RETUYT-InCo Submission at HUHU 2023: Detecting Humor and Prejudice through Supervised Methods

Ignacio Sastre[1,†], Alexis Baladón[1,†], Mauricio Berois[1,†], Fernanda Cánepa[1,†], Agustín Lucas[1,†], Santiago Castro[2], Santiago Góngora[1] and Luis Chiruzzo[1]

[1] *Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Uruguay*
[2] *University of Michigan – Ann Arbor, USA*

Abstract

People sometimes try to tell a joke to amuse others, but they can also hurt some other person or social group in the process, consciously or unconsciously. The HUrtful HUmour (HUHU) shared task tries to encourage the development of systems that detect and classify offensive texts and whether or not they are intended to be humorous. In this paper, we detail the participation of the RETUYT-InCo team in the HUHU shared task, where we got the first place for task 1 with an F_1 score of 0.820.

Keywords

computational humor, prejudice, Spanish, machine learning, large language models

1. Introduction

Disparagement humor is how some authors refer to the amusement through hurting some group of people. This concept is related to the phenomenon where a person attempts to make a joke by humiliating a social group based on some characteristics, such as gender, sexual orientation, appearance, beliefs, nationality, or other common backgrounds [1]. The fact of exposing those characteristics and making them look like a flaw is what facilitates laughing about others' shortcomings, what in German is called *schadenfreude*. Why *disparagement humor* works is still an object of study and different hypotheses have been put forward by several theorists. For instance, it can be a way to express hostile thoughts while trying to sound amusing, hence the listeners might not to consider it an honest and overtly dangerous opinion [1].

The HUrtful HUmour (HUHU) shared task [2] at IberLEF 2023 [3] tries to encourage the Spanish NLP community to develop models and systems that detect and classify offensive texts and whether or not they are intended to be humorous. Several previous IberLEF shared tasks have dealt with computational humor from different perspectives, such as humor detection and rating [4, 5, 6, 7], and humor analysis [8]. The analysis of the intersection between humor and offensiveness has been tackled in the past for English [9, 10]. However, this is the first time the hurtful and the prejudice dimensions and their relation with humor together with a prejudice score, have been included in one of these tasks, in particular for the Spanish language. In [8]

IberLEF 2023, September 2023, Jaén, Spain
isastre@fing.edu.uy (I. Sastre); alexis.baladon@fing.edu.uy (A. Baladón); mauricio.berois@fing.edu.uy (M. Berois); maria.canepa@fing.edu.uy (F. Cánepa); agustin.lucas@fing.edu.uy (A. Lucas); sacastro@umich.edu (S. Castro); sgongora@fing.edu.uy (S. Góngora); luischir@fing.edu.uy (L. Chiruzzo)

CEUR Workshop Proceedings (CEUR-WS.org)

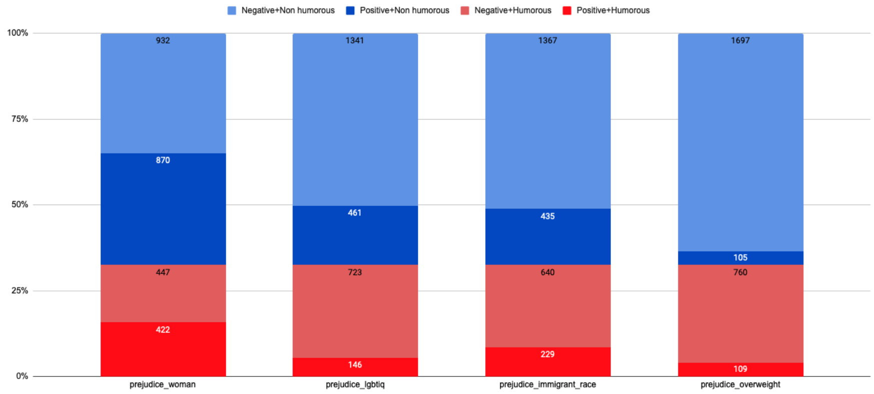

Figure 1: Balance of the categories within the dataset. From bottom to top, it shows the proportion of humorous tweets that belong and do not belong to the class, and then the same but for non-humorous tweets. The number shown in each stacked bar indicates the absolute amount of tweets represented; note that each class sums up 2671 tweets.

there was a related task about humor target detection that included categories such as women, LGBTIQ, ethnicity, or body shaming, amongst others; but it was only for the humorous tweets, and the prejudice value was not considered for that task.

