RoMa at SemEval-2021 Task 7: A Transformer-based Approach for Detecting and Rating Humor and Offense

Roberto Labadie Tamayo
Universidad de Oriente
Cuba
rlabadiet@gmail.com

Mariano J. Rodriguez Cisneros
Universidad de Oriente
Cuba
mjasoncuba@gmail.com

Reynier Ortega-Bueno
PRHLT Research Center
Universitat Politècnica de València
Valencia Spain
rortega@prhlt.upv.es

Paolo Rosso
PRHLT Research Center
Universitat Politècnica de València
Valencia Spain
prosso@dsic.upv.es

Abstract

In this paper we describe the systems used by the RoMa team in the shared task on *Detecting and Rating Humor and Offense (HaHackathon) at SemEval 2021*. Our systems rely on data representations learned through fine-tuned neural language models. Particularly, we explore two distinct architectures. The first one is based on a Siamese Neural Network (SNN) combined with a graph-based clustering method. The SNN model is used for learning a latent space where instances of humor and non-humor can be distinguished. The clustering method is applied to build prototypes of both classes which are used for training and classifying new messages. The second one combines neural language model representations with a linear regression model which makes the final ratings. Our systems achieved the best results for humor classification using model one, whereas for offensive and humor rating the second model obtained better performance. In the case of the controversial humor prediction, the most significant improvement was achieved by a fine-tuning of the neural language model. In general, the results achieved are encouraging and give us a starting point for further improvements.

1 Introduction

Detecting humor has become a popular research field at the same time that the bad phenomenon of offensiveness spreading exaggeratedly grows in social media. In this scenario it is very frequent to find out alarming volumes of heterogeneous data such as textual messages, images, advertisements, etc., that harm some age groups, ethnicity, sexual gender or other demographic characteristics (Betul Keles and Niall McCrae and Annmarie Grealish , 2020). Most of these harmful contents are often masquerade as innocent jokes or simply as a funny content. Therefore, it is crucial to shed light on the commonalities and differences between both phenomena in order to properly addressing the challenge of computationally distinguishing humorous messages from aggressive or offensive ones.
Recognizing humorous and offensive utterances on written messages is a very difficult task for human beings and even more for computers (Waseem, 2016). These difficulties increase when the textual messages are isolated from the context in which they are produced. Additional knowledge from gestures, prosody features, visual content, situational environment and sociocultural rules play an important role in how humans properly understand the real meaning behind funny and hateful contents. All this makes humor recognition and offensiveness detection challenging tasks within Natural Language Processing (NLP) and Human-Computer Interaction (HCI). On this line, the *Task 7, HaHackathon: Detecting and Rating Humor and Offense at SemEval-2021* aims at computationally recognizing humor and offensiveness in English tweets (Meaney et al., 2021).

To address the four subtasks launched in *HaHackathon* we propose two distinct architectures which rely on neural language model based representation *(deep-representation)*, particularly learned by Transformer architectures. Our first architecture combines the learned representation with a SNN in order to learn in automatically way a metric for discriminating a pair of messages of

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 297–305
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

the same class from a pair of messages of different classes. Also, we considered applying a graph-based clustering method to each class independently for creating representative prototypes. These prototypes were used to build the training and testing pairs. Our second architecture relied on the principle of fusing representations. For that, the *deep-representation* are mixed with linguistic information *(linguistic-representation)* and given as inputs to a linear regression model which is specialized in predicting humor and offensive scores. The paper is organized as follows: in Section 2 we briefly introduce the description of the four subtasks. Section 3 presents our proposed architectures and gives details about their modules. In Section 4 are described the experiments and results. Finally, we present our conclusions and provide interesting directions that we plan to explore in future work. The source code associated with this paper is online available on GitHub: `https://github.com/mjason98/semeval21_humor`.

2 Task Descriptions

We investigated the performance of our proposed architectures in the four subtasks introduced in *HaHackathon*: i) given a tweet determining whether it is humorous or not (*subtask 1a*); ii) given a tweet predicting the humor rate in the range of 0 to 5, where 0 indicates that it is not a funny message and 5 indicates that the message is strong humorous (*subtask 1b*); iii) given a tweet determining whether it is considered as controversial (i.e., it is rated with highly variable values of humor from one annotator to another) (*subtask 1c*); the last subtask, iv) given a tweet predicting its offensiveness rating in a range of 0 to 5, where 0 indicates the tweet does not contain any kind of offensiveness and 5 indicates that the message is strong offensive (*subtask 2*).

