INGEOTEC at IberLEF 2019 Task HaHa

José Ortiz-Bejar[1,3], Eric Tellez[2,3],
Mario Graff[2,3], Daniela Moctezuma[3,4], and Sabino Miranda-Jiménez[2,3]

[1] Universidad Michoacana de San Nicolás de Hidalgo, México
jortiz@umich.mx
[2] CONACyT Consejo Nacional de Ciencia y Tecnología,
Dirección de Cátedras, México
[3] INFOTEC Centro de Investigación e Innovación en Tecnologías
de la Información y Comunicación, México
{eric.tellez,mario.graff,sabino.miranda}@infotec.mx
[4] Centro de Investigación en Ciencias de Información Geoespacial A.C., México
dmoctezuma@centrogeo.edu.mx

Abstract. This manuscript describes INGEOTEC's participation in the second *Humor Analysis based on Human Annotation (HAHA)* task on IberLEF'2019. Our approach to solve the task was based on perform an extensive comparison of several text classifiers using a 80-20 holdout cross-validation methodology. We found that our generic text categorization and regression system (μTC) had the best performance. Finally, we conducted an analysis over the training dataset illustrating some of the task's complexity.

Keywords: Humor Analysis·Text Categorization·Model's Performance Analysis.

1 Introduction

Informal writing is a complex process; it is full of ambiguity, subjective, and figurative statements. A human learns to interpret this kind of language and expressions from its culture and its environment. While it is quite natural for almost anybody to understand informal language, the case of modeling it becomes a complicated task, it is full of variants and dependencies in cultural traits that become humor identification a challenging task.

Therefore, in the end, the identification process can be tackled with a large and diverse knowledge database, a robust enough model of the text, and a high performing learning algorithm. However, automatic humor detection as a supervised learning problem is complicated since the language traits that make something humorous is hard to bound in a set of rules. It is even painful to achieve agreement among humans on what is funny and what is not; therefore, a labeled dataset must be carefully curated.

The IberLEF-2019 forum ran a task devoted to Humor Analysis based on Human Annotation (HAHA). Here, a set of human-labeled messages from Twitter are provided to train and test algorithms for humor identification (classification) or ranking (regression). More detailed, each text is labeled as humorous or not humorous; a score of the humor-intensity is also given to define a ranking problem.

This manuscript describes the participation of INGEOTEC team in HAHA challenge. In addition to describing our methodology and internal comparisons, we performed an analysis of the training set to explain our performance. In particular, we use our μTC [13]. The HAHA challenge is described in Section 2. Section 3 describes our general methodology to solve the task. Section 4 is devoted to describing our systems while experimental methodology and results are discussed in Section . Finally, Section 7 concludes our contribution.

2 Task Description

Humor Analysis based on Human Annotation (HAHA) [3] asks for systems that classify tweets, in the Spanish language, as humorous or not. Also, it asks for systems that determine how funny tweets are. Those two tasks are described by *HAHA* organizers as follows:

Humor detection determining if a tweet is a joke or not (intended humor by the author or not). The results of this task will be measured using F-measure for the humorous category and accuracy. F-measure is the primary measure for this task.

Funniness score prediction predicting a funniness score value (average stars) for a tweet in a 5-star ranking, supposing it is a joke. The results of this task will be measured using the root-mean-squared error (RMSE).

The first task can be solved as a classification problem, while the second one can be tackled as a regression problem. The training set is a corpus of 24000 crowd-annotated tweets, as described in [2]. Multiple annotators evaluated each tweet, and each annotation consists of the class (humorous or not) and the intensity (number of stars 0-5). The final label is determined using a voting scheme. Table 2 shows an example of the content of the provided dataset.