In this paper, we describe the approaches followed by the RETUYT-InCo team for our participation in the HUrtful HUmour (HUHU) shared task [2]. We will describe the models and techniques used, as well as the final position we obtained for each of the HUHU tasks.

The rest of the paper is structured as follows: section 2 presents the data used for training or fine-tuning the models; section 3 describes the systems developed for our submissions; section 4 shows the results for our submissions and the positions obtained in the ranking; finally section 5 includes the conclusions of our work.

2. Dataset

The training set made available by the organizers contains 2671 tweets, all of them considered to be hurtful or conveying prejudice in some way. Each tweet is also labeled according to five dimensions: if it is considered humorous, and if it shows prejudice towards women, the LGBTIQ community, immigrants or people's race, and overweight people.

Figure 1 shows the balance of humorous/non-humorous tweets and the proportion of tweets belonging or not to each class for each dimension. For example, it can be noticed the *prejudice_overweight* category is pretty unbalanced towards the *negative* class, while the *prejudice_woman* category is more balanced in general.

Figure 2 shows a histogram of the mean prejudice values broken down by group. Note that there seems to be a difference in the distribution of the prejudice values for humorous and non-humorous tweets, with the humorous class shifted toward higher values.

The previously mentioned data provided by the shared task includes a labeled training set and an unlabeled test, but no dev data. In order to compare our different experiments internally,

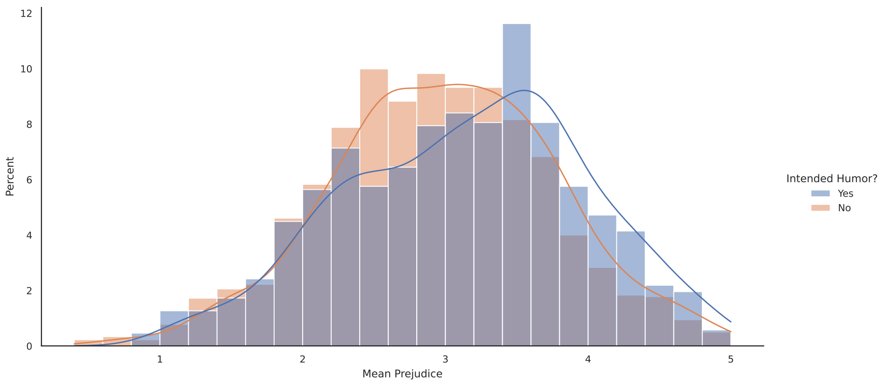

Figure 2: Histogram of the mean prejudice values across the dataset expressed as a percent of its group counts, separated by humorous and non-humorous tweets. Kernel density estimates are shown for reference.

we decided to make our own internal 90%-10% split of the training data into train and dev sets. Throughout the rest of the document, whenever we refer to the "train set" and "dev set", we will refer to our own internal splits. Our train set contains 2404 tweets, and our dev set contains 267 tweets, and we aimed to keep the partition as similarly balanced as the original set as possible, although this was difficult for the prejudice group 4 (*prejudice_overweight*), as there were too few examples of the positive class.

3. Systems description

The systems we proposed can be divided into two big groups: classic machine learning methods and fine-tuning pre-trained deep learning models. The code for these experiments can be found here: https://github.com/pln-fing-udelar/retuyt-inco-huhu-2023. We have applied these methods generally to all the tasks by only changing the part right before the output – changing the classification/regression head.