Organizers provided a dataset for training and test labeled according to the objectives of each subtask. The whole dataset was manually annotated by several annotators in order to minimize the noise in the data and increase the agreement in the annotation procedure. The dataset is composed by 8000 tweets for training and 1000 tweets for testing purposes, respectively. The training set contains 3068 funny and 4932 non-funny tweets. This slight imbalance in the training set imposes an additional difficulty to the learning algorithm for accurately predicting the funny messages. The problem increases in the task of controversial humor prediction where only 2465 tweets are labeled as controversial and the remainder 5535 are non-controversial. The most complex scenario regarding the data distribution is appreciated in the tasks of humor and offensiveness rating. At a first glance on the Figure 1 can be inferred that the majority of the offensive scores are accumulated in the interval (0,1). As a consequence of that, the tweets which are not offensive at all or those with scores closer to zero are over-represented whereas the tweets with strong offensive content are under-represented in the dataset. Therefore, from the learning perspective, it is more difficult to score tweets which strong offensive content. Conversely to this scenario, the funniness scores are distributed more uniform. Also, it is important to highlight that tweets with offensive scores greater than 0 in most cases also were scored as funny tweets. This relation reveals the usage of some humor devices as a way for masqueraded offensive messages.

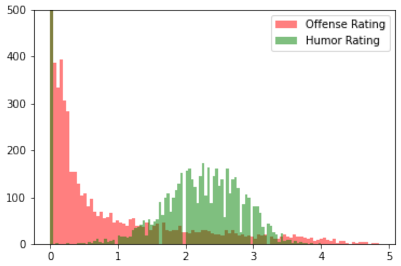

Figure 1: Histograms of humor and offensiveness score distribution

3 Our Proposals

In this section we present the proposed models and provide details about their modules. Our models have a modular structure. They are composed of both, an encoder module (*Encoder*) and a prediction module (*Classifier*), which are trained independently. Particularly, we evaluate two distinct methods for the classification module. The first one is based on a Siamese neural network and the second one relies on fusing representations and training a linear regression model.

3.1 Encoder Modules

The Encoder plays an important role because it is concerned with learning an abstract representation that vanishes the colinearity between its features and compresses the textual information on a single dense vector. In our proposal, the encoders are based on Transformer models (TM), specifically on RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2020) architectures. Moreover, we employed BERTweet (Nguyen et al., 2020) which is based on the structure and pre-training procedure like RoBERTa, but using an English tweets corpus that makes it easier to fine-tune on NLP tasks where the texts are short and informal.

For fine-tuning the TM-based encoders we add up an intermediate layer that receives the vectors from the output sequence of the TM. On this sequence of vectors, we explore three variants for selecting the best way of representing the message: i) the vector in the first position (associated to the *CLS* token), ii) the normalized sum of all vector in the sequence, and iii) the vector in the last position of the sequence. On this layer, we stacked an output layer that makes the final prediction for the targeted task. For that purpose, we follow the strategy proposed in the Universal Language Model Fine-Tuning (ULMFiT) (Howard and Ruder, 2018). For each layer of the TM a different learning rate is set up, increasing it using a multiplier while the neural network gets deeper. This multiplier increases 0.1 points from a layer L_i to another L_{i+1}. We use this dynamic learning rate to keep most information from the pre-training at shallow layers and biasing the deeper ones to learn about the specific tasks.

On the humor predicting subtask the BERTweet encoder was employed, whereas for offensiveness rating the three TMs were considered and trained using a multitask learning strategy for predicting offensive scores and irony together. Particularly, for irony detection we used the data proposed in *Task 3: Irony Detection Task at SemEval 2018* (Van Hee et al., 2018).

3.2 Classification Modules

In this section we describe the architecture of the two proposed classification modules.

3.2.1 SiaNet

To address the humor detection task we propose a SNN (Koch et al., 2015; Bromley et al., 1993) whose functionality lies on extracting features from the input messages, in such a way that a pair of messages belonging to the same class are closer and in case of belonging to opposite class move away w.r.t a distance function. In this work we use the Euclidean distance. The distance learned by this network is used as a criterion to determine, given an unlabeled message, whether it is more likely to belong to the positive class (e.g. humorous) than to the negative class (e.g. non-humorous). For that purpose, we define in each class a set of prototypes which are used to compare against the unlabeled message. These prototypes are obtained by means of a graph-based clustering method. After having the clusters, for each of them is selected a prototype (real message), which is able to represent the most information contained on that group.

