Aspie96 at FACT (IberLEF 2019): Factuality Classification in Spanish Texts with Character-Level Convolutional RNN and Tokenization

Valentino Giudice[0000−0002−8408−8243]

University of Turin, Italy
valentino.giudice@edu.unito.it

Abstract. Being able to determine the factual status of events described in a text is crucial to analyze them. This report describes the system used by the Aspie96 team in the FACT shared task (part of IberLEF 2019) for factuality analysis and classification in texts in Spanish.

Keywords: fact · factuality · neural network · natural language processing · Spanish · FACT

1 Introduction

The factuality of an event, as described in [5], expresses its factual status: it conveys whether it is characterized as corresponding to a fact, a possibility, or a situation that doesn't actually hold.

FACT (Factuality Analysis and Classification Task), a shared task organized within IberLEF 2019 (Iberian Languages Evaluation Forum), aimed at the creation of systems able to automatically label events in a text according to their factuality.

In each text, words representing events were marked and given one of three labels:

F (Fact) Situations presented as real by the author.
CF (Counterfact) Situations presented as non real by the author.
U (Undefined) Situations presented as uncertain by the author as they had not yet happened or the author was unaware of their truth value.

Thus, facts were not verified in accordance to the real word, just assessed accordingly to how they had been presented by the author.

In the training datasets, for each text, words representing events were highlighted and classified according to their factuality.

In the testing dataset, the factuality labels of events were not provided, but words representing events were already highlighted: thus, the task was only to label them correctly and identifying them was not necessary.

The results of the task were measured using the macro-average F1-score.

The competition was run using the CodaLab platform [1]. Each team was allowed a total maximum of 10 submissions. Each team could decide, at any moment, which one submission to include in the leaderboard, which was always visible to all participants and updated in real time.

The Aspie96 team took part in the task, using a neural network based on character-level features, adapted to classify words within the text.

The structure of the model and its results are described in the following sections.

2 Description of the System

The system used by the Aspie96 team is a neural network that strictly uses only the data provided for the task, without any additional information (such as pretrained word embeddings).

It is based on the system presented in [3] and on its adaptation presented in [4].

The system presented in [3] had, as a purpose, the (binary) classification of tweets, making use of a character-level representation of them. It was the system presented by the Aspie96 team at the IronITA 2018 task, described in [2], for irony detection in tweets in Italian.

The system presented in [4] by the Aspie96 team at the HAHA task described in [1] for humor detection in tweets in Spanish slightly modifies it, mainly to adapt it to the language. It constitutes the basis of the system used in the FACT task, thus it is crucial to understand it first.

The tweet classification system is represented in Figure 1. It begins with a series of unidimensional convolutional layers followed by a bidirectional recurrent layer. The output of the bidirectional layer, which is an individual dense vector representing information about the whole tweet, is the input of a simple fully connected layer, with one output, whose activation function is the logistic function.

The input is represented as a list with fixed length (leading to padding or truncation, where needed) of sparse vectors. Each vector of the list represents an individual character of the tweet and contains flags whose values are either 0 or 1. A more in-depth description of the input representation can be found in [4].

By removing the last layer from the tweet classifying neural network shown in [3] and [4], the resulting neural network would return, for any text given as input, a fixed-length vector representation. For the sake of simplicity, from this point, the neural network obtained in this way will be referred to as `networkA` and is the backbone of the system used in the FACT task.

[1] https://competitions.codalab.org/

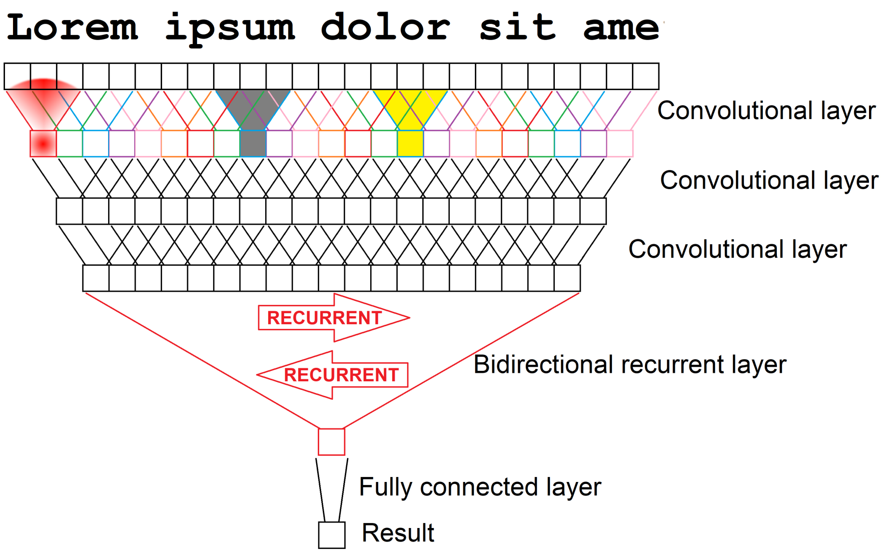

Fig. 1. Visualization of the tweet classification system used in IronITA and HAHA. Each box represents a vector.