3.1. Classic Machine Learning Methods

We employed a pipeline that consists of data augmentation, preprocessing the texts, extracting features, dimensionality reduction, and using classic machine learning methods. For data augmentation, we specifically considered the data balancing issue, as shown in Figure 1. We experimented with upsampling the data by employing back-translation (from Spanish to English and then back) on 20% of the training dataset. At the preprocessing time, we considered lowercasing, removing Spanish-specific accents, deleting numbers, and removing Twitter-specific devices such as mentions, URLs, and hashtags. All these preprocessing techniques were not always applied; we considered different combinations. After this, we tokenized the texts and also experimented with applying stemming and removing stop words. For dimensionality reduction, we considered applying Principal Component Analysis (PCA). The rationale behind employing dimensionality reduction is that, in some cases, we experimented with a large number of features yet the training set size is rather small. For the feature extraction, we explored multiple methods, including Bag of Words (BoW), tf–idf, and different pre-trained language models.

We explored features from several pre-trained language models, including RoBERTuito [11] (base uncased) and multiple models within SentenceTransformers [12]. We hypothesized that RoBERTuito would have a great performance in the context of the HUHU task since it was trained on a large dataset of tweets written in Spanish. However, the pretrained model does not provide or guarantee any fixed-length sentence representation (tweet representation, in our case). Still, to our surprise, we found that using the `[CLS]` token (without any fine-tuning; even when it was unused during training) presents great performance, as the next section of this paper shows. We also believed that SentenceTransformers models would be a sufficient fit since they were trained to provide fixed-length sentence representation without any fine-tuning. The SentenceTransformers pre-trained models we experimented with are Sentence-T5 [13] (base and large), `all-MiniLM-L6-v2` (based on MiniLM [14]), GTR [15] (base, large, and `cohere-io/gtr-t5-large-1-epoch`), XLM-RoBERTa [16] (`symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli`, `paraphrase-xlm-r-multilingual-v1`), BETO-based models [17] (`hiiamsid/sentence_similarity_spanish_es`), GPT-Neo [18] variant 2.7B, BERTIN-RoBERTA [19], and a Spanish variant of RoBERTa (`Maite89/Roberta_finetuning_semantic_similarity_stsb_multi_mt`). Finally, we also tried obtaining several real-valued features from pysentimiento [20], including the hate-speech-specific features: aggressiveness (`ag`), targeted (`tr`), and general hate-speech (`hs`). Note that we did not perform preprocessing when computing features from pre-trained language models.

To learn from the data using the features, the Machine Learning methods we explored were: Support Vector Machines (SVM) [21], k-Nearest Neighbors (kNN), Gradient Boosting, Decision Trees, Random Forests, Logistic/Linear Regression, SGD-optimized linear models, MLP Neural Networks, and Naive Bayes. These methods were implemented using the scikit–learn library [22].

We want to note that we did not try all combinations of the mentioned methods with the pre-processing techniques, still, we experimented with combinations that we believe make sense. See the next section for details on the presented results.

3.2. Fine-tuning Pre-trained Deep Learning Models

Apart from leveraging frozen features coming from pre-trained language models (among other features) as input to shallow machine learning models, we experimented with fine-tuning pre-trained learning language models. Concretely, we conducted experiments with XLM-Roberta [16], RoBERTuito [11], and BERT [30, 17]. When using BERT, we froze the backbone. For the other two models, all the parameters were trainable. In all cases, we tuned a new head on the training set. For XLM-RoBERTa, we average the backbone outputs to pass them to the new classifier head. In the case of BERT and RoBERTuito, we take the `[CLS]` token output. We explored using different probabilities of dropout as well as using one or two linear layers separated by ReLU activations. We trained the models using the Adam [31] optimizer with

Model	Humor F_1
RoBERTuito Embedding PCA 100 kNN	0.7349
RoBERTuito Embedding PCA 200 kNN	0.7317
RoBERTuito Embedding kNN	0.7317
RoBERTuito Embedding PCA 100 MLP	0.7416
RoBERTuito Embedding PCA 200 MLP	0.7473
SentenceTransformers XLM-RoBERTa Linear	0.5952
SentenceTransformers BERTIN-RoBERTa Linear	0.6383
SentenceTransformers SimilaritySpanish Linear	0.6590
SentenceTransformers T5 kNN	**0.7636**
RoBERTuito fine-tuning	**0.7935**
BERT Multilingual fine-tuning	0.6627

Table 1: Results for task 1 over our internal development set. Bold numbers are the highest scores, and the underlined numbers are the models selected for submission.

different learning rates.

