UMUTeam at SemEval-2021 Task 7: Detecting and Rating Humor and Offense with Linguistic Features and Word Embeddings

José Antonio García-Díaz
Facultad de Informática,
Universidad de Murcia,
Campus de Espinardo, 30100
Murcia, Spain
joseantonio.garcia8@um.es

Rafael Valencia-García
Facultad de Informática,
Universidad de Murcia,
Campus de Espinardo, 30100
Murcia, Spain
valencia@um.es

Abstract

In writing, humor is mainly based on figurative language in which words and expressions change their conventional meaning to refer to something without saying it directly. This flip in the meaning of the words prevents Natural Language Processing from revealing the real intention of a communication and, therefore, reduces the effectiveness of tasks such as Sentiment Analysis or Emotion Detection. In this manuscript we describe the participation of the UMUTeam in HaHackathon 2021, whose objective is to detect and rate humorous and controversial content. Our proposal is based on the combination of linguistic features with contextual and non-contextual word embeddings. We participate in all the proposed subtasks achieving our best result in the controversial humor subtask.

1 Introduction

In this manuscript we describe our participation in the shared task *HaHackathon 2021* (Meaney et al., 2021), proposed in the Forum for Information Retrieval Evaluation (IberEval'2021) whose objective is the identification and evaluation of humorous and offensive texts written in English. Humor allows you to present the reality by highlighting the comic and ridiculous side of life. Humor is hard to identify, even for humans (Vrticka et al., 2013). On the one hand, there are many forms of humor: from mild forms, such as jokes or puns, that result in better and safer social environments, to biting forms, like sarcasm, which is used as a rhetorical device. Humor can also have a constructive end, as happens in satire, where irony, double meaning, and sharp analogies are used to ridicule someone or something. On the other hand, the sharper the humor, the more effort it takes to understand it. When humor is misunderstood, or when humor itself has a transgressive purpose, it can lead to controversies and confrontations. Furthermore, an added difficulty is that humor is highly subjective and context dependent, making it even more difficult to understand (Jiang et al., 2019). Nevertheless, the benefits of endowing to a machine basic notions about humor and figurative language understanding outweigh the challenges they pose, because they lead to natural language-based interfaces to feel more naturally, such as chat-bots and virtual assistants (Ritschel et al., 2019) and more reliable results in tasks such as opinion mining.

We participate in all the proposed subtasks of HaHackathon'2021 with two systems that combines linguistic features (LF) extracted with a tool developed by our research group with (1) pretrained word embeddings (PWE) from fastText (Mikolov et al., 2017) and GloVe (Pennington et al., 2014), and (2) contextual word embeddings from BERT (Devlin et al., 2018) (CWE). Our best result was achieved in task 1c (controversial humor), in which we trained a classification neural network that distinguishes between *non-humor*, *humor*, and *offensive* but outputs a binary prediction which indicates whether a text is controversial or not. The code is available at https://github.com/Smolky/hahackathon-2021.

2 Background

The HaHackathon 2021 challenge consists in two binary classification problems of humorous content and controversial tweets and two regression problems to identify how humorous and controversial the annotators considered the texts. We only used the dataset given by the organizers, two pretrained word embeddings models, and the contextual word embeddings from BERT.

The dataset provided consisted in 10k tweets written in English and posted in three subsets, namely training, development, and testing, with a ratio of 80-10-10. All tweets were annotated by

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 1096–1101
Bangkok, Thailand (online), August 5–6, 2021.

US English-speaking annotators of different genders and age groups with the following questions: (1) *Is the intention of this text to be funny?*, (2) *[If so] How humorous do you find the text? (on a scale of 1 to 5)*, (3) *Is this text generally offensive?*, and (4) *[If so] How generally offensive is the text? (on a scale of 1 to 5)*. Based on the data we had during the competition, we observed that the training dataset was imbalanced for subtask 1a, with a predominance of humorous tweets (61.65%) and almost balanced for subtask 2a. For regression subtasks, the average rating was 2.26 (σ of 0.5670) for subtask 1b and 0.58 (σ of 0.98) for subtask 2a.

