MALNIS at IberLEF-2022 DETESTS Task: A Multi-Task Learning Approach for Low-Resource Detection of Racial Stereotypes in Spanish

Juan Ramirez-Orta[1,*], María Virginia Sabando[2,3], Mariano Maisonnave[1,3] and Evangelos Milios[1]

[1]*Faculty of Computer Science, Dalhousie University, 6050 University Avenue, Halifax, NS B3H 1W5, Canada*
[2]*Institute for Computer Science and Engineering (UNS-CONICET), San Andrés 800, Bahía Blanca, Buenos Aires, Argentina.*
[3]*Department of Computer Science and Engineering, Universidad Nacional del Sur, San Andrés 800, Bahía Blanca, Buenos Aires, Argentina.*

Abstract

This paper describes our submission for the DETESTS (DETEction and classification of racial STereotypes in Spanish) shared task at IberLEF 2022. The DETESTS shared task is divided into two sub-tasks: in the first one, the objective consists of detecting racial biases in online comments as a binary classification problem, whereas in the second one, the goal is to determine whether the comments exhibit one or more of ten different racial biases as a multi-label classification problem. Our approach consists of a Multi-Task Learning strategy applied to pre-trained deep language models, which allows to learn a sequence representation for each comment. This representation is then used to train a joint classifier for all the categories of the second task, combining them using $LOGICAL\_OR$ to produce the predictions for the first one. The intuition behind our approach is that the joint training process allows the model to leverage the information present in each one of the categories and benefit from how they complement each other, boosting the performance of those categories with less examples. Our approach obtained ninth place in the first task and first place in the second one. We provide the source code to reproduce our results at https://github.com/jarobyte91/detests_2022.

Keywords

Multi-Task Learning, Multi-Label Classification, Natural Language Processing, CEUR-WS

1. Introduction

The DETESTS (DETEction and classification of racial STereotypes in Spanish) shared task [1] consists of detecting and classifying racial biases and stereotypes in short text fragments. These fragments come from comments and replies to various online news articles related to immigration, written in Spanish, and can simultaneously contain one or more stereotypes within ten different categories. The DETESTS task comprises two sub-tasks: *Task 1*, consisting

IberLEF 2022, September 2022, A Coruña, Spain.
*Corresponding author.
juan.ramirez.orta@dal.ca (J. Ramirez-Orta); virginia.sabando@cs.uns.edu.ar (M. V. Sabando); mariano.maisonnave@dal.ca (M. Maisonnave); eem@cs.dal.ca (E. Milios)
0000-0002-4385-7149 (M. V. Sabando); 0000-0002-0184-8009 (M. Maisonnave); 0000-0001-5549-4675 (E. Milios)

of the detection of racial bias in the comments as a binary classification problem, and *Task 2*, which consists of detecting each stereotype category separately and modeling the bias detection as a multi-task problem.

The detection of racial bias and stereotypes in online comments constitutes a challenging task from different perspectives. On one hand, due to the inherent bias in existing data sets, many existing approaches tend to wrongly associate non-biased triggering terms related to race, gender, sexual orientation or religion with hate speech or explicit bias [2, 3]. On the other hand, there are social constructs and situational factors that contextually bias an otherwise seemingly non-biased comment, which are often missed by language models. Moreover, racial biases might be implicit in text, for example by means of sarcasm, mockery or irony. Finally, the different types of implicit biases are often complementary or correlated, which analyzed separately might result in poor overall stereotype detection performance.

Multi-Task Learning (MTL) [4] constitutes a promising approach for those scenarios where it is necessary to model different targets simultaneously. MTL allows building and training predictive models for arbitrary combinations of tasks. Usually, an MTL approach consists of a set of shared layers which are used to build a shared representation of the input, followed by a set of individual layers for each task. We present a depiction of such an approach in Figure 1.

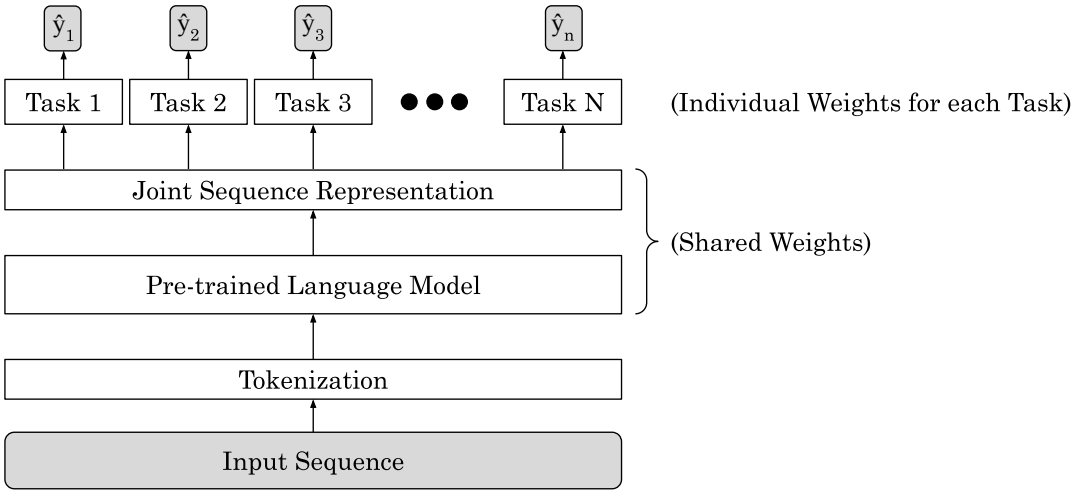

Figure 1: Overview of a Multi-Task Learning approach.

