UTMN at HAHA@IberLEF2019: Recognizing Humor in Spanish Tweets using Hard Parameter Sharing for Neural Networks

Anna Glazkova[1] [0000−0001−8409−6457], Nadezhda Ganzherli[1], and Elena Mikhalkova[1] [0000−0003−0781−8633]

University of Tyumen, Tyumen, Russia
{a.v.glazkova,n.v.ganzherli,e.v.mikhalkova}@utmn.ru

Abstract. Automatic humor detection is a hard but challenging task. For the competition HAHA at IberLEF 2019 we built a neural networks classifier that uses different types of neural networks for specific sets of features. After being trained separately, the layers are concatenated to give the general output. The performance of our system on the binary detection of humorous tweets reaches F-score of 0.76 which is comparably higher than results of baseline machine learning classifiers and earns us the ninth place in the ranking table. As for task 2, where the system has to guess how funny the tweet was based on the number of stars that it got, our result is similarly good: $RMSE = 0.945$. However, much needs to be done to evaluate contribution of each of the feature sets and our choice of the type of neural network.

Keywords: Humor detection · Neural networks · Hard parameter sharing · Feature engineering.

1 Introduction

Humor detection is a non-trivial task that was considered by (16) to be of the *AI-complete* kind, as humor is “one of the most sophisticated forms of human intelligence”. However, with the rise of neural networks and semantic vector algorithms, e.g. the one suggested by (7), it has recently gained much attention of researchers and organizers of competitions: SemEval by (11; 8) and HAHA at IberLEF by (2; 4). A lot of systems at these competitions, including winners, apply machine learning and semantic vectors:

1. INGEOTEC by (10) uses a combination of machine learning methods and word embeddings.
2. UO UPV by (9) is based on a Bidirectional LSTM neural network and word embeddings.

3. JU-CSE-NLP by (12) "is a rule-based implementation of a dependency network and a hidden Markov model".
4. Idiom Savant by (5) "consists of two probabilistic models... using Google n-grams and Word2Vec".
5. PunFields by (6) applies a linear SVM classifier to a manually built thesaurus of English words.

Our approach at HAHA@IberLEF2019 is not an exception from this trend.

2 Dataset and Preprocessing

"HUMOR: A Crowd-Annotated Spanish Corpus for Humor Analysis" by (3) was created in 2017. At HAHA@IberLEF2019 the training set consisted of 16,000 tweets, manually annotated as *humorous* and *not humorous* and with the score of funniness calculated based on the average number of "stars" (5 maximum) given to a tweet by several independent readers. 1,200 tweets from the Training dataset were used for validation. The test set included 4,000 tweets. In addition, we used several datasets like pre-trained word embeddings and a sentiment dictionary. They are described in the next section.

We first preprocess tweets with the help of our own software that includes the following steps:

1. Convert some markers of emotions into words: *:(* to *tristeza* and *:) :D xD XD* to *reir*.
2. Convert repetitive sequences of *jaja...* and *JAJA...* to a simple *ja*.
3. Pre-tokenize: add a space before and after punctuation symbols except # and .
4. Convert repetitions of the same letter of length ≥ 3 into a single letter and add a lemma *EMPHASIS* to the tweet so that it lexically denotes sentiment implied by letter repetitions: *sooooomos* to *somos EMPHASIS*.
5. Tokenize hashtags # and mentions :
   (a) De-capitalize capitalized sequences of more than one letter leaving the first letter capitalized: *NUET* to *Nuet*.
   (b) Add a space before and after every non-letter character in the sequence: *PP#CiU* to *PP # CiU*.
   (c) Add a space before every capitalized character in the sequence: *DaviniaBono* to *Davinia Bono*.
6. Convert emoticons into their word representations using unicode tables in Spanish[1].

Our Python script for tweet preprocessing (except emoticons) and the system we used at the Competition will soon be available at `https://github.com/evrog/Spanish_Humor`. As concerns the choice of steps, it is basically a tradeoff between not ruining words with traditional orthography and extracting as many

[1] We used tables from `https://unicode-table.com/es`.

lemmas from Internet-specific speech as possible. The final stage of preprocessing is lemmatization with the help of SpaCy[2]: the output is a list of lemmas and special characters in a tweet.

3 System Architecture

As mentioned above, our model is based on a neural network. It processes several types of features in parallel, then concatenates the layers' outputs and passes them to the *Dense* layer. (14) calls this approach "hard parameter sharing" and binds it to the work on Multitask Learning by (1).