For RoBERTuito, which has the same architecture as RoBERTa, we introduced a classification head consisting of 592,130 additional parameters. This brings the total number of trainable parameters during the fine-tuning step to 108,196,608. We utilized the tokenizer provided by the pretrained model for consistency. During training, we employed a learning rate of 3×10^{-5} with the Adam optimizer, and the training was conducted for two epochs.

3.3. Task-Specific Details

The methods carried out for each task differ mostly in the last part of the modeling. For Task 1, we employed softmax for the classification and used the cross-entropy loss. For Task 2a, we treated the problem as a multi-label classification and thus computed probabilities for each class separately, dividing it into four binary classification tasks (and then proceeded similarly to Task 1). For both of these tasks, we also consider kNN applied only to the centroid as opposed to the whole training set. For each class, a positive and negative centroid was computed to then compare new instances. For Task 2b, we regressed the output variable and used the mean squared error as the loss.

4. Submissions and Results

Tables 1 to 3 show the results for the three subtasks over our internal development split. We show a summary of the most important approaches we experimented with. Five researchers participated in doing the experiments, so when choosing which models to send as submissions we considered two aspects: firstly, which models yielded the highest performance over the dev split; and secondly, trying to submit results for models created by different researchers in order to maximize variability.

Model	Group 1	Group 2	Group 3	Group 4	Mean
tf–idf GradientBoosting	0.8837	0.7525	0.9231	**0.9143**	0.8684
BoW RandomForest	0.8750	0.7800	0.9077	**0.8824**	0.8613
SentenceTransformers XLM-RoBERTa Linear	0.8571	0.6885	0.8480	0.5263	0.7300
SentenceTransformers BERTIN-RoBERTa Linear	0.8669	0.7154	0.9268	0.7059	0.8038
SentenceTransformers SimilaritySpanish Linear	0.8913	0.7818	0.9375	0.6471	0.8144
SentenceTransformers T5 kNN	**0.9333**	**0.8966**	**0.9920**	0.8235	**0.9114**
RoBERTuito fine-tuning	**0.9111**	**0.9000**	**0.9760**	**0.8824**	**0.9174**
BERT Multilingual Fine-tuning	0.8881	0.8673	0.9688	0.8485	0.8931

Table 2: Results for task 2a over our internal development set, the columns show the F_1 score for detecting prejudice over one of the four groups, and the mean F_1 score. Bold numbers are the highest scores, and the underlined numbers are the models selected for submission.

Model	Prejudice RMSE
pysentimiento HateSpeech SGD	0.8168
pysentimiento HateSpeech Linear Model	0.8206
RoBERTuito Embedding PCA 100 SVR	0.7345
RoBERTuito Embedding PCA 200 SVR	**0.7299**
BoW BayesianRidge	0.7371
SentenceTransformers GPT-Neo XGBRegressor	0.7440
SentenceTransformers T5 kNN	**0.6989**
RoBERTuito fine-tuning	0.7619
BERT Multilingual fine-tuning	0.8535

Table 3: Results for task 2b over our internal development set. Bold numbers are the highest scores, and the underlined numbers are the models selected for submission.

Table 4 shows the results of our two submissions over the test set, including which model was used to create the results for each subtask. Our models performed very well across all tasks. In particular, we got the first position in task 1 with the RoBERTuito fine-tuning model, and positions 5 and 6 in task 2 using a combination of several methods.

5. Conclusions

We presented the submissions made by the RETUYT-InCo team for the HUHU shared task presented at IberLEF 2023. Our experiments include a variety of classical and neural machine learning models, trained with diverse features (from BoW to sentence embedding features), as well as some fine-tuning from pre-trained LLM experiments. The models performed well over the test set, obtaining the first place for Task 1, with an F_1 score of 0.820, and a good ranking in general for the other tasks.

