Aspie96 at HAHA (IberLEF 2019): Humor Detection in Spanish Tweets with Character-Level Convolutional RNN

Valentino Giudice[0000−0002−8408−8243]

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

Abstract. A characterization of humor based upon machine learning that allows its automatic detection is not yet specified. This report describes the system used by the Aspie96 team in the HAHA shared task (part of IberLEF 2019) for humor recognition in tweets in Spanish: a neural network using exclusively character-level features.

Keywords: humor · neural network · natural language processing · Twitter · Spanish · HAHA

1 Introduction

HAHA (Humor Analysis based on Human Annotation), a shared task organized within IberLEF 2019 (Iberian Languages Evaluation Forum) and described in [2], proposed two different subtasks related to automatic humor detection in tweets in Spanish:

Humor Detection Telling if a tweet was intended to be humorous by the author or not. The results of the subtask were measured using F1-score for the positive (humorous) class.

Funniness Score Prediction Predicting the level of funniness of humorous tweets. The results of the subtask were measured using RMSE (root-mean-squared error).

The tweets had been crowd-annotated in [1] according to a voting scheme: annotators could, for each tweet, mark it as not humorous or mark it as humorous and give it a number of stars (1 to 5) according to its funniness (5 being the best), for a total of six options. Combining the votes of the annotators, each tweet was labelled as humorous or not humorous and each humorous tweet was given a funniness score, from 1 to 5, equal to its average number of star.

In the training dataset, for each tweet, the following information was provided:

Tweet ID The ID of the tweet, on Twitter, not intended to be used to extract metadata.
Text The text of the tweet.
Is humorous Whether the tweet is humorous.
Votes (Not humor, 1 star, 2 stars, 3 stars, 4 stars, 5 stars) The number of votes given to the tweet, for each possibility.
Funniness score The average funniness score of the tweet, only provided for humorous tweets.

In the testing dataset, just the text of each tweet was provided.

The competition was run using the CodaLab platform[1]. Each team was allowed up to 10 submissions per day and a total maximum of 20 submissions. Each team could decide, at any moment, which one submission to include in the leaderboards, which were always visible to all participants and updated in real time. A separate leaderboard was used for each subtask.

Each submission could be meant for the humor detection subtask only or for both subtasks and annotated each tweet in the testing dataset with a binary label, indicating whether it had been detected as humorous and, if it was meant for the funniness score prediction dataset also, the predicted funniness score (even for tweets not detected as humorous). The predicted funniness score was only considered for tweets classified as humorous in the gold testing dataset.

The Aspie96 team took part in both subtasks, using a neural network with character-level features.

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 strictly works at character level, without using any word-level features (such as word embeddings) or any data other than what is provided for the specific task at hand.

It is a neural network adapted from [5], where it was used for the IronITA 2018 task of irony detection in tweets in Italian described in [3].

The input of the system is a fixed-size list of arrays. The neural network begins with a series of unidimensional convolutional layers: each filter has a small width (of 3) and convolves trough the input. The output of each convolutional layer is again a list of arrays: the length of each array is constant for each layer and is set trough a hyperparameter (it is 8 for all convolutional layers), while the length of the list is slightly smaller than that of the input one, because the convolutional layers don't use padding. The series of convolutional layers is 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

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

activation function is the logistic function (the logistic function has an output between 0 and 1).

The purpose of the unidirectional convolutional layers is to create a higher-level dense representation of each input vector, together with its surrounding ones, providing context. The purpose of the bidirectional recurrent layer is to produce an individual vector which can be considered as a representation of the tweet. Additional layers meant for regularization are used (a gaussian noise layer applied to the input of the neural network and several dropout layers).

The input tweet is represented as a list with fixed length (leading to padding on the left or truncation to the right, 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.

Most of the flags are mutually exclusive and are used to identify a character among a list of known ones. Additional flags are used to represent properties of the character.

The full list of known characters is the following:

```
Space ! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7
8 9 : ; = ? @ [ ] _ a b c d e f g h i j k l m n o p q
r s t u v w x y z | ~
```

Emojis are represented similarly to their Unicode name (in English), with additional flags.