We used a subset of 1,200 tweets from the Training dataset for validation and accuracy, as a validation measure. To avoid overfitting, we used the early stopping strategy (the value of patience is 5) and the dropout regularization for the output layer (the fraction of the input units to drop is 0.8).

The model learns on four sets of features in parallel:

1. **Tweets represented as sequences.** The length of word emebeddings is 300. The weight matrix is built from pretrained word embeddings for Spanish[3]. In our experiment, the *Convolutional neural network* learned better from these features than the *Recurrent* one, so these features are fed to CNN and a further combination of *MaxPooling* and a flattening layer. The Convolutional layer contains 64 filters with a kernel size of 5. The size of the max pooling windows is 20.
2. **Tweets represented as a Bag-of-Words and smoothed with TF-IDF.** These features are restricted to the 5,000 most frequent words, due to computation complexity of a larger vocabulary, and passed to a *Dense* layer.
3. **Features of sentiment and topic modelling, extracted from tweets.** To represent sentiment in every tweet, we used "affective norms" calculated by (15).[4] For each word in a tweet we collected its six norms from the dictionary, getting a word vector. We summed these vectors to create a vector of a sentence and applied MinMax normalization to scale its values between 0 and 1:
$$y_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{1}$$
As concerns topics, we used LDA from Gensim (13) to extract 20 most general topics from the collection (topic distribution) and calculate distance from each tweet to each topic. The features are passed to a *Dense* layer.
4. **Additional features.** These features include:
   (a) presence (0) or absence (1) of emoji, lists, word repetitions, special characters (e.g. *!, ?*);

[2] https://spacy.io/api/lemmatizer
[3] https://www.kaggle.com/rtatman/pretrained-word-vectors-for-spanish/
[4] The dictionary of norms can be downloaded here: http://crr.ugent.be/archives/1844.

(b) normalized with MinMax quantitative features: number of words, lines, minimum and maximum distance between embeddings, minimum Levenstein distance between a pair of words (to detect puns) applied to all possible pairs of words in a tweet.

The features are also passed to a *Dense* layer.

Scheme 1 demonstrates the general outline of our best performing neural network used for the task of binary classification. The scheme includes main parameters, e.g. the size of windows is 20 in *20: MaxPooling*. The optimizer is *adam* (adaptive moment estimation); the loss function is *binary crossentropy*; the activation functions at hidden layers are $ReLU$, and for the output layer it is *softmax*. The last layer includes *Dropout* regularization with probability $\epsilon = 0.8$. For the second task the architecture is similar to that of the first task, but for the input values that are separated into classes according to the average number of stars that the tweets earned. Also, in the second task, we use the *mean standard error* for validation.

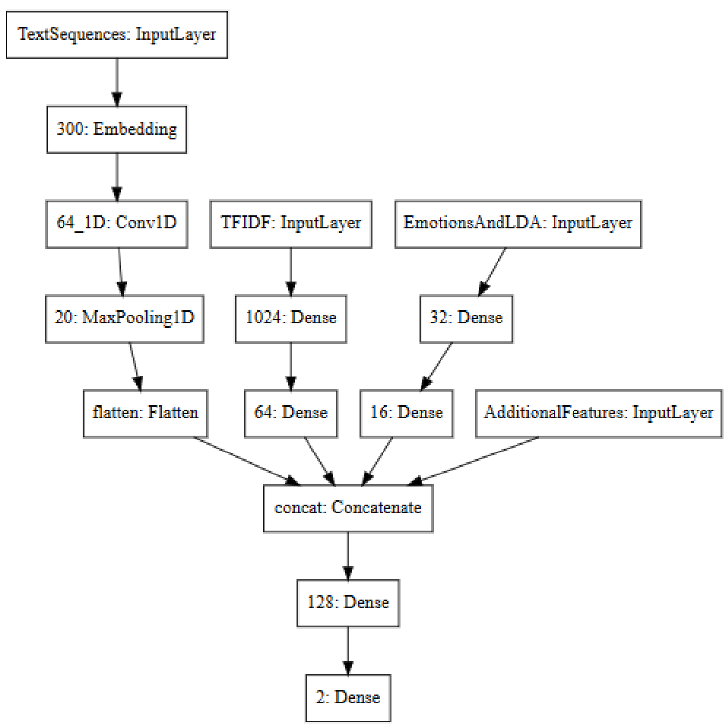

Fig. 1. Architecture of the best-performing neural network.

4 Test Results

Table 1 demonstrates the result of our system compared to the winners of two tasks. The first four measures are for Task 1, and the last column presents Task 2. As concerns Task 1, the performance of our system is average compared to other teams' results. However, it is high above the chance value and comparably higher than results of baseline machine learning classifiers.