Prototype Selection Strategy

The SiaNet model requires a pair of messages as input in both training and test phases. During the training stage, pairs of two labeled messages are used, and in the test phase, the label of a new message is predicted considering its similarity with positives (humorous) and negatives (non-humorous) messages. As consequence, the methods employed to obtain the pairs and select the humor and non-humor messages for comparing at the training stage, impact directly on the learning process of the model.

In this work, instead of sampling positive and negative messages randomly, we propose to include an additional step that aims at obtaining prototypical instances (*prototypes*, henceforth) of both classes. For that, we build a graph of β-distance, analogous to the β-similarity graphs proposed in (Garcia, 2005), for the humor (G_P) and non-humor (G_N) classes. The nodes in the graphs represent the tweets from the training set and the edges joining two nodes are weighted with the distance between them.

In the β-distance's graphs the edges with weights greater than the threshold β are removed, allowing only the closest representations being in the same connected subgraph. Notice that, the representation of the messages associated to nodes are obtained from the Encoder module. Afterwards, we detect communities on the β-distance's graph G_P and G_N respectively, using the InfoMap (Edler et al., 2020) algorithm based on the map equation (Rosvall et al., 2009). The map equation is a flow-based and information-theoretic method. By minimizing it over all the possible network partitions, InfoMap reveals important aspects of the network structure with respect to the dynamics on the network.

As result it is obtained a set of subgraphs $g_P^i \in G_P$ and $g_N^i \in G_N$ and the nodes they contain with their respective flow values. For each subgraph

g_C^i, $C \in [P, N]$ we select the node x_{max} with the highest flow value. We assume that this node acts as a representative node for g_C^i, and consequently it is also a prototypical message for the class C. All prototypical messages for the humorous and non-humorous classes are obtained and defined as Humor Prototype Set (P_{Set}) and Non-Humor Prototype Set (N_{Set}) respectively. In Figure 2 are depicted the projection of each class messages, and their respective prototypes.

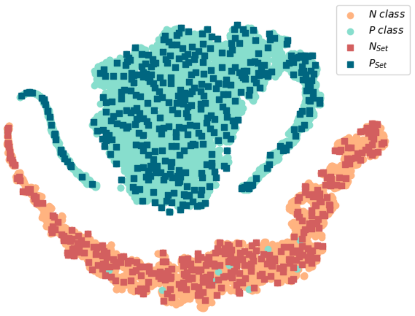

Figure 2: Scatter encoder representations per class with the identified prototypes

Siamese Neural Net Architecture

The network architecture consists of two input messages and one output that indicates how distant they are according to their representation (Bromley et al., 1993). Both messages are encoded by using the fine-tuned Transformers model (see Section. 3.1). Later, each input is passed through two dense layers (with 64 hidden neurons), which map the encoding to a smaller dimension by learning specific features. We must annotate that both input messages are fed to the same two dense layers (i.e., the new encodings are computed using the same weights in both cases). Later, the representations of the messages are compared to each other through a distance metric. The specific features the model learns to extract, make that messages representations corresponding to opposite classes have a distance greater than the threshold defined in the loss function used. Particularly, we used the *Contrastive Loss* (Hadsell et al., 2006) with a threshold of 0.85, this value was set empirically.

For training the SNN, the dataset needs to be processed for constructing pairs of messages from the same class and pairs of messages from distinct classes. Once defined the sets of prototypes P_{Set} and N_{Set} (as described in the Prototype Selection Strategy), we create training examples associated to each message x into the training dataset, with $x \notin P_{Set} \cup N_{Set}$. For that, we sampled randomly k *intra-class* examples, by pairing x with prototypes from its class, and generate m *inter-class* examples paring x with their closest prototypes from the contrary class.

