4285	56	24
comment	0.9377	771	723

Table 6: Test experiments: official run

In the first analysis of results we can verify that the neural network system, on the base case, has worked correctly (almost perfect) for the *comment* class that have a sufficient number of examples of train and dev, and much worse for those with very few examples (classes support, deny and query).

We have to finish a more exhaustive analysis of these results, especially of the mislabelled samples. For instance, in the analysis of the truth label *support*, our system tags the most of the cases with the *comment* label. In this case, we can conclude that the *comment* label has been overtrained because of the greater number of examples (high bias).

6 Conclusions and future work

Our proposal explores how transferred embeddings and data augmentation may help in a text classification task like RumourEval. By augmenting official training data with paraphrasing, no improvement is noticed on classifying development data, due to the poor quality of the paraphrasing tool. So, we plan to explore other augmentation strategies, like a forward-backward translation. Neither temporal expression detection has been found useful in this task, at least with the model proposed. We have found also that the models trained exhibits high variance. That means that we are overfitting the model on training data, so despite the use of the dropout technique, early stopping, fewer parameters or more training data could help to produce a more robust model. Attention mechanism in the neural network could also help (Wang et al., 2016), along with a pre-training of the LSTM with a large corpus of tweets for a language model (predicting next word) and then transfer those weights and retrain them for this specific task.

Finally, we intend to incorporate a module that takes into account the reputation of the user who makes comments, based on non-textual parameters, such as the relationship between the user of the original tweet and the user of the reply tweet, number of followers, knowledge of the subject, etc. We will use that information to work in the second task, predicting the veracity of the original tweet.

Acknowledgements

This research work is partially supported by a grant from the Ministerio de Educación Cultura y Deporte (MECD - scholarship FPU014/00983), the project REDES (TIN2015-65136-C2-1-R) and a grant from the Fondo Europeo de Desarrollo Regional (FEDER).

References

Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–236.

Yi-Chin Chen, Zhao-Yang Liu, and Hung-Yu Kao. 2017. IKM at SemEval-2017 Task 8: Convolutional Neural Networks for stance detection and rumor verification. In *Proceedings of SemEval-2017*, pages 465–469, Vancouver, Canada. ACL.

Marianela García Lozano, Hanna Lilja, Edward Tjörnhammar, and Maja Karasalo. 2017. Mama Edha at SemEval-2017 Task 8: Stance Classification with CNN and Rules. In *Proceedings SemEval-2017*, pages 481–485, Vancouver, Canada. ACL.

Genevieve Gorrell, Kalina Bontcheva, Leon Derczynski, Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2019. SemEval-2019 Task 7: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of SemEval*. ACL.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE.

Elena Kochkina, Maria Liakata, and Isabelle Augenstein. 2017. Turing at SemEval-2017 Task 8: Sequential Approach to Rumour Stance Classification with

Branch-LSTM. In *Proceedings of SemEval-2017*, pages 475—480, Vancouver, Canada. ACL.

Hector Llorens, Estela Saquete, and Borja Navarro-Colorado. 2012. Automatic System for Identifying and Categorizing Temporal Relations in Natural Language. *International Journal of Intelligent Systems*, 27(7):680–703.

Hector Llorens, Estela Saquete, and Borja Navarro-Colorado. 2013. Applying Semantic Knowledge to the Automatic Processing of Temporal Expressions and Events in Natural Language. *Information Processing & Management*, 49(1):179–197.

James Pustejovsky, José M. Castaño, Robert Ingria, Roser Saurí, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev. 2003. Timeml: Robust specification of event and temporal expressions in text. In *New Directions in Question Answering, Papers from 2003 AAAI Spring Symposium, Stanford University, Stanford, CA, USA*, pages 28–34.

Victoria L. Rubin, Yimin Chen, and Niall J. Conroy. 2015. Deception detection for news: Three types of fakes. In *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*, ASIST '15, pages 83:1–83:4, Silver Springs, MD, USA. American Society for Information Science.

Roser Saurí, Jessica Littman, Robert Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. *TimeML Annotation Guidelines 1.2.1 (http://www.timeml.org/)*.

Vikram Singh, Sunny Narayan, Md Shad Akhtar, Asif Ekbal, and Pushpak Bhattacharyya. 2017. Iitp at semeval-2017 task 8 : A supervised approach for rumour evaluation. In *Proceedings of SemEval-2017*, pages 497–501, Vancouver, Canada. ACL.

Feixiang Wang, Man Lan, and Yuanbin Wu. 2017. ECNU at SemEval-2017 Task 8: Rumour Evaluation Using Effective Features and Supervised Ensemble Models. In *Proceedings of SemEval-2017*, pages 491–496, Vancouver, Canada. ACL.

Yequan Wang, Minlie Huang, Li Zhao, et al. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

Heiga Ze, Andrew Senior, and Mike Schuster. 2013. Statistical parametric speech synthesis using deep neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7962–7966. IEEE.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.