In particular, neural-based MTL models allow to efficiently perform joint learning of the different predictive targets during the training stage, therefore exploiting the information of each task while at the same time benefiting from their complementarity. It has been observed that MTL models attain high predictive performance in scenarios where several related but different tasks are involved, especially when it comes to low-resource problems and diverse data sets.

Several research efforts have been made in recent years for racial bias and stereotype detection, hate speech automatic moderation, and sentiment analysis. Most of the approaches found in

the literature are built upon pre-trained deep language models for Natural Language Processing (NLP), such as BERT [5] and RoBERTa [6], and were later fine-tuned for the specific task under study, attaining state-of-the-art results[3, 7, 8].

Multilingual [9] and Spanish text approaches [10, 11, 12, 13] are particularly relevant for the DETESTS task, successfully employing multilingual and Spanish versions of these pre-trained language models, like BETO [14], Multilingual RoBERTa [6] and Multilingual BERT [5].

MTL strategies have also been effectively employed for hate and stereotype detection [15, 2], sentiment classification [16] and toxic speech detection [13]. In particular, Plaza-del Arco et al. [13] combined an MTL approach with pre-trained multilingual deep language models, which were fine-tuned on a variety of public data sets related to toxic speech and bias detection, and reached the first place in the DETOXIS 2021 task [17], thus demonstrating the potential of such techniques for stereotype detection.

In this scenario, we designed a model to detect stereotypes and racial bias in short text fragments in Spanish. Our model is based on pre-trained Spanish deep language models that are subsequently fine-tuned by taking into account the information of each of the ten different categories of stereotypes provided in the DETESTS 2022 data set using an MTL approach.

While each category is predicted separately and thus treated as an individual predictive target, our model jointly learns the information of all categories, which we hypothesize favors the overall detection capabilities of the model. The outcomes of this model are afterwards analyzed both individually (*Task 2*) and by means of a $logical\_OR$ function that combines all predictions in an overall stereotype detection prediction (*Task 1*).

2. Methodology

The goal of our method is to develop a model that learns a sequence representation that is useful for all the ten stereotype categories. An overview of our architecture is shown in Figure 2.

The first step in our method is to tokenize the text fragments, and then to encode (both semantically and positionally) and process the tokens in the input using a pre-trained deep language model, which constitutes the backbone of our approach.

The second step consists of learning a joint sequence representation for the input text fragments that can later be used to jointly train all stereotype categories at the same time with MTL. The rationale behind this strategy is that each of the different stereotype categories have distinct traits that correlate with the overall stereotype detection task (*Task 1*), but at the same time are complementary.

Our MTL strategy allows the model to leverage this complementarity while exploiting the particular predictive traits of each target. In addition, MTL results especially beneficial in low-resource settings, in the case of those predictive tasks or categories for which there is very little information, since they can leverage the information provided by the remaining categories during the training process, which would be missing in a single-task setup.

In a neural-based MTL model, this is accomplished via parameter sharing, by taking the representation generated by the last layer of the pre-trained language model over one token that attends all the other tokens in the sequence. For our models, we selected the $[CLS]$ token, but theoretically any other token can be selected when using Transformed-based [5] architectures.

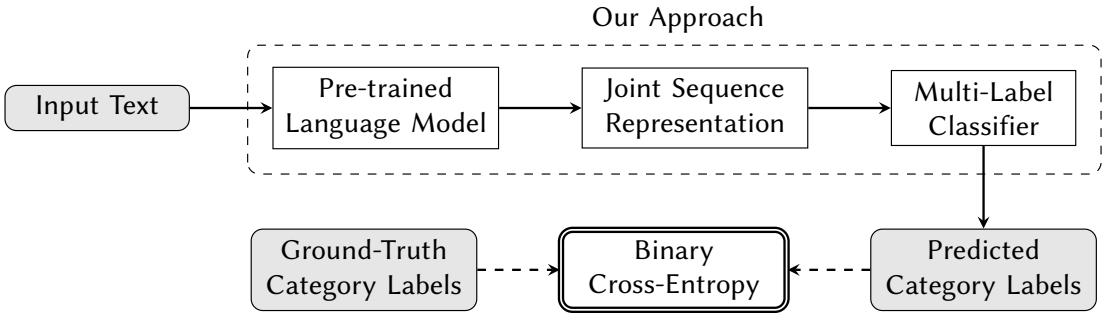

Figure 2: Overview of our approach for DETESTS 2022. First, the raw input is tokenized, encoded and processed with a pre-trained language model, such as BERT or RoBERTa. The representation on top of the [CLS] token is taken as a joint representation of the whole sequence, which is then fed into a multi-label classifier that predicts whether the instance belongs to zero, one or more of the stereotype categories in the data set. Finally, the predicted categories are compared with the ground-truth labels of the ten stereotype categories using point-wise Binary Cross-Entropy to update the parameters of the whole architecture. The outputs of the model are the predictions for *Task 2*, which can be aggregated using $logical\_OR$ to produce the predictions for *Task 1*.