Most of the literature found on the detection of humor highlights the importance of figurative language in which, contrary to the literal sense of language, words and expressions change their conventional meaning to refer to something without saying it directly (del Pilar Salas-Zárate et al., 2020). The reader can find in that work a compendium of works that analyze sarcasm, irony, and satire in English. Modern approaches rely on novel deep-learning transformers, such as the work described in Javdan et al. (2020), focused on sarcasm from a figurative language perspective, and in which the authors used BERT to build an aspect-based sentiment analysis system to determine whether the response is sarcastic. Other works, such as the one described in del Pilar Salas-Zárate et al. (2017), explores the differences and similarities in how satire is perceived in countries that share the language, but not the cultural background. To do this, they compare the use of linguistic characteristics with a corpus of satirical news written in Spanish from Spain and another written in Spanish from Mexico.

3 System overview

Our proposal is based on the usage of LF, PWE, and CWE to detect humor and controversial content. CWE have outperformed previously state of the art results regarding text classification tasks. However, we state that both PWE and CWE ignore relevant clues that are present in writings. For example, we observed in the dataset provided, the presence of capital letters (commonly used when shutting or to raise the voice), quoted sentences or dialogues that reproduce real or figurative conversations that are not captured with any of the above techniques.

To obtain the LF, we use a subset of the features extracted with UMUTextStats (García-Díaz et al., 2020, 2021). This tool is inspired in LIWC (Tausczik and Pennebaker, 2010) but designed by our research group from scratch for the Spanish language. As the HaHackathon 2021 dataset only deals with English, we only extract statistical features discarding all features that we had that work with Spanish lexicons. Specifically, we select:

- **Expressive lengthening**, drawing out or emphasizing a verbalized word, giving it character.
- **Common typing mistakes**, such as starting sentences in lowercase, numbers, consecutive repetitions of the same word, and incorrect use of punctuation.
- **Corpus statistics**, such as the standard type/token ratio (TTR), the number of words, syllables, sentences, quoted sentences, interrogative and exclamatory sentences, and the average length of the words.
- **Punctuation symbols and emoticons.**
- **Common traits used in social media communication**, such as the presence of hyperlinks, hashtags or jargon.

The next step was to obtain the best deep-learning architecture for each subtask. We evaluate two main approaches. On the one hand, we combine the LF with PWE and different deep-learning architectures. On the other, we combine the LF with CWE from BERT (*bert-base-uncased*) using HuggingFace[1]. After performing these two models, we sent our results to the platform to evaluate them with the development dataset, achieving our best result with BERT for subtask 1a and RNNs for subtasks 1b, 1c, and 2a. After performing these two models, we sent our results to the platform to evaluate them with the development dataset, achieving our best result with BERT for subtask 1a and RNNs for subtasks 1b, 1c, and 2a.

4 Experimental setup

Our experimental setup is depicted in Figure 1 and, in a nutshell, we can described as follows.

First, we perform a preprocessing stage that consist of:

1. Removing social media language, such as hyperlinks or mentions.

[1] https://huggingface.co/ (v3.4.0)