The FACT task was quite different from tweets classification: the given texts were much longer and they were not to be classified as a whole: instead, individual words within the text were to be classified.

This required the creation of a neural network capable of processing the individual words within the text as separate tokens. Because the whole premise of the presented work is using strictly the data provided for the task, word embeddings are not a solution.

`networkA`, given a character-level representation of a text, as input, outputs an individual vector representing the given text. It constitutes the basis for the system used by the Aspie96 team in the FACT task.

In each text, all words are detected: they are considered to be tokens. Each word is represented as a fixed-size list of vectors, each of which representing an individual character. The word being represented is centered in its representation and, because of the length of the representation of each word being fixed, the neighbouring characters, on the left and on the right, whether belonging to other words or not (such as in the case of spaces or other special characters) are used as padding. In the vector representation of each character one more flag is added, the *token flag*: its value is 1 if the character belongs to the word being represented, specifically, and 0 otherwise.

As an example, let us consider the following sentence:

```
Lorem ipsum dolor sit amet, consectetur
adipisci elit, sed do eiusmod tempor
incidunt ut labore et dolore magna
aliqua.
```

And let's assume the length of the representation of each word to be 14.
The 5th word (`amet`) has a lenght of 4 and will therefore need $14-4=10$ characters of padding: $10/2=5$ on the left and $10/2=5$ on the right.
It will, therefore, be represented as:

 sit amet, co

The underlining means that the token flag for the marked characters has value 1.
The whole text has 19 words and will thus be represented as 19 lists of 14 vectors: each list representing a word (and its neighbouring characters) and each vector representing an individual character.

Because the representations of each token already encodes the neighboring characters as well there is no need to consider anything other than a word as a token.

This representation is still a character based representation, but the described tokenization allows a neural network using this representation to recognize individual words within the text and process individual words separately.

The neural network used for the FACT tasks uses `networkA` to convert the representation of each word (which is a sparse matrix of fixed size) into an individual fixed-size vector (`networkA` convolves trough the representation of the text, considering each word one by one).

Thus, the representation of each individual word (which includes the neighbouring characters) is fed trough `networkA` separately, obtaining a vector representation of each word: this produces a representation of the text in which each word is encoded into an individual vector. Then, to each of such word-representing vectors one entry is added: the *event flag*, indicating whether the word is an event or not (such information is included in both the training dataset and the testing dataset for every word).

The representation obtained in this way is the input to the following layer of the neural network: a recurrent layer. All outputs of the recurrent layer, each of which being a vector, are considered (as many as the words in the text), not just the last one. A dense layer is then applied to get, for each word, its classification in one of the three classes. The classification is ignored for words that do not represent events.

The full network presented by the Aspie96 team in the FACT task is shown in Figure 2. The purpose of the recurrent layer is to read the text, in a human-like fashion, encoding, in each word, its meaning, according to the previous ones. Thanks to the usage of the neighbouring characters of each word as padding, there is no need to use additional data to represent punctuation. Also, the same word will be represented in different ways depending on its surroundings: this ensures a more unambiguous representation of the meaning of each word (a word may have different meanings depending on its surroundings, also it is easier to infer the meaning of a word if its context is provided, given the fact that the meaning of each word must be inferred for character-level features only), also including information about the following words (the padding is big enough to allow this. Note that humans don't usually need to read much ahead to fully understand the meaning of a word). It must be noted that the structure of `networkA` has been slightly adapted from task to task and is not identical to the neural network used in [4].