The third and final step in our approach is to feed the joint sequence representation computed in the previous step into a feed-forward layer with as many output units as the number of categories to be predicted—ten outputs, in the case of DETESTS 2022—, each implementing a sigmoid activation function [18] to produce the final estimation of the probability of the sequence belonging to each of the categories. These predictions are finally compared with the category labels using point-wise Binary Cross Entropy [18], which produces an objective function that can be optimized using a standard Deep Learning training pipeline.

3. Experimental Setup

3.1. Data

The data set of the DETESTS 2022 shared task comprises fragments of comments published in response to different articles extracted from Spanish online newspapers from August 2017 to August 2020 and from June 2020 to November 2021. It consists of $5,629$ sentences, being on average 24% of them tagged as positive for at least one stereotype category. Participants were provided with 70% of the sentences and their ground-truth labels for the different stereotype categories (*Task 2*) and overall stereotype detection (*Task 1*), namely the *Train* partition, which was used to train the models. The remaining 30% of the data set, namely the *Test* partition, was later released to evaluate the trained models. Table 1 summarizes the number of positive examples per stereotype category for the *Train* partition of the data set.

Each sentence in the data set can either be tagged as negative or positive for one or more stereotype categories, which indicates that the comment exhibits stereotypical content related to: 1) 'benefits' with respect to social policies, 2) 'cultural and religious differences', 3) 'dehumanization', 4) 'economic resources', 5) 'public health', 6) 'migration control', 7) 'security', 8)

Table 1
Summary of positive instances for *Task 1* and each of the ten stereotype categories of the DETESTS 2022 shared task in the *Train* partition of the data set.

No.	Category	# Positive Instances (3817 total)
	Task 1	**871 (29.566%)**
1	Benefits	206 (5.397%)
2	Culture	189 (4.952%)
3	Dehumanization	65 (1.703%)
4	Economic	55 (1.441%)
5	Health	17 (0.445%)
6	Migration	321 (8.410%)
7	Security	255 (6.681%)
8	Suffering	63 (1.651%)
9	Xenophobia	16 (0.419%)
10	Others	67 (1.755%)

'suffering victims', 9) 'xenophobia', and 10) 'other' types of stereotypes, as listed in Table 1.

For the task of overall stereotype detection (*Task 1*), the objective was to detect the presence of at least one stereotype category in the text fragment, therefore signaling the instance as positive, while for *Task 2* the goal was to determine for which of the ten categories the comment was positive.

3.2. Experimental workflow

3.2.1. Training and validation

Our experimental workflow consisted of a **model selection** stage, where we tested different modeling and data tokenization approaches, as well as various hyper-parameter combinations, followed by a **training and validation** stage. During these two stages, we used stratified 5-fold cross-validation to estimate the performance of the models without compromising any training data and without biasing the results by only using a fixed partition of the data.

The five folds were obtained from the *Train* partition of the data set using the standard implementation from Scikit-Learn [19]. Since the ten category labels are not mutually exclusive, we stratified the folds by using the ground-truth label for *Task 1* in order to produce balanced splits of the *Train* partition. All models employed the same folds throughout the two stages of the experimental workflow, and the average results of the five validation folds we taken into account for the evaluation.

Finally, we conducted a **final evaluation** stage where we computed the results on the *Test* partition. First, we identified the three architectures for *Task 2* that performed the best in the training and validation stage, and then we conducted a final training phase using all the data available in the *Train* partition to predict the target labels for both *Task 1* and *Task 2* on the *Test* partition, for which the ground-truth labels have not been disclosed.

3.2.2. Baselines

In order to validate our approach, we compared our results with several baselines from the state of the art in NLP. The main difference between our method and the baselines is that these methods either train a different classifier for each one of the categories, or model *Task 1* directly as a binary classification task, whereas our approach jointly learns to predict all ten stereotype categories and then model *Task 1* by means of a consensus strategy consisting of a $logical\_OR$ function applied over the predictions of the ten categories.

Six out of the seven baselines are models based on traditional methods from Machine Learning: *Logistic Regression* (LR), *Random Forest* (RF) and *Support Vector Machine* (SVM). These models were trained using the default implementation provided by Scikit-Learn [19], except for the balanced loss to account for the class imbalance among the ten stereotype categories. For the input, we employed a *Bag-Of-Words* representation with TF-IDF weighting, both implemented in Scikit-Learn [19]. We tokenized using either word unigrams and bigrams or character trigrams, setting a minimum document frequency of five in both cases.