In future work, we want to explore the possibility of adding more information to the dataset

	Submission 1			Submission 2		
Task	Model	Score	Rank	Model	Score	Rank
1	RoBERTuito fine-tuning	0.8201	1 / 58	RoBERTuito Embedding PCA 200 MLP	0.7520	17 / 58
2a-g1	SentenceEmbedding T5 kNN	0.8529		SentenceEmbedding SimilaritySpanish Linear	0.8353	
2a-g2	RoBERTuito fine-tuning	0.7955		BERT Multilingual fine-tuning	0.7238	
2a-g3	SentenceEmbedding T5 kNN	0.5509		BERT Multilingual fine-tuning	0.6957	
2a-g4	RoBERTuito fine-tuning	0.8932		TFIDF GradientBoosting	0.7879	
2a-mean		0.7731	5 / 49		0.7607	6 / 49
2b	SentenceEmbedding T5 kNN	0.9254	18 / 48	RoBERTuito Embedding PCA 200 SVR	0.9238	17 / 48

Table 4: Results for all the tasks over the official test set for our two submissions. We show the score obtained for each subtask and each prejudice group, and the rank obtained over all the competition results for tasks 1, 2a, and 2b.

to enrich the models. For example, we could add more tweets from the HAHA 2021 dataset [8], which includes information on humor target, a category that is closely related to the prejudice groups used in the HUHU shared task.

References

[1] M. A. Ferguson, T. E. Ford, Disparagement humor: A theoretical and empirical review of psychoanalytic, superiority, and social identity theories, HUMOR: International Journal of Humor Research 21 (2008) 283–312. URL: https://doi.org/10.1515/HUMOR.2008.014.

[2] R. Labadie-Tamayo, B. Chulvi, P. Rosso, Everybody Hurts, Sometimes. Overview of HUrtful HUmour at IberLEF 2023: Detection of Humour Spreading Prejudice in Twitter, in: Procesamiento del Lenguaje Natural (SEPLN), 2023.

[3] S. M. Jiménez-Zafra, F. Rangel, M. Montes-y Gómez, Overview of IberLEF 2023: Natural Language Processing Challenges for Spanish and other Iberian Languages, Procesamiento del Lenguaje Natural 71 (2023).

[4] S. Castro, L. Chiruzzo, A. Rosá, Overview of the HAHA task: Humor analysis based on human annotation at IberEval 2018, in: IberEval@SEPLN, 2018, pp. 187–194. URL: https://ceur-ws.org/Vol-2150/overview-HAHA.pdf.

[5] S. Castro, L. Chiruzzo, A. Rosá, D. Garat, G. Moncecchi, A crowd-annotated Spanish corpus for humor analysis, 2017. URL: https://arxiv.org/abs/1710.00477.

[6] L. Chiruzzo, S. Castro, A. Rosá, HAHA 2019 dataset: A corpus for humor analysis in Spanish, in: Proceedings of the Twelfth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2020, pp. 5106–5112. URL: https://aclanthology.org/2020.lrec-1.628/.

[7] L. Chiruzzo, S. Castro, M. Etcheverry, D. Garat, J. J. Prada, A. Rosá, Overview of HAHA at

IberLEF 2019: Humor analysis based on human annotation, in: IberLEF@SEPLN, 2019, pp. 132–144. URL: https://ceur-ws.org/Vol-2421/HAHA_overview.pdf.

[8] L. Chiruzzo, S. Castro, S. Góngora, A. Rosá, J. Meaney, R. Mihalcea, Overview of HAHA at IberLEF 2021: Detecting, rating and analyzing humor in Spanish, Procesamiento del Lenguaje Natural 67 (2021) 257–268. URL: http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6394.

[9] J. Meaney, S. R. Wilson, L. Chiruzzo, W. Magdy, Don't take it personally: Analyzing gender and age differences in ratings of online humor, in: Social Informatics: 13th International Conference, SocInfo 2022, Glasgow, UK, October 19–21, 2022, Proceedings, Springer, 2022, pp. 20–33. URL: https://link.springer.com/chapter/10.1007/978-3-031-19097-1_2.