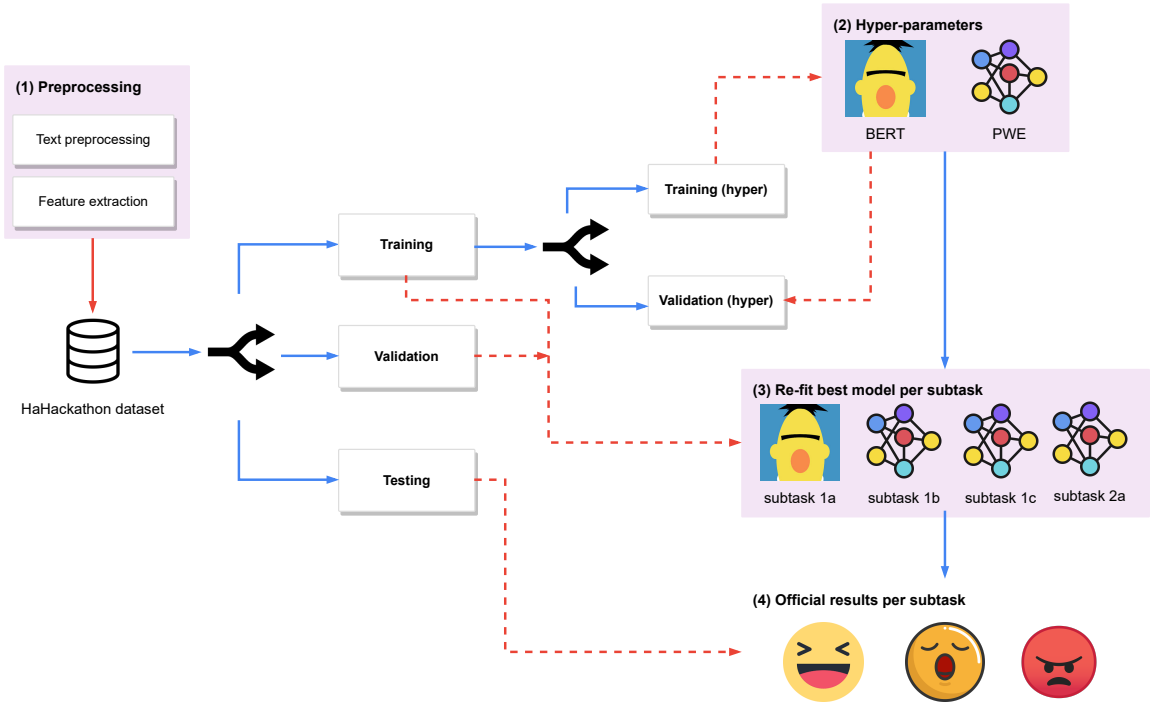

Figure 1: Pipeline of the participation of UMUTeam at HaHackathon'2021.

2. Removing digits.
3. Removing word elongations.
4. Transforming the tweets into their lowercase form.
5. Removing punctuation symbols.

Second, as the organizers of the shared task released the labels of the development dataset in the last stage of the competition, we begun the competition by dividing the 8K tweets of the training data into two folds in a ratio of 80-20. Third, two main approaches were evaluated. On the one hand, we use Talos[2] to evaluate different pretrained word embeddings models (FastText, GloVe, and word2vec) and several neural networks architectures (MLP, CNN, LSTM, BiLSTM, GRU, and BiGRU). On the other, we evaluate contextual word embeddings from BERT. These two processes are described in detail in the next paragraph. Finally, the classification subtasks (1a and 1c), the results are evaluated with the Accuracy and the F1-measure whereas the regression subtasks (1b and 2a) are evaluated with the Root Mean Squared Error (RMSE).

In case of BERT, we proceed as follows: we fine tune BERT with the training split of the Hahackathon 2021 dataset for 2 epochs and a batch size of 64. Then, we freeze the resulting model and concatenate it to the LF in a new model composed of two hidden layers of 32 and 16 neurons respectively and trained for 10 epochs. In case of PWE, we evaluate word embeddings from fastText and GloVe but also we evaluate that the weights of the embeddings were learned from scratch in the embedding layer. Next, the LF and the word embeddings are concatenated, and used as input to a MLP, in which we evaluated (1) the number of hidden layers (between 1 and 8), (2) the number of neurons (8, 16, 48, 64, 128, 256), and (3) the shape of the network (*funnel*, *rhombus*, *long_funnel*, *brick*, *diamond*, and *triangle*). We also evaluate the dropout rate to avoid overfitting (0, 0.2, 0.5, and 0.8), the activation function of the hidden layers (*relu*, *sigmoid*, *tanh*, *selu*, and *elu*), the learning rate, and the batch size (16, 32, and 64). Each combination of parameters was evaluated during 1000 epochs with an early stopping mechanism. Due to time constraints, we evaluated only 1000 combinations of these hyperparameters randomly selected.

5 Results

First, the best hyper-parameter combinations (validation set) are shown in Table 1.

[2] https://github.com/autonomio/talos (v0.6.6)