The seventh baseline is a single-task Transformer-based fine-tuning pipeline, as was described originally in Radford et al. [20], Devlin et al. [5]. For this baseline, the model architecture is very similar as ours, but each category is learned separately instead of adopting a joint learning strategy. In this case, the embeddings, the Transformer layers, the sequence representation and the top classifier are fine-tuned to a different value for each category, whereas in our proposed approach, these elements are shared across all the stereotype categories.

3.2.3. Pre-trained checkpoints

Since nowadays there is a huge variety of pre-trained language models that are publicly available online, we selected the ones we considered to be the most relevant for our tasks and adapted them to our proposed architecture. All the pre-trained language models we employed were retrieved from the Transformers library [21]. The URLs to access their respective checkpoints are provided in Table 2.

Table 2
Checkpoints of the pre-trained deep language models used in our experimental workflow. All models hereby listed were retrieved from the Transformers library [21].

Name	URL
Spanish RoBERTA	https://huggingface.co/PlanTL-GOB-ES/roberta-base-bne
Spanish BERT (BETO)	https://huggingface.co/dccuchile/bert-base-spanish-wwm-uncased
Multilingual BERT	https://huggingface.co/bert-base-multilingual-uncased

3.2.4. Hyper-parameter selection and hardware

All of our models were trained with a constant learning rate $\alpha = 10^{-5}$, without using dropout nor weight decay regularization. We trained seven trials of each model during $n = \{5, 10, 15, 17, 20, 22, 25\}$ epochs, all of them using a batch size of 2 to ensure that the

models had enough time to converge. All the models were trained using 4 CPU cores, 16 GB of RAM, and a single NVIDIA A100 GPU with 40 GB of memory. The elapsed training time for every fold during the cross-validation stage took around 3 minutes for the largest models, taking at most 15 minutes to evaluate each model's performance on the whole data set.

4. Results and Discussion

In this section, we present the results obtained using the six proposed MTL-based models as well as the results obtained using the seven baseline (*single-task* or *non-MTL*) models. We hereby report the performance of the models using four metrics traditionally used for classification tasks: $Accuracy$, $Precision$, $Recall$, and $F1 - Score$.

We do not report the results of all the models tested along our experimental workflow on the *Test* data partition, i.e., the *held-out* set, given that the ground-truth labels were not made available by the organizers. Therefore, we only report the metrics computed by the organizers upon the predictions yielded by our submitted models, namely $F1 - Score$ for *Task 1* and the ICM [22], Hierarchical F-measure [23] and Propensity F-measure [24] for *Task 2*.

For the classical baselines (*LR, RF*, and *SVM*), we show the results of the best performing model using character-based features and the best performing models using word-based features, giving a total of six baseline models. The seventh baseline is the best performing pre-trained Transformer-based deep language model Spanish BERT (BETO) [14], trained for 15 epochs using a single-task learning approach, namely $BETO_{15}$. We hereby show the results for the best-performing baseline models based on the average $F1 - Score$ for *Task 1*.

Likewise, we selected the best performing MTL-based Spanish BERT (BETO), RoBERTa, and BERT multilingual models, based on the performance on *Task 1* in terms of average $F1 - Score$ over the five validation splits of the 5-fold cross-validation process. The best-performing models found were *BETO* MTL_{17}, *RoBERTa* MTL_{10} and *BERT multilingual* MTL_{22}, where the sub-index in the names of all BERT-based approaches indicates the number of epochs for which the models were fine-tuned.

Lastly, we show the results of three additional MTL models based on their performance on *Task 2*, which were chosen according to their average $F1 - Score$ for each category. We selected them by rating all the performances attained in each category and considering the models that reached the top-three in most categories. These models, whose predictions we submitted to DETESTS 2022, are *RoBERTa* MTL_{15}, *RoBERTa* MTL_{17}, and *BETO* MTL_{20}.

4.1. *Task 1*

As mentioned before, all the models were trained and evaluated following a stratified 5-fold cross-validation approach, thus we estimated the global performance of each model for *Task 1* by averaging the metrics obtained on the validation sets of the five folds. The average $F1 - Score$, $Accuracy$, $Precision$ and $Recall$ obtained by the models are shown in Figure 3.