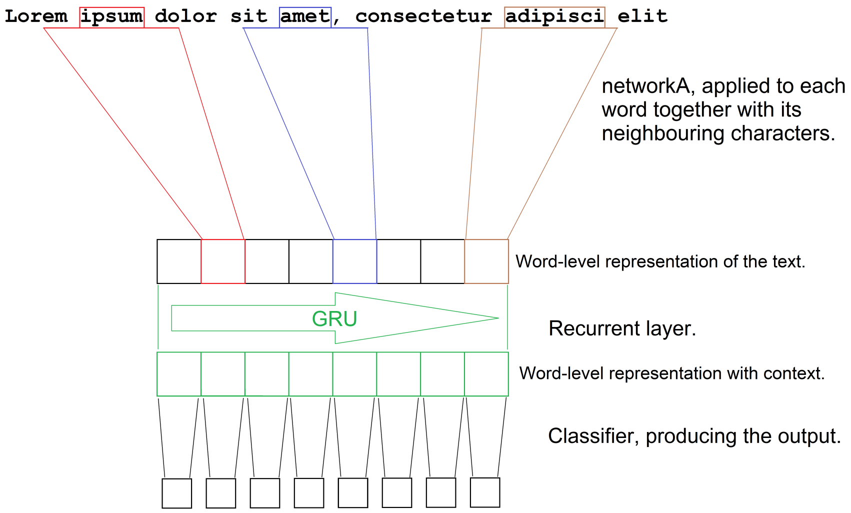
Lorem ipsum dolor sit amet, consectetur adipisci elit
networkA, applied to each word together with its neighbouring characters.
Word-level representation of the text.
GRU
Recurrent layer.
Word-level representation with context.
Classifier, producing the output.

Fig. 2. Visualization of the event classification system used in FACT. Each box represents a vector. The addition of the event flag is ignored.

3 Results

A total of 5 teams took part in the FACT task.

The Aspie96 team ranked 2nd, with a macro-average F1-score of 0.554, just below team premjithb, with a macro-average F1-score of 0.561. The 3rd ranking team was jimblair, with a macro-average F1-score of 0.489. The accuracy of the system presented by the Aspie96 team was 0.635.

A baseline system was provided by the task organizers. The baseline system (aiala) assigned labels randomly with a 0.7 probability for the **F** class, a 0.1 probability for the **CF** class and a 0.2 probability for the **U** class.

The results of all teams are showed in Table 1.

Table 1. Results of all teams at the FACT shared task. The score is the macro-averaged F1-score.

Team	Score	Accuracy
premjithb	0.561	0.721
Aspie96	0.554	0.635
jimblair	0.489	0.622
macro128	0.362	0.579
aiala (baseline)	0.340	0.524
garain	0.301	0.512

4 Discussion

This paper presented the system used by the Aspie96 team in the FACT task for factuality classification of events within a text. The system produced good results, with a macro-average F1-score very close to that of the first ranking team.

The presented system, based upon adaptations of the one originally presented in [3], is a character-level convolutional recurrent neural network which makes no use of pretrained features (such as word embeddings), nor of additional knowledge or intuition about the task, but takes advantage of tokenization to classify individual words within the text.

The features of the neural network are meant to make it as general as possible: it should be possible to use the system to, in general, classify words within a text, regardless of the specific high-level task (thus, regardless of it being factuality analysis or not).

This result has not been reached yet: despite the system proving itself to be able to achieve good results in factual analysis and classification, much work is still ahead to make it more general, as much worse results have been obtained for other tasks.

As for the structure of `networkA`, it had been slightly tweaked between different tasks (not always out of necessity, but resulting in several slightly different versions).

Thus, further research is needed to create an individual, more stable, tweet-classifying neural network, usable for different tasks (by changing only the number of outputs according to the number of classes) and, based on that, an individual system for classification of individual words within a text, like in FACT.

Time will reveal where the limits of such an approach lay. The results obtained in FACT, considering the structure of the network having nothing to do with event classification specifically, are quite promising in this direction: more work is needed to allow convergence towards an individual system.

References

1. Chiruzzo, L., Castro, S., Etcheverry, M., Garat, D., Prada, J.J., Rosá, A.: Overview of HAHA at IberLEF 2019: Humor Analysis based on Human Annotation. In: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019). CEUR Workshop Proceedings, CEUR-WS, Bilbao, Spain (9 2019)
2. Cignarella, A.T., Frenda, S., Basile, V., Bosco, C., Patti, V., Rosso, P., et al.: Overview of the evalita 2018 task on irony detection in italian tweets (ironita). In: Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18). pp. 26–34. CEUR Workshop Proceedings, CEUR-WS (2018), http://ceur-ws.org/Vol-2263/paper005.pdf
3. Giudice, V.: Aspie96 at IronITA (EVALITA 2018): Irony Detection in Italian Tweets with Character-Level Convolutional RNN. In: Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18). pp. 160–165. CEUR Workshop Proceedings, CEUR-WS (2018), http://ceur-ws.org/Vol-2263/paper026.pdf
4. Giudice, V.: Aspie96 at HAHA (IberLEF 2019): Humor Detection in Spanish Tweets with Character-Level Convolutional RNN. In: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019). CEUR Workshop Proceedings, CEUR-WS, Bilbao, Spain (2019)
5. Saur, R.: A Factuality Profiler for Eventualities in Text. Ph.D. thesis, Brandeis University (2008), https://www.cs.brandeis.edu/~roser/pubs/sauriDiss_1.5.pdf