Subtask 1a							
ACC	**architecture**	**features**	**shape**	**# layers**	**1st neuron**	**dropout**	**PWE**
83.926	MLP	lf+we	triangle	7	48	-	none
83.828	MLP	lf+we	long_funnel	7	16	-	fastText
83.809	CNN	lf+we	diamond	7	256	0.2	fastText
83.633	MLP	lf+we	brick	4	256	-	none
83.613	MLP	lf+we	rhombus	5	48	-	fastText
Subtask 1b							
RMSE	**architecture**	**features**	**shape**	**# layers**	**1st neuron**	**dropout**	**PWE**
0.79820	CNN	lf+we	funnel	2	256	-	gloVe
0.81080	BIGRU	lf+we	brick	5	64	0.5	fastText
0.81272	BILSTM	lf+we	brick	2	128	0.2	glove
0.81958	BILSTM	we	brick	3	48	0.2	glove
0.82004	LSTM	we	triangle	4	128	0.5	fastText
Subtask 1c							
ACC	**architecture**	**features**	**shape**	**# layers**	**1st neuron**	**dropout**	**PWE**
61.125	BILSTM	lf+we	triangle	1	48	0.5	gloVe
60.688	CNN	we	brick	2	48	-	gloVe
59.625	BILSTM	lf+we	triangle	4	48	0.2	gloVe
59.125	LSTM	we	triangle	3	256	0.5	fastText
59.000	LSTM	we	brick	5	8	0.5	none
Subtask 2a							
RMSE	**architecture**	**features**	**shape**	**# layers**	**1st neuron**	**dropout**	**PWE**
0.68037	CNN	we	triangle	4	128	0.5	fastText
0.68085	BIGRU	we	triangle	7	48	-	fastText
0.68399	LSTM	we	triangle	4	128	0.5	fastText
0.70100	CNN	lf+we	diamond	8	48	-	gloVe
0.70353	CNN	lf+we	funnel	2	256	-	gloVe

Table 1: Results of the best five hyperparameter combination for each subtask trained and evaluated with the training dataset with a ratio of 80-20.

We can observe that MLP and CNN perform better for subtask 1a, whereas recurrent neural networks achieve better results in subtasks 1b, 1c, and 2a. Regarding the feature sets, we observe that for subtasks 1a, 1b, and 1c, the combination of LF and WE achieved better results whereas only word embeddings achieved better results in subtask 2a. It draw our attention that the shape of the neural network (*shape*, *# layers*, and *1st neuron*) and the dropout rate vary according to the subtask. For example, in subtask 1a, four of the fifth best results were achieved without dropout, whereas for subtask 1b and 2a, a high dropout (0.5) resulted in better results.

Next, we compare our results with the rest of the participants and the baselines (see Table 2) with the test dataset. The organizers of the task provided two baselines based on a Naive Bayes for the classification subtasks (1a, 1c) and a Support Vector Regression for the regression subtasks (1b, 2a); both trained with a bag of words. In subtask 1a we achieve an F1-score of 91.6% and an accuracy of 93.25% reaching position 45. The best result is for *PALI* with an F1-score of 98.54% and an accuracy of 98.2%. In subtask 1b, we achieve an RMSE of 0.8847, reaching position 47 and falling below the baseline. The best result is for *abcbpc*, with an RMSE of 0.4959. For subtask 1c, we achieve an F1-score of 57.22% and an accuracy of 46.5%, reaching position 14. The best result is for *PALI*, with an F1-Score of 63.02% and an accuracy of 49.43%. In subtask 2a we achieved an RMSE of 0.8740, reaching position 46 and falling below the baseline. The best result was for *DeepBlueAI* with an RMSE of 0.412.

During the hyper parameter tuning stage, we evaluated the reliability of the PWE without the LF with our custom training and evaluating splits. The results in our development dataset were slightly better with the combination of both feature sets in three subtasks: 83.516% (LF+PWE) vs 83.379% (PWE) of accuracy in subtask 1a, 0.79820 vs 0.81958 of RMSE in subtask 1b, and 61.125% (LF+PWE) vs 60.688% (PWE) of accuracy in subtask 1c. However, in subtask 2a, PWE performed better without LF: 0.68037 (PWE) vs 0.70100 (LF+PWE).

Subtask 1a			
#	**Team**	**F1**	**Accuracy**
1	PALI	98.54	98.20
2	stce	98.54	98.20
3	DeepBlueAI	96.76	96.00
4	SarcasmDet	96.75	96.00
45	**UMUTeam**	**91.60**	**93.25**
53	baseline	88.40	88.57
Subtask 1b			
#	**Team**		RMSE
1	abcbpc		0.4959
2	mmmm		0.4977
3	Humor@IITK		0.5210
4	YoungSheldon		0.5257
46	baseline		0.8609
47	**UMUTeam**		**0.8847**
Subtask 1c			
#	**Team**	**F1**	**Accuracy**
1	PALI	63.02	49.43
2	mmmm	62.79	46.99
3	SarcasmDet	62.70	46.99
4	ThisIstheEnd	62.61	46.02
14	**UMUTeam**	**57.22**	**46.50**
31	baseline	46.24	43.74
Subtask 2a			
#	**Team**		RMSE
1	DeepBlueAI		0.4120
2	mmmm		0.4190
3	HumorHunter		0.4230
4	abcbpc		0.4275
42	baseline		0.6415
46	**UMUTeam**		**0.8740**