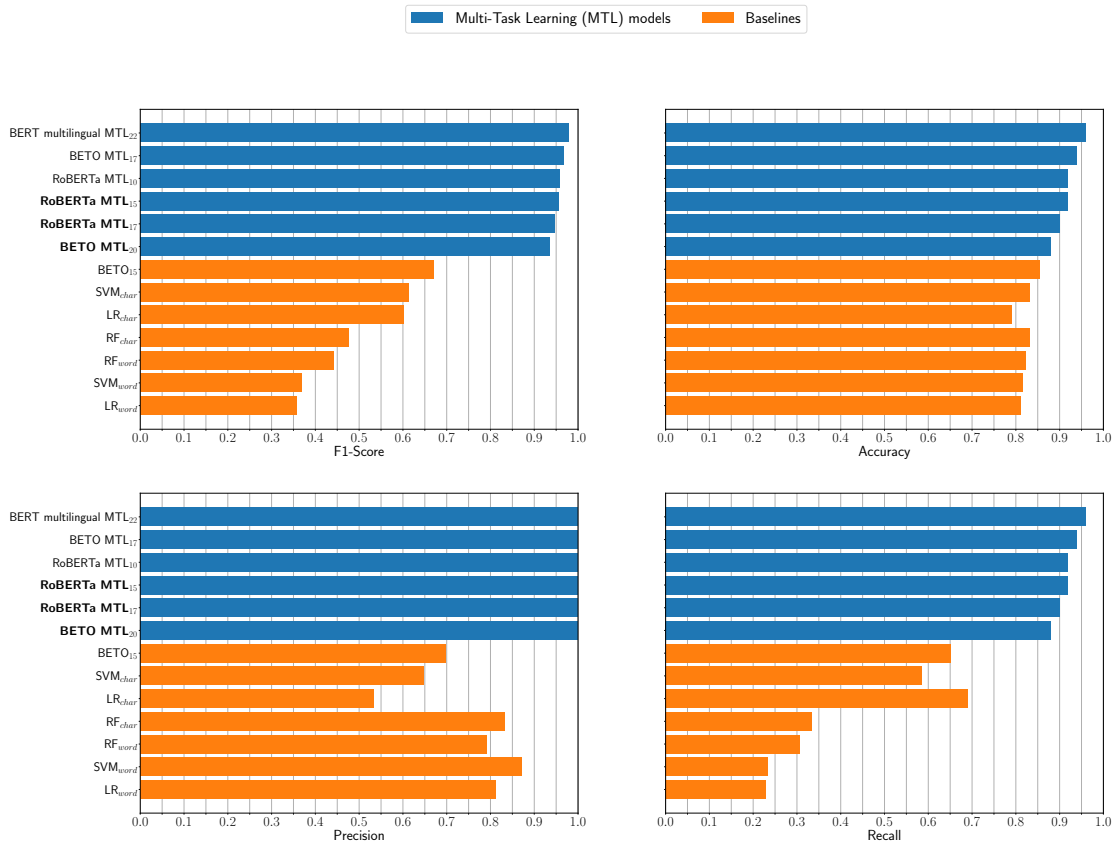

Figure 3: Overall performance on *Task 1*. Only the best model from each class is shown. The models with the *MTL* tag are the architectures using the Multi-Task Learning framework. The exact values displayed here can be found in Table 4.

4.2. *Task 2*

As explained before, all the models were trained and evaluated following a stratified 5-fold cross-validation approach, thus we estimated the global performance of each model for *Task 2* by averaging the metrics obtained on the validation sets of the same five folds as in *Task 1*. The average $F1 - Score$, $Accuracy$, $Precision$ and $Recall$ obtained by the models averaged across all categories are shown in Figure 4, while the performance of each model for *Task 2* broken down by category is shown in Figure 5.

4.3. *Final Results*

Our approach obtained ninth place out of thirty-nine teams in the first task and first place out of five teams in the second task. As mentioned previously, we submitted three models for both *Task 1* and *Task 2* of the DETESTS shared task. The official evaluation metrics used in DETESTS 2022 were ICM [22], Hierarchical F-Measure [23] and Propensity F-Measure [24]. The metrics obtained by our three models in the *Test* partition for both tasks are shown in Table 3.

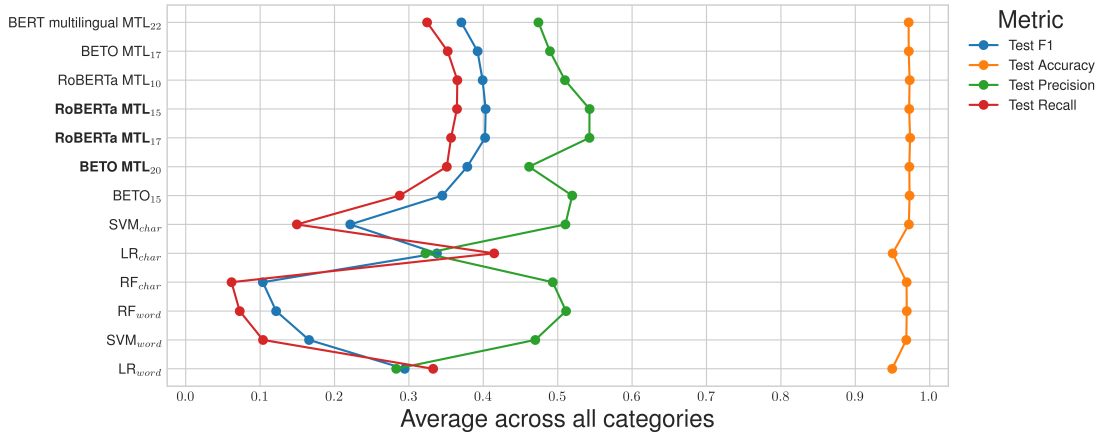

Figure 4: Overall performance on *Task 2*. Only the best model from each class is shown. The models with the *MTL* tag are the architectures using the Multi-Task Learning framework. The metrics displayed are averaged across all ten categories. The exact values displayed here can be found in Table 5.