The full list of additional flags is:

Uppercase letter Indicates whether the character is an uppercase letter.
Accent Indicates whether the character is an accented vowel, regardless of the accent being acute or grave.
Emoji Indicates whether the character is part of the Unicode name of an emoji.
Emoji start Indicates whether the character is the first in the Unicode name of an emoji.
Letter Indicates whether the character is a letter.
Number Indicates whether the character is a numerical digit.
Inverted Indicates if the character is an inverted question mark or exclamation mark (¿ or ¡).
Tilde Whether the character is an N with virgulilla (ñ or Ñ).

Multiple spaces are represented as one and unknown characters are ignored.

A more in-depth description of the role of each layer is given in [5]. A visualization of the network is given in Figure 1.

The main difference between the model used for the HAHA task and the one presented in [5] for IronITA is the language: Spanish contains some characters which are not part of the Italian language.

The same structure was used for both HAHA subtasks: for the humor detection subtask, the output of the network was rounded to 0 or 1 to get a binary value and for the funniness score prediction subtask it was multiplied by five to

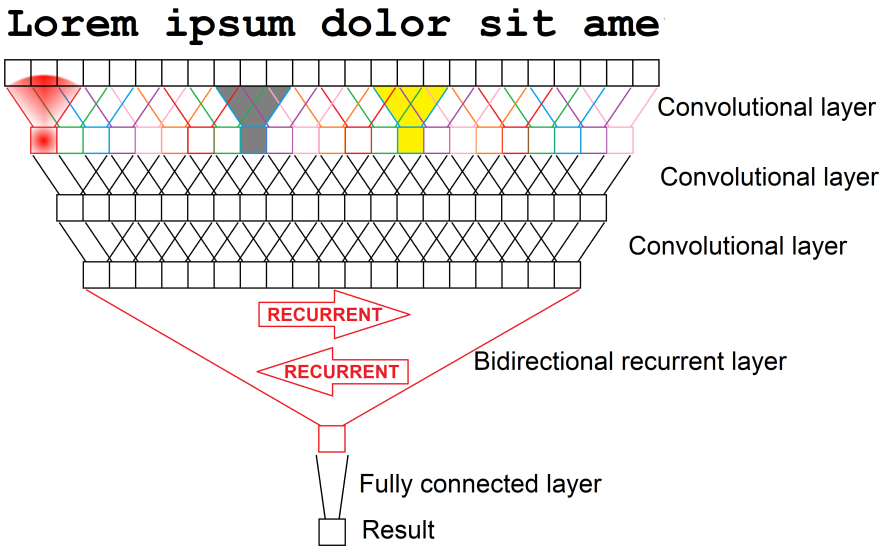

Fig. 1. Visualization of the proposed model. Each box represents a vector.

get a value between 0 and 5 (in principle this could have allowed the model to output funniness scores below 1, but that was not the case for any of the labels in the submission).

3 Results

A total of 18 teams took part in the humor detection subtask.

In the humor detection subtask, the Aspie96 team got an F1-score, for the positive class, of 0.711, ranking in the 13th position, below LadyHeidy, with an F1-score of 0.725 and above dodinh, with an F1-score of 0.660.

As a comparison, the best ranking team (adilism) got an F1-score of 0.821.

The values for precision, recall and accuracy for the Aspie96 team were, respectively, 0.678 (14th position), 0.749 (11th position) and 0.763 (13th position).

The baseline system was one marking tweets as humorous with a probability of 0.5. It got an F1-score, precision, recall and accuracy of, respectively, 0.440, 0.394, 0.497 and 0.505.

The results of the Aspie96 team are summarized in Table 1 and compared with the best system and the baseline system.

A total of 13 out of 18 teams took part in the funniness score prediction subtask.

In the funniness score prediction subtask, the Aspie96 team got a RMSE of 1.673, ranking in the 12th position, below garain, with a RMSE of 1.653 (a lower RMSE is better) and above dodinh, with a RMSE of 1.810.

Table 1. Results of the Aspie96 team at the humor detection subtask, compared with the best system and the baseline system.

System	F1-score	Precision	Reacall	Accuracy
adilism (best system)	0.821	0.791	0.852	0.855
Aspie96 system	0.711	0.678	0.749	0.763
baseline	0.440	0.394	0.497	0.505

As a comparison, the best ranking team (adilism) got a RMSE of 0.736.