Table 1. Humorous tweet example

Text:	La semana pasada mi hijo hizo un triple salto mortal desde 20 metros de altura. Es trapecista? - Era :(
English translation:	Last week my son made a triple somersault from 20 meters high. - Is it a trapeze artist? - He was :(
Is humorous:	True
Average stars:	3.25

3 Our humor detection approach

We select to use the modeling procedure for humor analysis described in our previous work [8]. Briefly, the idea is to create a single model for both classification and regression tasks, and it is based on computing a sparse or a dense vector space model, and then try different learning methods that support the data model. The sparse vector space is created through μTC using an optimized text model, based on the hyper parameter selection of the tool. For the dense modeling, we use diverse word embedding models and then summarize word's vectors into a single dense vector.

Figure 1 illustrates our generic supervised model for humor classification and regression. For our sparse vector models we start the process with the training set T, a set of short text messages; the text is preprocessed and tokenized using multiple schemes like word n-grams, character q-grams, and skip-grams. This bag of tokens is vectorized through a weighting scheme. This procedure generates the vector space X, which can be used by a classifier or a regressor, Y depict the output associated to the training set. The model's quality depends on the entire pipeline. The entire process is documented in [15].

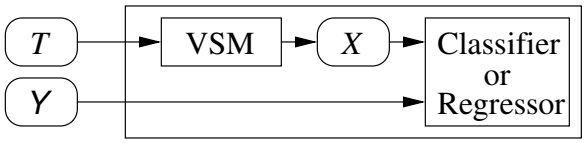


Fig. 1. The generic diagram for our humor classification and regression system.

A similar procedure is made for dense models, that is, the text is preprocessed. We use both pretrained word embeddings and computed word embeddings to model our text; X is compute using the vector sum and normalization word-vectors that compose each text.

4 Systems description

Our best solutions at both tasks were obtained using μTC system. However, we explored the use of *fastText* and *flair* along with multiple combinations of word embeddings which range from simple character to the state-of-the-art BERT. In the following sections, we describe several approaches in more detail as well as some findings to hypothesize why our attempts did not show any improvement concerning our μTC baseline.

μTC [15] is a minimalist tool that generates text classifiers that maximize a performance measurement. It manages the entire pipeline of a text classifier, as specified in the previous section. Under the hood, μTC uses a linear Support Vector Machine as the classifier. The core idea behind μTC is to define a parameter space describing a massive number of text-classifiers. The search in this space for a competitive text classifier using a set of heuristics to perform the search based on random search and hill climbing.

We also probe EvoMSA [5], our multilingual sentiment analysis system based on genetic programming. The core idea is the use of several and diverse models to solve the task using a stacking scheme guided by genetic programming. This approach is particularly robust with unbalanced classes. Besides, we tested our baseline algorithm for multilingual sentiment analysis (B4MSA) [12]; this system is a sentiment classifier for informal text such as Twitter messages. The design is similar to μTC, but the internal problem is solved differently, along with the use of specific features for sentiment analysis and some language-dependent capabilities.

Additionally, we also test third-party tools like FastText [6], which is a library for text classification and word vector representation. It transforms the text into continuous vectors that can later be used on any language related task. FastText represents sentences with a weighted bag of words, and each word is represented as a bag of character n-gram to create text vectors. This representation is based on the skip-gram model [7], which take into account subword information and sharing information across classes through a hidden representation. Also, it employs a hierarchical softmax classifier that takes advantage of the unbalanced distribution of the classes to speed up computation. In addition to the default configuration, we optimized many of the parameters of FastText along with different preprocessing functions. We used random search over a configuration space for this purpose. Finally, we also test the multilingual library Flair [1], which implements state-of-the-art NLP models, such as named entity recognition, part-of-speech tagging, sense disambiguation, and classification. Flair allows to use and combine different word and document embeddings, among which stand out flair embeddings, BERT [4] embeddings, and ELMo (see [11]) embeddings.

5 Experiments and results

We tested multiple approaches by using the tools described above. The experimental setup consisted of using the set T of 24000 tweets human annotated by the task organizers. Firstly, T was split in training (T_t) and validation (T_v) sets following a 80-20 proportion. Table 2 describe the validation and training sets.