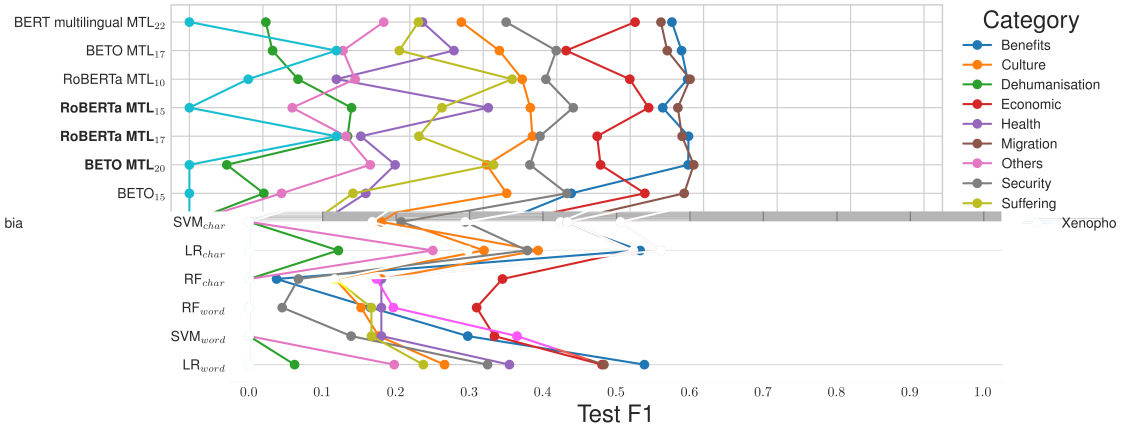

Figure 5: Performance on *Task 2* by category. Only the best model from each class is shown. The models with the *MTL* tag are the architectures using the Multi-Task Learning framework. The exact values displayed here can be found in Tables 6 and 7.

Overall, we observed that the traditional Machine Learning methods (Logistic Regression, Random Forest and Support Vector Machine) trained on the TF-IDF embeddings (for both word- and character- based n-grams) provided a very strong baseline for this task, albeit being very simple and much less computationally expensive than neural-based methods.

From the analysis of the different baselines in Figure 3, we found that all models using features based on characters outperformed the ones using features based on words. Although this result might be surprising, poor performance of models using word-based features is expected in low-resource scenarios, where there might not be enough data to learn a dense *Bag-Of-Words* representation. Instead, a character-based sequence representation yields a smaller vocabulary,

Table 3
Final metrics obtained by our method on the *Test* partition of DETESTS 2022. Participants were provided with 70% (*Train* partition) of the data set to train their models, while the remaining 30% was used to test the trained models (*Test* partition).

Model	*Task 1*	*Task 2*		
	F1-Score	ICM	Hierarchical F-Measure	Propensity F-Measure
RoBERTa MTL_{15}	0.6382	-0.2381	0.8808	0.8718
RoBERTa MTL_{17}	0.6371	-0.2380	0.8813	0.8717
BETO MTL_{20}	0.6035	-0.3759	0.8725	0.8616

which in turn allows for representations that are more dense and have less dimensions.

The experimental results support our hypothesis that MTL-based approaches are crucial for boosting the performance of models in small-data settings. Even when using large pre-trained deep language models, such as $BETO_{15}$, the performance of single-task approaches is significantly lower than that attained by MTL-based models. Although *BERT multilingual MTL* is the best model in terms of average $F1 - Score$, the difference in performance among all MTL-based models is negligible. It is note-worthy how the choice of pre-trained model influenced the results: the categories that were benefited from using RoBERTa, BETO or multilingual BERT are completely different in each case.

Additionally, from analyzing Figures 3 and 4, we can see that although *BERT multilingual* is the best performing model for *Task 1*, but on average, its performance for *Task 2* is subpar. Moreover, and as shown in Figure 4, our submitted models are the best performing ones for *Task 2*. These results highlight the fact that the best-performing method for *Task 1* is not necessarily the best for *Task 2*, but attaining a high performance in *Task 2* by means of a multi-task learning approach derives in a high performance in *Task 1*.

Lastly, it is worth mentioning that although the MTL-based methods clearly outperform the baselines in terms of $Recall$, $Precision$ and, thus, $F1 - Score$, the difference is much smaller when looking at the performance in terms of $Accuracy$, since this metric is not a good performance indicator for highly imbalanced scenarios such as this one.

5. Conclusions and Future Work

In this paper, we present a model architecture based on Multi-Task Learning for the DETESTS 2022 shared task. The main idea behind our approach is that training the model to jointly learn several related predictive tasks, in this case, the ten stereotype categories, is beneficial in low-resource scenarios.

We compared our approach with several baselines, both traditional Machine Learning methods, as well as from the current state of the art in NLP, and obtained small but consistent improvements across the majority of the stereotype categories.

For future work, we would like to explore the approach hereby presented in more diverse low-resource, multi-label classification tasks. We would also like to explore how the categories in the DETESTS 2022 shared task influence each other, and to further explore the factors that

make Multi-Task Learning architectures so successful in this kind of settings.