The baseline system was one giving every tweet a funniness score of 3. It got an RMSE of 2.455.

The results of the Aspie96 team are summarized and compared to the baseline and to the best system in Table 2.

Table 2. Results of the Aspie96 team in the funniness score prediction subtask, compared with the results of the best system and the baseline system.

System	RMSE
adilism (best system)	0.736
Aspie96 system	1.673
baseline	2.455

4 Related work

Similar models had been presented before.

A roughly similar model based on convolutions for text classification was presented in [6] in the context of EMNLP 2014, described in [7]. It used word-level features (trough a pretrained and then fine-tuned embedding layer) as the input of an individual convolutional layer, which used multiple kernels of different sizes, to detect different high-level features. A timewise max-pooling layer was then used to produce a vector whose length was the same as the number of kernels in the convolutional layer. The resulting vector was the input of a fully connected layer producing the output of the neural network. The model produced results better of those of the state of the art at the time on 4 out of 7 tasks.

In [8], a model more similar to the one proposed was presented. The model used a character-level convolutional neural network for text classification, achieving competitive results. However, it did not use a recurrent layer and represented each input character as a one-hot encoded vector. The model was trained using very big datasets as this can work better for character-level neural networks, which don't rely on pretrained word embeddings, making use of information outside the training set. Because of its structure and attributes, the model wasn't much flexible and easily adaptable to different kinds of usage.

5 Discussion

The results, for both HAHA subtasks, were significantly better than the baseline, however there is clearly much room for improvement.

As the system is a mere adaptation of the one presented in [5], this allows to compare its performance among the two different tasks.

In the binary classification subtask presented in [3], the model of the described in [5] had a performance which was not too much worse from the one of the teams ranking above, the best of which described in [4].

However, in the HAHA binary classification subtask (humor detection) the results in the leaderboard were much more spread out and the results obtained by the Aspie96 team were quite different from those of the best systems. Despite the scores of the best ranking teams being better than those shown in [3], the results obtained by the Aspie96 team were not.

The structure of the neural network used in the HAHA task by the Aspie96 team, although having been originally presented in [5] for the classification of Italian tweets, was not built specifically for the Italian language, but for both Italian and English and its structure suggests its applicability for different tasks of tweets classification (as long as the language is an alphabetical one and uses the Latin alphabet). The results obtained in HAHA, however, show the neural network to be less flexible than anticipated and the nature of the specific task at hand to influence how close its results can be to the best ones achieved.

This does not mean a character-level approach cannot be used effectively for humor detection, but the structure of the neural network must be improved in order to be more general, resulting in better results and consistency among different tasks.

For the simple case of binary tweet classification, the ideal result would be that of obtaining a structure capable of working across different languages (at least Spanish, English and Italian), regardless of the high-level task (whether humor detection, irony detection or any other).

Results outside the scope of this paper, regarding the performance of the model, in different tasks, in different languages, also confirm the current inconsistency of the results obtained by the structure, but show its applicability and its ability to get good results for at least some different tasks, suggesting it is indeed needed to make it more general.

References

1. Castro, S., Chiruzzo, L., Rosá, A., Garat, D., Moncecchi, G.: A Crowd-Annotated Spanish Corpus for Humor Analysis. In: Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media. pp. 7–11 (2018), https://aclweb.org/anthology/papers/W/W18/W18-3502/
2. 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)

3. 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
4. Cimino, A., De Mattei, L., Dell'Orletta, F.: Multi-task learning in deep neural networks at evalita 2018. In: Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18). pp. 86–95. CEUR Workshop Proceedings, CEUR-WS (2018), http://ceur-ws.org/Vol-2263/paper013.pdf
5. 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 (2018), http://ceur-ws.org/Vol-2263/paper026.pdf
6. Kim, Y.: Convolutional Neural Networks for Sentence Classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1746–1751. Association for Computational Linguistics, Doha, Qatar (2014), https://doi.org/10.3115/v1/D14-1181
7. Moschitti, A., Pang, B., Daelemans, W. (eds.): Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar (2014), https://doi.org/10.3115/v1/D14-1
8. Zhang, X., Jake Zhao, J., LeCun, Y.: Character-level Convolutional Networks for Text Classification. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28. pp. 649–657. Curran Associates, Inc. (2015), http://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classification