Table 1. Test results at the competition. Baseline result was suggested by the competition organizers and was marked in the scoring table as "hahaPLN".

System	F-score	Precision	Recall	Accuracy	Task 2: RMSE
Our	0.760 (9)	0.756 (9)	0.765 (8)	0.812 (9)	0.945 (8)
Winner	0.821 (1)	0.791 (4)	0.852 (1)	0.855 (1)	0.736 (1)
Baseline	0.440 (19)	0.394 (19)	0.497 (18)	0.505 (18)	2.455 (14)

5 Conclusion

Application of computer methods, in particular word embeddings and neural networks, in analysis of figurative speech and its varieties, such as humor, has recently proved to be very effective in annotation of large corpora. But also, it gives a new perspective on the analysis of language features that are important in humor production and appreciation. Our approach included testing sets of different features in a growing combination: at each step we added a feature set and a subset of a neural network into the architecture and checked if they improved our result. For example, including the sentiment dictionary (see above: "affective norms") improved our F-score by 0.015. The features we chose are usual in the systems we mentioned in 1: vocabulary represented as word embeddings, TF-IDF weighted Bag-of-Words, sentiment dictionary, special characters that represent emotions in Twitter (e.g. *:)*).

As for the system architecture, we tested the so-called hard parameter sharing. Our system uses different neural networks that we empirically found to be more capable of dealing with each specific set of features. In general, we use CNN for embeddings and Dense layers for other feature sets. The result of our system is much higher than that of the baseline and is medium compared to other participants. However, the value of each of the sets features and the model of neural network have yet to be evaluated more closely. We plan to combine our features with other types of neural networks. Also, the model might have given a higher result in Task 2 if we used a regression model instead of multi-class classification. However, this is yet to be tested.

Acknowledgements

The reported study was funded by RFBR according to the research project No. 18-37-00272.

Bibliography

[1] Caruana, R.A.: Multitask learning: A knowledge-based source of inductive bias. In: Machine Learning Proceedings 1993, pp. 41 – 48. Morgan Kaufmann, San Francisco (CA) (1993). https://doi.org/https://doi.org/10.1016/B978-1-55860-307-3.50012-5, `http://www.sciencedirect.com/science/article/pii/B9781558603073500125`

[2] Castro, S., Chiruzzo, L., Rosá, A.: Overview of the haha task: Humor analysis based on human annotation at ibereval 2018. In: CEUR Workshop Proceedings. vol. 2150, pp. 187–194 (2018)

[3] 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)

[4] 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)

[5] Doogan, S., Ghosh, A., Chen, H., Veale, T.: Idiom savant at semeval-2017 task 7: Detection and interpretation of english puns. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). pp. 103–108 (2017)

[6] Mikhalkova, E., Karyakin, Y.: Punfields at semeval-2017 task 7: Employing rogets thesaurus in automatic pun recognition and interpretation. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). pp. 426–431 (2017)

[7] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)

[8] Miller, T., Hempelmann, C., Gurevych, I.: SemEval-2017 Task 7: Detection and interpretation of English puns. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). pp. 58–68 (2017)

[9] Ortega-Bueno, R., Muniz-Cuza, C.E., Pagola, J.E.M., Rosso, P.: Uo upv: Deep linguistic humor detection in spanish social media. In: Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018) (2018)

[10] Ortiz-Bejar, J., Salgado, V., Graff, M., Moctezuma, D., Miranda-Jiménez, S., Tellez, E.S.: Ingeotec at ibereval 2018 task haha: μtc and evomsa to detect and score humor in texts. In: Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages

(IberEval 2018) co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018) (2018)

[11] Potash, P., Romanov, A., Rumshisky, A.: Semeval-2017 task 6:# hashtag-wars: Learning a sense of humor. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). pp. 49–57 (2017)

[12] Pramanick, A., Das, D.: Ju cse nlp @ semeval 2017 task 7: Employing rules to detect and interpret english puns. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). pp. 432–435 (2017)

[13] Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010), http://is.muni.cz/publication/884893/en

[14] Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 (2017)

[15] Stadthagen-Gonzalez, H., Imbault, C., Sánchez, M.A.P., Brysbaert, M.: Norms of valence and arousal for 14,031 spanish words. Behavior research methods **49**(1), 111–123 (2017)

[16] Stock, O., Strapparava, C.: Hahacronym: Humorous agents for humorous acronyms. Stock, Oliviero, Carlo Strapparava, and Anton Nijholt. Eds pp. 125–135 (2002)