[10] J. Meaney, S. Wilson, L. Chiruzzo, A. Lopez, W. Magdy, SemEval 2021 task 7: Hahackathon, detecting and rating humor and offense, in: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), 2021, pp. 105–119. URL: https://aclanthology.org/2021.semeval-1.9/.

[11] J. M. Pérez, D. A. Furman, L. Alonso Alemany, F. M. Luque, Robertuito: a pre-trained language model for social media text in spanish, in: Proceedings of the Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 7235–7243. URL: https://aclanthology.org/2022.lrec-1.785.

[12] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. URL: https://aclanthology.org/D19-1410. doi:10.18653/v1/D19-1410.

[13] J. Ni, G. Hernandez Abrego, N. Constant, J. Ma, K. Hall, D. Cer, Y. Yang, Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models, in: Findings of the Association for Computational Linguistics: ACL 2022, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 1864–1874. URL: https://aclanthology.org/2022.findings-acl.146. doi:10.18653/v1/2022.findings-acl.146.

[14] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, M. Zhou, MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 5776–5788. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[15] J. Ni, C. Qu, J. Lu, Z. Dai, G. H. Ábrego, J. Ma, V. Y. Zhao, Y. Luan, K. B. Hall, M.-W. Chang, et al., Large dual encoders are generalizable retrievers, arXiv preprint arXiv:2112.07899 (2021). URL: https://arxiv.org/abs/2112.07899.

[16] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 8440–8451. URL: https://aclanthology.org/2020.acl-main.747. doi:10.18653/v1/2020.acl-main.747.

[17] 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. URL: https://pml4dc.github.

io/iclr2020/papers/PML4DC2020_10.pdf.

[18] S. Black, L. Gao, P. Wang, C. Leahy, S. Biderman, GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, 2021. URL: https://doi.org/10.5281/zenodo.5297715. doi:10.5281/zenodo.5297715, If you use this software, please cite it using these metadata.

[19] J. De la Rosa, E. G. Ponferrada, M. Romero, P. Villegas, P. G. de Prado Salas, M. Grandury, BERTIN: Efficient pre-training of a spanish language model using perplexity sampling, Procesamiento del Lenguaje Natural 68 (2022) 13–23. URL: http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6403.

[20] J. M. Pérez, J. C. Giudici, F. Luque, pysentimiento: A Python Toolkit for Sentiment Analysis and SocialNLP tasks, 2021. URL: https://arxiv.org/abs/2106.09462. arXiv:2106.09462.

[21] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (1995) 273–297. URL: https://link.springer.com/article/10.1007/BF00994018. doi:10.1007/BF00994018.

[22] 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, Journal of Machine Learning Research 12 (2011) 2825–2830. URL: https://scikit-learn.org/.

[23] J. D. Hunter, Matplotlib: A 2D graphics environment, Computing in science & engineering 9 (2007) 90–95. URL: https://matplotlib.org/.

[24] 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. URL: https://aclanthology.org/2020.emnlp-demos.6. doi:10.18653/v1/2020.emnlp-demos.6.

[25] 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: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[26] C. R. Harris, K. J. Millman, S. J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, et al., Array programming with NumPy, Nature 585 (2020) 357–362. URL: https://numpy.org/.

[27] S. Bird, E. Klein, E. Loper, Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit, O'Reilly Media, Inc., 2009. URL: https://www.nltk.org/.

[28] M. Honnibal, I. Montani, S. Van Landeghem, A. Boyd, spaCy: Industrial-strength Natural Language Processing in Python, 2020. URL: https://spacy.io/. doi:10.5281/zenodo.1212303.

[29] The pandas development team, pandas-dev/pandas: Pandas, 2023. URL: https://github.com/pandas-dev/pandas.

[30] 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. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[31] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations (ICLR), 2014. URL: https://arxiv.org/abs/1412.6980.