Other interesting future directions would be to explore if a instance weighting scheme would be beneficial given the scarcity of data in the task, or if the gradients can be aligned like in other Multi-Task Learning algorithms to further improve the performance of the models.

References

[1] A. Ariza, W. S. Schmeisser-Nieto, M. Nofre, M. Taulé, E. Amigó, B. Chulvi, P. Rosso, Overview of the DETESTS Task at IberLEF-2022: DETEction and classification of racial STereotypes in Spanish, Procesamiento del Lenguaje Natural 69 (2022).

[2] A. Vaidya, F. Mai, Y. Ning, Empirical Analysis of Multi-Task Learning for Reducing Identity Bias in Toxic Comment Detection, in: Proceedings of the International AAAI Conference on Web and Social Media, volume 14, 2020, pp. 683–693.

[3] M. Mozafari, R. Farahbakhsh, N. Crespi, Hate speech detection and racial bias mitigation in social media based on BERT model, PloS one 15 (2020) e0237861.

[4] Y. Zhang, Q. Yang, A Survey on Multi-Task Learning, IEEE Transactions on Knowledge and Data Engineering (2021).

[5] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pre-training Approach, arXiv preprint arXiv:1907.11692 (2019).

[7] H. S. Alatawi, A. M. Alhothali, K. M. Moria, Detecting white supremacist hate speech using domain specific word embedding with deep learning and BERT, IEEE Access 9 (2021) 106363–106374.

[8] K. J. Madukwe, X. Gao, B. Xue, A ga-based approach to fine-tuning bert for hate speech detection, in: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2020, pp. 2821–2828.

[9] A. Velankar, H. Patil, R. Joshi, Mono vs Multilingual BERT for Hate Speech Detection and Text Classification: A Case Study in Marathi, arXiv preprint arXiv:2204.08669 (2022).

[10] J. A. Benitez-Andrades, Á. González-Jiménez, Á. López-Brea, C. Benavides, J. Aveleira-Mata, J.-M. Alija-Pérez, M. T. García-Ordás, BERT Model-Based Approach for Detecting Racism and Xenophobia on Twitter Data, in: Research Conference on Metadata and Semantics Research, Springer, 2022, pp. 148–158.

[11] J. A. Benítez-Andrades, Á. González-Jiménez, Á. López-Brea, J. Aveleira-Mata, J.-M. Alija-Pérez, M. T. García-Ordás, Detecting racism and xenophobia using deep learning models on Twitter data: CNN, LSTM and BERT, PeerJ Computer Science 8 (2022) e906.

[12] C. Arcila-Calderón, J. J. Amores, P. Sánchez-Holgado, D. Blanco-Herrero, Using Shallow and Deep Learning to Automatically Detect Hate Motivated by Gender and Sexual Orientation on Twitter in Spanish, Multimodal Technologies and Interaction 5 (2021) 63.

[13] F. M. Plaza-del Arco, M. D. Molina-González, L. Alfonso, SINAI at IberLEF-2021 DETOXIS task: Exploring Features as Tasks in a Multi-task Learning Approach to Detecting Toxic Comments, CEUR-WS 2943 (2021).
[14] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish Pre-Trained BERT Model and Evaluation Data, in: PML4DC at ICLR 2020, 2020.
[15] P. Kapil, A. Ekbal, A deep neural network based multi-task learning approach to hate speech detection, Knowledge-Based Systems 210 (2020) 106458.
[16] A. Singh, S. Saha, M. Hasanuzzaman, K. Dey, Multitask Learning for Complaint Identification and Sentiment Analysis, Cognitive Computation 14 (2022) 212–227.
[17] M. Taulé, A. Ariza, M. Nofre, E. Amigó, P. Rosso, Overview of the DETOXIS task at IberLEF 2021: DEtection of TOXicity in comments In Spanish, Procesamiento del Lenguaje Natural 67 (2021) 209–221. URL: http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6390.
[18] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An Imperative Style, High-Performance Deep Learning Library, in: Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019, p. 1234.
[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, the Journal of machine Learning research 12 (2011) 2825–2830.
[20] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving Language Understanding by Generative Pre-training, https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf (2018).
[21] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-Art Natural Language Processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. doi:10.18653/v1/2020.emnlp-demos.6.
[22] E. Amigo, A. Delgado, Evaluating extreme hierarchical multi-label classification, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 5809–5819. URL: https://aclanthology.org/2022.acl-long.399. doi:10.18653/v1/2022.acl-long.399.
[23] E. Costa, A. Lorena, A. Carvalho, A. Freitas, A review of performance evaluation measures for hierarchical classifiers, in: Evaluation methods for machine learning II: Papers from the AAAI-2007 workshop, 2007, pp. 1–6.
[24] H. Jain, Y. Prabhu, M. Varma, Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 935–944.

A. Tables

Table 4
Overall performance on Task 1.