During the test phase, given an unlabeled message, we obtain the encoding of z by using the Encoder module. After that, we predict the distance of z with respect to the prototypes in the P_{Set} and N_{Set} using the SNN. Based on the previously computed distances, we evaluate two rules for deciding whether z should be classified under the humorous or non-humorous classes:

i) *Minimum*, we assign z to the class of its nearest prototype as follows:

$$\hat{y} = arg \min_i \{SNN(z, x_{i,j})\} \tag{1}$$

where $x_{i,j}$ is the prototype message j with label $i = \{0, 1\}$.

ii) *Mean*, we assign z to the class of the Prototype Set with lowest average distance:

$$S_i = \frac{1}{C_i} \sum_{j=1}^{C_i} SNN(z, x_{i,j})$$
$$\hat{y} = arg \min_i \{S_i\} \tag{2}$$

where $C_i \in \{|P_{Set}|, |N_{Set}|\}$ and $x_{i,j}$ is the prototype message j with label $i = \{0, 1\}$.

3.2.2 Multiview-based Linear Regression Module

Ensemble methods usually combine data representations or the decisions of multiple models to obtain improved results over those obtained individually. These decisions are made from valuable features extracted by models' intermediate layers, which vary depending on their architecture and the dataset they have been trained on.

Combining all those information into a single prediction unit instead of synthesized predictions, is consistent if we seek to take into account different views of the information, especially when dealing with such complex and subjective tasks as offensiveness detection and in general sentiment analysis are.

We propose to fusing four distinct representations of the tweets and use this mixing deep-features for training a linear regression method. Three of the representations are based on fine-tuned transformer encoders and the other is based on affective features.

Encoder Settings

Considering the underlying relation between humorous and offensive language observed in the HaHackathon dataset (please, see Figure 1), the relation among offensiveness with other forms of toxic speech (e.g. aggressiveness and hate) presented in (Poletto et al., 2020) and the common usage of figurative devices like irony in social media for communicating indirectly hateful messages (Cignarella et al., 2018; Frenda, 2018). We fine-tuned the RoBERTa-based models on the dataset provided in the shared tasks HatEval 2019 (Basile et al., 2019), OffensEval 2019 (Zampieri et al., 2019), Irony Detection Task at SemEval 2018 (Van Hee et al., 2018) and HaHackathon itself. The fine-tuning was carried out using a smooth learning rate on the Masked Language Modeling (*MLM*) task. We masked randomly 15% of the tokens from each message, and fit them for three epochs, following the strategy proposed in (Liu et al., 2019).

For training the Encoders to address the HaHackathon specific target, the placed intermediate layer after the encoder heads, is fed with the concatenation of the three variants to get the TM output (see Section. 3.1). This layer is the one employed to obtain the message encodings. We combine the offensiveness rating in HaHackathon with the labels of irony in the dataset of *SemEval 2018 Task 3* (Van Hee et al., 2018). This idea relies on the observed relation between humor and offensiveness ratings within the provided data, where many offensive messages can be considered as ironic or harmful forms of humor (see last two examples in Table 4).

To avoid outliers in the dataset for misleading the training process, we employed the *Minkowski error* (Bishop, 1995) in the regression subtask, which is less sensitive to outliers than the standard mean squared error. It is defined as follows:

$$Error_{Minkowski} = \frac{\sum(|y-\hat{y}|)^{kc}}{n} \qquad (3)$$

Where y is the label for one example, $\hat{y}$ is the predicted value, n the number of examples and kc the Minkowski coefficient which we set to 1.4 empirically.

The Affective Features

Conversely to the three representations above, the Affective Features representation was obtained from a word-level recurrent neural network, trying to capture how affective information from different dimensions flows along the messages (Kar et al., 2018). For this purpose, we constructed an embedding matrix whose features were based on an affective information set proposed by (Farías et al., 2016) containing basic emotions (i.e., sadness, surprise, fear, etc). The embedding vectors involved 52 components between binary and no binary values, and the vocabulary was built from the affective resources, hence words not expressing emotional charge at all were encoded with the null vector.

The information obtained from this embedding for a message was fed into an BiLSTM (*Bidirectional Long Short Term Memory*) architecture similar to ELMO (Peters et al., 2018). Deeper BiLSTM output was fed, to an intermediate layer to condense the information passed later to the output layers. For training this model we used a multitask approach focusing on predicting how offensive a message is, as well as how funny it is, by using the data provided for HaHackathon.

Linear Regression

As we can be observed in Figure 3, the encoding provided by transformer models differ regarding the space region in which the offensive features are projected. We can infer that one representation helps the others by providing information not captured simultaneously.