Table 2. Training and validation sets class proportion

Set	humor	no humor	total
Training	6472	10327	16799
Validation	2782	4419	7201

5.1 Classification task

Our first intent was to use μTC and FastText with default parameters; they both were used as baselines. Significant improvement for the FastText baseline was achieved by optimizing FastText parameters. However, any further efforts to improve μTC were useless. We tried to optimize Flair by using different word embeddings combination

and EvomSA enriched by different lexicon and decision functions learned from other sentiment analysis tasks. Table 3 summarizes the results of our experiments.

Table 3. Performance of the different systems on the validation set.

System	Accuracy	Macro F1	Macro Recall	F1 humor
μTC	**0.8448**	**0.8363**	**0.8362**	**0.7989**
evomsa + Emo-TM	0.8441	0.8354	0.8349	0.7974
evomsa + Emo-HA-TH-TM	0.8425	0.8343	0.8350	0.7974
evomsa + lexicon	0.8405	0.8330	0.8353	0.7974
b4msa	0.8394	0.8394	0.8301	0.7916
evomsa	0.8390	0.8318	0.8352	0.7971
Optimized FastText	0.8289	0.8177	0.8147	0.7726
FastText Default	0.8091	0.7904	0.7816	0.7278
Flair + BERT embbedings	0.6683	0.7989	0.7953	0.7479

5.2 Analysis of task

To understand why there is no improvement over μTC, we performed an analysis of the training and validation set vocabularies to determine the similarities/differences between them. For this propose we need to detail into the weighting scheme that our μTC uses to tackle the problem, the *entropy*+b term-weighting defined in [14], it is based on representing each term by terms entropy computed from the empirical distribution of the available classes, using a smoothing parameter b, in this case, $b=3$. More precisely, defined as follows:

$$entropy_b(w) = \log|C| - \sum_{c \in C} p_c(w,b) \log \frac{1}{p_c(w,b)} \tag{1}$$

where C is the set of classes, and $p_c(w,b)$ is the probability of term w occurs in class c parametrized with b. More detailed,

$$p_c(w,b) = \frac{freq_c(w)+b}{b \cdot |C| + \sum_{c \in C} freq_c(w)} \tag{2}$$

Here $freq_c$ denotes the frequency of the term in the class c. Using this approach, it is possible to find the set of most discriminant terms among classes; we can define thresholds for any vocabulary size if we normalize $entropy_b$ by the logarithm of the size's vocabulary such that terms are weighted with values between 0 to 1.

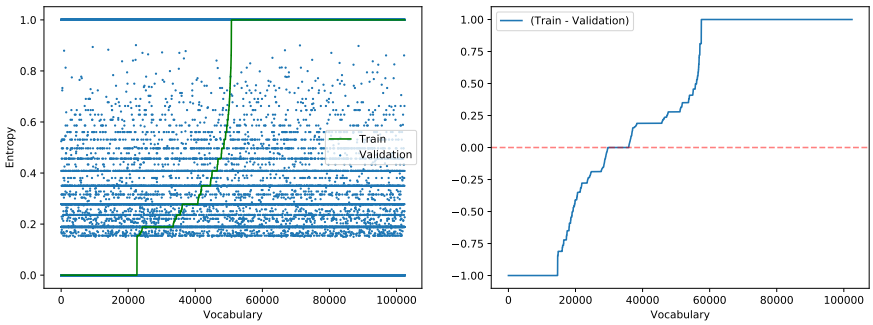
Table 4 shows the sizes of the vocabularies after removing those terms with less than 0.15 of normalized $entropy_b$. The size of the training vocabulary is close to 80 thousand items while the vocabulary of the validation set has close 40 thousand entries. Please recall that we used an 80-20 partition, but it is also under a combination of tokenizers as determined by μTC. To explain the findings, we refer to the terms set

Table 4. Terms similarities and differences for training and validation sets

Set	Cardinality
M_t	79,838
M_v	38,786
$M_t \cap M_v$	16,186
$M_t \cup M_v$	102,438

for training set as model M_t and the terms for validation as M_v. The union and intersection sizes of training and validation sets are also listed in the table.