Table 2: Comparison of our results with other participants and the baseline for each subtask

6 Conclusions

While we are pleased with our participation since it has given us the opportunity to evaluate novel techniques and to improve our methods, we consider our results to be far from competitive. It should be noted that, for reasons unrelated to this competition, we did not have time to do all the tests we wanted. On the one hand, the final labels of the development dataset were published before the final stage, but we did not fit the models with this new information. On the other hand, we only submit one run in the final stage. Compared with the rest of the participants, we observe they submitted an average of 7.5737 runs (σ of 8.1720). However, although we believe that our results could have been somewhat better, there is still a long way to go. First, it caught our attention that BERT does not outperform the results achieved with dense,

recurrent, and convolutional neural networks for subtasks 1b, 1c, and 2a. At this respect, we will review our pipeline to detect weakness. It also draw our attention that our results did not beat the baselines in the regression tasks which indicates some kind of implementation or conceptualization error. Second, we only evaluated a subset of the LF that we had for Spanish. Accordingly, we will adapt UMUTextStats to English and compare their reliability with de-facto tools like LIWC. Third, we will focus on the interpretability of the models, as we believe that LF result in more interpretable models. Finally, we will evaluate machine learning ensembles as a mean of combining the LF.

Acknowledgments

This work was supported by the Spanish National Research Agency (AEI) through project LaTe4PSP (PID2019-107652RB-I00/AEI/10.13039/501100011033). In addition, José Antonio García-Díaz has been supported by Banco Santander and University of Murcia through the industrial doctorate programme.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

José Antonio García-Díaz, Mar Cánovas-García, Ricardo Colomo-Palacios, and Rafael Valencia-García. 2021. Detecting misogyny in spanish tweets. an approach based on linguistics features and word embeddings. *Future Generation Computer Systems*, 114:506–518.

José Antonio García-Díaz, Mar Cánovas-García, and Rafael Valencia-García. 2020. Ontology-driven aspect-based sentiment analysis classification: An infodemiological case study regarding infectious diseases in latin america. *Future Generation Computer Systems*, 112:641–657.

Soroush Javdan, Behrouz Minaei-Bidgoli, et al. 2020. Applying transformers and aspect-based sentiment analysis approaches on sarcasm detection. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 67–71.

Tonglin Jiang, Hao Li, and Yubo Hou. 2019. Cultural differences in humor perception, usage, and implications. *Frontiers in psychology*, 10:123.

J.A. Meaney, Steven R. Wilson, Luis Chiruzzo, Adam Lopez, and Walid Magdy. 2021. Semeval 2021 task 7, hahackathon, detecting and rating humor and offense. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Tomás Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2017. Advances in pre-training distributed word representations. *CoRR*, abs/1712.09405.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

María del Pilar Salas-Zárate, Giner Alor-Hernández, José Luis Sánchez-Cervantes, Mario Andrés Paredes-Valverde, Jorge Luis García-Alcaraz, and Rafael Valencia-García. 2020. Review of english literature on figurative language applied to social networks. *Knowl. Inf. Syst.*, 62(6):2105–2137.

María del Pilar Salas-Zárate, Mario Andrés Paredes-Valverde, Miguel Ángel Rodríguez-García, Rafael Valencia-García, and Giner Alor-Hernández. 2017. Automatic detection of satire in twitter: A psycholinguistic-based approach. *Knowl. Based Syst.*, 128:20–33.

Hannes Ritschel, Ilhan Aslan, David Sedlbauer, and Elisabeth André. 2019. Irony man: Augmenting a social robot with the ability to use irony in multimodal communication with humans. In *AAMAS '19*, page 86–94, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Yla R Tausczik and James W Pennebaker. 2010. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54.

Pascal Vrticka, Jessica M Black, and Allan L Reiss. 2013. The neural basis of humour processing. *Nature Reviews Neuroscience*, 14(12):860–868.