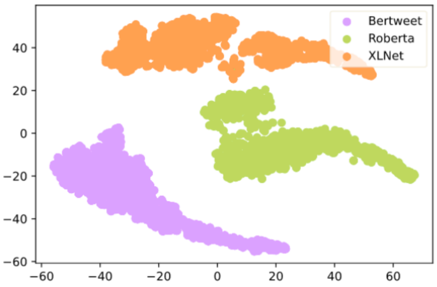

Figure 3: Scatter encoders representations

Considering that, there is no co-linearity between the features extracted from one encoder to another. We hypothesize they can be combined through a parsimonious model to prevent overfitting. Based on that, we decided to employ a Ridge Regression Model, setting the α hyper-parameter employed for the L2 regularization on the loss function to 1.0.

During the experiments we also construct another ensemble based on Recurrent Neural Networks (RNN) which receive all four encodings and treat them as a sequence of the message. The elements of this sequence are weighted through an Additive Attention layer and combined employing an LSTM layer in order to decide from a time step *i.e., from one encoding to another*, which of the features must be kept through the entire sequence analysis *i.e., all four encodings*. The output of this layer is then fed into two parallel output layers to predict whether a message is offensive or not and its offensiveness degree.

The data over-representation for messages with offensiveness rating equal to 0 makes that, from the standpoint of deciding whether a message is offensive or not, the data be balanced (i.e., labeling the messages with offensiveness rating higher than 0 as offensive). This allowed to us using the alternative classification task in the multitask learning approach employed for this model, taking into account that it helps to learn common features among these offensiveness-related tasks.

4 Experiments and Results

In this section we describe the conducted experiments for evaluating the performance of our systems on HaHackathon development dataset (*dev-dataset*). For that, we employed the metrics proposed by the task organizers, *F1-score* over the positive class and accuracy *(Acc)* for classification subtasks, and the Root Mean Squared Error *(RMSE)* for regression subtasks.

Encoder Modules

The SiaNet model and the Ridge Regression model are fed with information of the messages extracted through their respective encoder modules, this makes our first effort focused on tuning them for obtaining the best representation. For both approaches, SiaNet and Ridge Regressor, the encoders were optimized using the RMSprop method (Hinton et al., 2012).

Firstly, for obtaining the multi-viewed representation of the messages, the RoBERTa, BERTweet and XLNet encoders were fine-tuned using *(MLM)* unsupervised learning. For that, we considered three additional related-datasets (Basile et al., 2019; Zampieri et al., 2019; Van Hee et al., 2018) and the HaHackathon dataset itself. Afterwards, since the multi-viewed representation was constructed for rating offensiveness, the three models were trained specifically for this regression task by exploring two main ideas based on multitask learning strategy *(MTL)*: i) The first one aims at capturing the information shared among the four subtasks proposed in the HaHackathon dataset; ii) The second one, aims at capturing the indirect negative speech behind humorous messages, for that we introduced the irony prediction task *(Irony)* combined with offensiveness prediction. Specifically, we used the dataset proposed in (Van Hee et al., 2018) for addressing the irony prediction task.

Table 1 shows the results of applying both strategies for each transformer encoder. As can be observed, by making the model to extract features also useful for irony detection we achieved the best performance. Nevertheless, the first strategy yields our best result at predicting whether or not a message can be considered as controversially humorous. We also tried to avoid using MTL with the three transformers encoders, fine-tuning them for the offensiveness regression subtask, but in terms of RMSE the performance decreased on 0.07 in average.

Model	Strategy		
	HAHA	*Irony*	*No MTL*
BERTweet	0.70	0.65	0.81
RoBERTa	0.75	0.63	0.67
XLNet	0.69	0.68	0.70

Table 1: MTL strategies for offensiveness rating subtask. *HAHA* refers to MTL with all HaHackathon subtasks and *Irony* refers to MTL with irony detection task

Similarly it happened when it was not accomplished the *MLM* fine-tuning. The error was slightly increased for RoBERTa from 0.58 to 0.64 when this stage was avoided and for BERTweet, it increased from 0.65 to 0.91. We hypothesize this technique helped the model to reduce the impact of isolated offensive terms, which may influence the regression stage on messages that are not even offensive.

For fine-tuning the encoder module of SiaNet we explored if it was more convenient to set a single learning rate for the whole model or follow the ULMFiT strategy addressing the humor prediction task. The second approach obtained the best performance in terms of F1-Score/Acc with (0.94/0.92) w.r.t (0.90/0.88) reached by the first one. We also tried to apply MTL to this approach, but this did not yield any improvement, reaching 0.93/0.91.