The intersection and union sizes of Table 4 indicate the number of non-shared terms between the training and validation sets, due to it may be supposed that semantic models will be achieve good performance, however from our experiments using multiple state-of-the-art word embeddings do not outperformed μTC model. To gain more understand we produce Figure 2 by sorting all terms according its entropy at the training set, it is set a value of 0 for all terms which are not in any of M_t or M_v.



(a) Training and validation (b) Difference

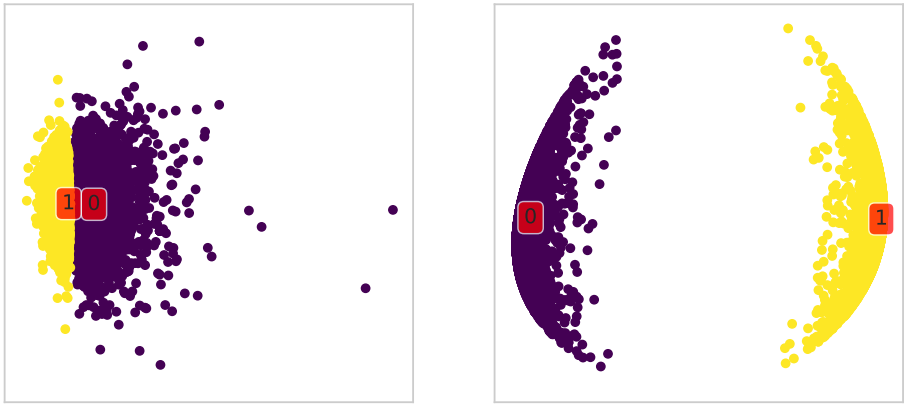
Fig. 2. Vocabulary relevance for training and validation

Figure 2(a) shows the normalized $entropy_b$ of terms. It is set a value of 0 for all terms which are not in any of M_t or M_v. The green line shows the accumulated distribution of weights for the training set, the initial values of the curve zeros since we removed terms having a weight below 0.15 to simplify the analysis. It is possible to see a large number of ones, which is the maximum discriminating value of $entropy_b$. The cloud of blue points show the density based on the entropy but for the model generated for the validation set, again the bottom of the figure is free of points since we removed low weighted terms. This figure help us to understand the density and distribution of the vocabularies and the computed weights for its terms.

On the right, Figure 2(b) shows the difference between weights in the training and validation vocabularies, seen each vocabulary as a vector with weights ranging from 0 to 1. We identified three zones in the figure: the negative, zero, or close to zero and the positive one. On the positive zone, we found those terms that discriminate better at

the training set than in the validation set. The negative zone collects those terms with a higher discriminant power in the training set. The zone around zero gathers terms that have a similar entropy score at both training and validation sets. Therefore, those terms with $entropy_b = 1$ are highly discriminant in the training set but are not part of the vocabulary of the validation set; conversely, terms with -1 weight are those that are not part of the training set but exhibits unitary entropy values in the validation set.

The previous discussion suggests that semantic models may work better; however, pure word embeddings models achieved lower performance as described in our experimental results. Trying to improve, we also tried kernel methods, specialized in working with non-linear problems; in particular, we use the technique described in [9]. This approach separates the training set while does not show any significant improvement over validation dataset scores; this behavior is an obvious symptom of overfitting.



(a) Raw vectors of the training set.

(b) Kernelized projection of the training set.

(c) Raw vectors of the validation set.

(d) Kernelized projection of the validation set.

Fig. 3. PCA projections of raw vectors and kernelized vectors for both training and validation sets.

Figure 3 shows this overfitting process. Here we applied Principal Component Analysis (PCA) to get a two-dimensional representation. The input dataset for PCA is composed of embeddings obtained by optimizing FastText and the projection by using an RBF kernel with eight references selected by using the Farthest First Traversal algorithm, as explained in [9]. The projected centroids for both classes are plotted as red squares; labels are 1 for humorous texts and 0 non-humorous messages. As can be seen from figure 3 classes overlap at the training partition is not present in projection, while for the validation set the overlap is much larger and kept after projection. It is worth to mention that almost any kernel or number of references obtain a perfect score over the training set with these kernel methods.