Model	Accuracy	Precision	Recall	F1-Score
BERT multilingual $\text{MTL}_{22-epochs}$	0.960	1.000	0.960	0.979
BERT multilingual $\text{MTL}_{17-epochs}$	0.940	1.000	0.940	0.968
BETO $\text{MTL}_{17-epochs}$	0.940	1.000	0.940	0.968
BERT multilingual $\text{MTL}_{20-epochs}$	0.920	1.000	0.920	0.958
RoBERTa $\text{MTL}_{10-epochs}$	0.920	1.000	0.920	0.958
BETO $\text{MTL}_{15-epochs}$	0.920	1.000	0.920	0.958
RoBERTa $\text{MTL}_{15-epochs}$	0.920	1.000	0.920	0.957
BETO $\text{MTL}_{25-epochs}$	0.920	1.000	0.920	0.957
RoBERTa $\text{MTL}_{25-epochs}$	0.920	1.000	0.920	0.957
$\text{BETO}_{15-epochs}$	0.854	0.670	0.697	0.651
$\text{BETO}_{1-epochs}$	0.848	0.653	0.698	0.632
$\text{BETO}_{10-epochs}$	0.854	0.652	0.714	0.602
SVM_{char}	0.832	0.647	0.586	0.614
$\text{LR}_{word+char}$	0.811	0.577	0.656	0.614
LR_{char}	0.791	0.533	0.690	0.601
$\text{SVM}_{word+char}$	0.838	0.699	0.509	0.589
RF_{char}	0.833	0.832	0.334	0.476
$\text{RF}_{word+char}$	0.824	0.783	0.317	0.450

Table 5
Overall performance on *Task 2*. The models with the *MTL* tag are the architectures based on Multi-Task Learning. The metrics displayed are averaged across all ten categories.

Model	F1-Score	Accuracy	Precision	Recall
BERT multilingual MTL_{22}	0.370	0.972	0.474	0.324
BETO MTL_{17}	0.392	0.972	0.490	0.352
RoBERTa MTL_{10}	0.399	0.973	0.510	0.365
RoBERTa $\boldsymbol{MTL_{15}}$	0.403	0.972	0.543	0.364
RoBERTa $\boldsymbol{MTL_{17}}$	0.402	0.974	0.543	0.357
BETO $\boldsymbol{MTL_{20}}$	0.378	0.973	0.461	0.351
$BETO_{15}$	0.345	0.973	0.519	0.288
SVM_{char}	0.221	0.972	0.510	0.149
LR_{char}	0.338	0.950	0.322	0.414
RF_{char}	0.104	0.969	0.493	0.061
RF_{word}	0.121	0.969	0.511	0.073
SVM_{word}	0.166	0.969	0.470	0.104
LR_{word}	0.294	0.950	0.283	0.333

Table 6
Test F1-Score on *Task 2* by category (first five categories). The models with the *MTL* tag are the architectures based on Multi-Task Learning.

Model	Benefits	Culture	Dehumanisation	Economic	Health
BERT multilingual MTL$_{22}$	0.657	0.370	0.104	0.607	0.317
BETO MTL$_{17}$	0.670	0.422	0.113	0.513	0.360
RoBERTa MTL$_{10}$	0.678	0.453	0.148	0.599	0.200
RoBERTa MTL$_{15}$	0.644	0.464	0.221	0.625	0.407
RoBERTa MTL$_{17}$	0.679	0.467	0.215	0.555	0.233
BETO MTL$_{20}$	0.679	0.405	0.051	0.560	0.280
BETO$_{15}$	0.520	0.432	0.101	0.620	0.240
SVM$_{char}$	0.424	0.207	0.000	0.433	0.180
LR$_{char}$	0.532	0.320	0.121	0.523	0.393
RF$_{char}$	0.037	0.117	0.000	0.345	0.180
RF$_{word}$	0.165	0.152	0.000	0.310	0.180
SVM$_{word}$	0.297	0.176	0.000	0.334	0.180
LR$_{word}$	0.538	0.266	0.062	0.480	0.354

Table 7
Test F1-Score on *Task 2* by category (last five categories). The models with the *MTL* tag are the architectures based on Multi-Task Learning.

Model	Migration	Others	Security	Suffering	Xenophobia
BERT multilingual MTL$_{22}$	0.642	0.264	0.431	0.312	0.000
BETO MTL$_{17}$	0.650	0.209	0.499	0.286	0.200
RoBERTa MTL$_{10}$	0.681	0.226	0.485	0.439	0.080
RoBERTa MTL$_{15}$	0.665	0.140	0.522	0.344	0.000
RoBERTa MTL$_{17}$	0.671	0.213	0.477	0.312	0.200
BETO MTL$_{20}$	0.686	0.246	0.464	0.414	0.000
BETO$_{15}$	0.674	0.125	0.514	0.223	0.000
SVM$_{char}$	0.505	0.000	0.294	0.168	0.000
LR$_{char}$	0.560	0.250	0.379	0.298	0.000
RF$_{char}$	0.173	0.000	0.067	0.116	0.000
RF$_{word}$	0.196	0.000	0.045	0.166	0.000
SVM$_{word}$	0.364	0.000	0.139	0.166	0.000
LR$_{word}$	0.483	0.197	0.324	0.237	0.000