Prediction Modules

For classifying unlabeled tweets with SiaNet we evaluated the two methods described in Section 3.2.1 alongside the upper bound of clusters extracted from G_P and G_N by InfoMap (50, 250 and 300 clusters) and how the TM output sequence was used according to the strategies described in Section 3.1. Among the combinations resulting from that evaluation, the best-performed was the one involving the *minimum* criterion for labeling the messages, the highest upper bound for allowed instances on P_{Set} and N_{Set} respectively (300) and taking the normalized sum of the TM output sequence, reaching under F-Score/Acc the measurements (0.9505/0.9370).

Also, we added Gaussian noise to the encoding inputs for decreasing overfitting when training the Siamese as part of the conducted experiments, resulting on improving the loss in the dev-set from 0.11 when the noise is not added to 0.06.

In the training phase of the Linear Ridge regression method we evaluated the impact of the distinct representations on the performance of our model. Looking at Table 2, we noted that each transformer encoder played an important role in characterizing the messages, also the affective features captured important information about the offensive language, which helped in each combination. The LSTM based models also had a good performance when combining all the representations, especially the one with no attention mechanism.

model	XLNet	RoB	BT	AF	Off
Ridge	+	+	+	+	**0.55**
	+	+	+	-	0.61
	-	+	+	-	0.65
	+	+	-	+	0.58
	+	-	+	+	0.59
	-	+	+	+	0.62
LSTM	+	+	+	+	**0.55**
LSTM-Att	+	+	+	+	0.57

Table 2: Feature representation combination through the ensemble

Summarizing, participating in HaHackathon we addressed the humor prediction task with the SiaNet model. For the humor rating subtask we used the Multiview-based Linear Ridge Regression model, fine-tuning the transformer encoders under the humor and offensiveness rating subtasks simultaneously after applying *MLM*. The controversy humor prediction subtask was addressed through the BERTweet model using *MTL* with all four subtasks from HaHackathon. Finally, the offensiveness rating was predicted by the Multiview-based Linear Ridge Regression, but fine-tuning the encoders with *MTL* and combining offensiveness rating subtask with irony detection.

4.1 Error Analysis

In the humor prediction subtask, we found out that more than 40% of prototypes obtained from the humorous class have the structure *question?argumentation* (Q?A, see Table 3). We hypothesize that some tweets were misclassified as humorous due to sharing this structure with positive prototypes. In fact, within the examples labeled by our architecture as funny when they were not, the ones having this structure represented the 38% of this type of misclassification.

Tweet
What do you call an Asian guy that always shows up before he needs to? Earl Lee
Why did the slave go to college? So he could pickup his Master's degree.
What do you call a 60-year old whose puberty just started? A late boomer.

Table 3: Prototype tweets annotated as humor with the structure of Q?A

For the offensiveness prediction task, the most critical failures (i.e., absolute difference between the real value and the predicted one) were analyzed from two standpoints: first when the model predicts a lower value than the real one as the first two examples in Table 4 or a higher value as in the last two cases. As we can observed how it happened in the most mispredicted examples, the model gives higher offensiveness values to messages containing phrases that characterize social groups usually being a target of hate spreading or bullying on social media. This is possibly caused by the origin of data used for pre-training the transformer encoders, which were in charge of finding an encoding for the tweets.

4.2 Official Results

Regarding the official results on the test set, we made submissions in all four sub-tasks. The baseline proposed by the organizers consisted of a Naive Bayes model with bag of words features

Tweet	Value	Predicted
What do you call a **homosexual man** on a **wheel chair**? A human being	0.15	2.5
What do you call it when two female spies fall in love? **Lesbianage**	0.6	1.89
Wanna hear a **joke**? **Women's rights**.	3.35	1.79
What belongs to me but is used the most by others? **My ex-wife**	1.9	0.43

Table 4: Some examples mispredicted by our model

and Support Vector Regression for the classification and regression substasks respectively.

In subtask 1a we ranked at place 22^{nd} among 58 teams, with F-Score/Acc of 0.948/0.9576, whereas the best system reached 0.982/0.9854. In subtask 1b with a RMSE of 0.5905 and among 50 teams we ranked at 30^{th} place and the best system had an RMSE of 0.4959. For subtask 1c we obtained the 10^{th} position from 36 teams, our F-Score/Acc was 0.4732/0.6197 and the best system obtained 0.4943/0.6302. Finally, in subtask 2 we were the 14^{th} team of 48 in total, with a RMSE of 0.4532 with a difference from the best ranked system of 0.0412.