5.3 Regression

The regression sub-task was tackled using the classification model of μTC but replacing the Linear SVM classifier by the linear SVM regressor (SVR) available at scikit-learn [10]. This regressor was used to evaluate the submitted test dataset.

6 Task Results

Our best result using μTC was ranked fifth out of 19 contestants in both, classification and regression tasks. Table 5 shows scores for μTC, the organizer's baseline and the ones for the winner approach. For the classification task, measure F1 was used to decide the winner, while Root Mean Squared Error (RMSE) was used for the regression task.

Table 5. Comparative for our best results, winner approach and organizer's baseline

	Classification task				**Regression task**
Team	**F1(humor)**	**Precision**	**Recall**	**Accuracy**	**RMSE**
Adilism	0.821	0.791	0.852	0.855	0.736
INGEOTEC	0.788	0.758	0.819	0.828	0.822
Baseline	0.440	0.394	0.497	0.505	2.455

7 Conclusions

This paper describes the participation of the INGEOTEC team in the HAHA'19 challenge. Our final approach uses our μTC system to perform both classification and regression tasks. For this edition, we test several approaches based on different semantic models, and our stacking solution EvoMSA; however, any of them was able to beat μTC. We performed a qualitative analysis to find possible reasons for this situation. Possible reasons could be that our train and validation partitions contained very different vocabulary, semantics, and high bias, as was experimentally shown.

Acknowledgements

The authors would like to thank to the *Consorcio en Inteligencia Artificial* for partially funding this work through Project FORDECyT 296737.

References

1. Akbik, A., Blythe, D., Vollgraf, R.: Contextual string embeddings for sequence labeling. In: COLING 2018, 27th International Conference on Computational Linguistics. pp. 1638–1649 (2018)
2. Castro, S., Chiruzzo, L., Rosá, A., Garat, D., Moncecchi, G.: A crowd-annotated spanish corpus for humor analysis. In: Proceedings of SocialNLP 2018, The 6th International Workshop on Natural Language Processing for Social Media (2018)
3. 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)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Graff, M., Miranda-Jiménez, S., Tellez, E.S., Moctezuma, D.: Evomsa: A multilingual evolutionary approach for sentiment analysis. arXiv preprint arXiv:1812.02307 (2018), https://github.com/INGEOTEC/EvoMSA
6. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. pp. 427–431. Association for Computational Linguistics (April 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. 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)
9. Ortiz-Bejar, J., Tellez, E.S., Graff, M., Miranda-Jiménez, S., Ortiz-Bejar, J., Moctezuma, D., Sánchez, C.N.: I3go+ at ricatim 2017: A semi-supervised approach to determine the relevance between images and text-annotations. In: 2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC). pp. 1–6. IEEE (2017)
10. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
11. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
12. Tellez, E.S., Miranda-Jiménez, S., Graff, M., Moctezuma, D., Suárez, R.R., Siordia, O.S.: A simple approach to multilingual polarity classification in Twitter. Pattern Recognition Letters **94**, 68–74 (2017). https://doi.org/10.1016/j.patrec.2017.05.024
13. Tellez, E.S., Moctezuma, D., Miranda-Jiménez, S., Graff, M.: An automated text categorization framework based on hyperparameter optimization. Knowledge-Based Systems **149**, 110–123 (2018), https://github.com/INGEOTEC/microtc
14. Tellez, E., Miranda-Jiménez, S., Graff, M., Moctezuma, D.: Gender and language-variety identification with MicroTC: Notebook for PAN at CLEF 2017. In: CEUR Workshop Proceedings. vol. 1866 (2017)
15. Tellez, E., Moctezuma, D., Miranda-Jiménez, S., Graff, M.: An automated text categorization framework based on hyperparameter optimization. Knowledge-Based Systems (2018). https://doi.org/10.1016/j.knosys.2018.03.003