5 Conclusions and Future Work

In this work we presented two models for addressing humor and offensiveness prediction in English tweets. Both models employ the deep-representations learned by Transformers methods for encoding the messages. The first model is based on a Siamese Neural Network combined with a graph-based clustering method. The second model combines feature representations learned by three transformers language models with affective features captured by an BiLSTM-based model. These representations are used to train a linear regression model. The achieved results show that the Siamese architecture outperformed the fine-tuned Transformer models for humor detection task. The performance of this architecture relies on how the tweets are represented by the encoder and the strategy to find the Positive and Negative sets of prototypes. In the second model, the affective features play an important role to determine the offensiveness scores with any combination of features learned by the state-of-the-art language models, showing that they successfully captured underlying affective cues present in offensive and funny speech. We plan to investigate two interesting directions as future works. The first direction is an in-depth study of the harmfulness of humor on human stereotypes taking advantage of the overlapping between offensiveness and humor in the HaHackathon dataset. The second one is an exhaustive analysis of clustering methods for building prototypes and how they may influence the learning of the Siamese Neural Network for humor prediction.

Acknowledgements

The work of the last two authors was in the framework of the research project MISMIS-FAKEnHATE on MISinformation and MIScommunication in social media: FAKE news and HATE speech (PGC2018-096212-B-C31), funded by Spanish Ministry of Science and Innovation, and DeepPattern (PROMETEO/2019/121), funded by the Generalitat Valenciana.

References

Betul Keles and Niall McCrae and Annmarie Grealish . 2020. A systematic review: the influence of social media on depression, anxiety and psychological distress in adolescents. *International Journal of Adolescence and Youth*, 25(1):79–93.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Nozza Debora, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, Manuela Sanguinetti, et al. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceeding of the 13th International Workshop on Semantic Evaluation*, pages 54–63. Association for Computational Linguistics.

Christopher M. Bishop. 1995. Neural Networks for Pattern Recognition. page 208. Oxford University Press, Inc., USA.

Jane Bromley, James W Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688.

Alessandra Cignarella Cignarella, Simona Frenda, Valerio Basile, Cristina Bosco, Viviana Patti, and Paolo Rosso. 2018. 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), Turin, Italy. CEUR.org.

Daniel Edler, Eriksson Anton, and Martin Rosvall. 2020. The MapEquation software package. *URL: https://mapequation. org*.

Delia Irazú Hernández Farías, Viviana Patti, and Paolo Rosso. 2016. Irony detection in twitter: The role of affective content. *ACM Transactions on Internet Technology (TOIT)*, 16(3):1–24.

Simona Frenda. 2018. The role of sarcasm in hate speech. A multilingual perspective. In *Doctoral Symposium of the XXXIVInternational Conference of the Spanish Society for Natural Language Processing (SEPLN 2018)*, pages 13–17. Lloret, E.; Saquete, E.; Martínez-Barco, P.; Moreno, I.

Reynaldo Jose Gil Garcia. 2005. *Algoritmos de agrupamiento sobre grafos y su paralelización*. Ph.D. thesis, Universidad Jaume I.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. 2012. Lecture 6a overview of mini–batch gradient descent. *Coursera Lecture slides https://class. coursera. org/neuralnets-2012-001/lecture,[Online*.

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Sudipta Kar, Suraj Maharjan, and Thamar Solorio. 2018. Folksonomication: Predicting Tags for Movies from Plot Synopses using Emotion Flow Encoded Neural Network. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2879–2891, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

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*.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. 2020. Resources and benchmark corpora for hate speech detection: a systematic review. In *Language Resources and Evaluation*, pages 1–47. Springer Netherlands.

Martin Rosvall, Daniel Axelsson, and Carl T Bergstrom. 2009. The map equation. *The European Physical Journal Special Topics*, 178(1):13–23.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. SemEval-2018 Task 3: Irony Detection in English Tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50, New Orleans, Louisiana. Association for Computational Linguistics.

Zeerak Waseem. 2016. Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 138–142, Austin, Texas. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv preprint arXiv:1906.08237*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.