BERTicelli at HaSpeeDe 3: Fine-tuning and Cross-validating Large Language Models for Hate Speech Detection

Leonardo Grotti[1,2], Patrick Quick[1]

[1]Universiteit Antwerpen, Faculty of Arts, Prinsstraat 13, B-2000, Antwerp
[2]CLiPS Research Center, University of Antwerp, Belgium

Abstract
The present paper describes the results from the experiments carried out for the HaSpeeDe 3 shared task, an Italian-language Hate Speech (HS) detection task, at EVALITA 2023. Two BERT-based language models were selected: UmBERTo (cased) and Italian BERT (cased). For the Textual task, the models were fine-tuned and cross-validated across 5 folds. For the Contextual task, we adopted an ensemble approach: the additional features were added to the fine-tuned models through the GradientBoosterClassifier algorithm. The models perform better than the baselines (DummyClassifier and LogisticRegression) and above the average performance of participants in the shared task. While the addition of contextual features did not improve the performance of UmBERTo, it significantly bettered the results obtained with Italian BERT.

Keywords
Hate Speech detection, Italian language, BERT-based language models, Fine-tuning, Contextual features

1. Introduction

The escalating issue of toxic language has been amplified by the rapid growth in social media usage over the past decade [1]. Platforms such as Facebook and Twitter have transformed the way individuals interact, making it faster and often anonymous, thereby creating an ideal environment for the propagation of harmful content [2]. Furthermore, previous studies have shown that this content can be targeted at and posted by both individuals and groups, inciting and driving violent acts in the offline world [2, 3, 4].

As such, countering the phenomenon of toxic language has garnered significant attention from legal authorities, social media platforms, and companies [5]. Platforms like Facebook, Twitter, YouTube, and other websites have taken measures to combat toxic language by implementing bans. However, research has pointed out the limitations of companies' control systems and their heavy reliance on user reports to identify problematic comments or posts [6]. The manual filtering of messages containing toxic language has proven to be not only time-consuming but also detrimental to human annotators [7]. Additionally, studies have revealed that human-labeled data can be influenced by individual annotators' biases [8].

Such interest was reflected in the field of Natural Language Processing (NLP), which has witnessed a surge in interest and popularity in automatic toxic language detection [8]. Researchers aim to develop models that can alleviate the harm caused by online HS [9]. Automating the detection process not only overcomes the challenges of manual filtering but also enables efficient analysis of large volumes of data.

As a reminder, we here use the terms HS as an umbrella term and do not distinguish between its subcategories on a theoretical level. For a more extensive discussion of HS hierarchies and definitions, refer to Zampieri [7] and Caselli et al. [10]. It is worth noting that scholars often do not agree on what constitutes HS and how it differs from, e.g., offensive or aggressive language [11].

2. Related Work

As we have mentioned, the growing interest in addressing toxic language is evident in the numerous tasks dedicated to its detection and its various subcategories. These include Aggression Identification [12], Offensive Language Identification [7], and HS detection in Italian Facebook and Twitter messages [13], among others. Over time, the quantity and quality of available models for toxic language detection have significantly increased. Markov, Gevers, and Daelemans (2021) note that the advent of transformer-based pre-trained language models, coupled with the abundance of user-generated content on social media, has greatly improved detection accuracy.

Despite the overall improvement of the models, a series of challenges remain. For instance, it has been shown how the lack of data in languages other than English [14] has exacerbated already existing issues, such as the high occurrence of code words and misspellings in HS text

EVALITA 2023: Final Workshop of the 8th evaluation campaign, September 08–09, 2021, Parma, IT
lgrotti@uantwerpen.be (L. Grotti);
patrick.quick@student.uantwerpen.be (P. Quick)
https://github.com/corvusMidnight (L. Grotti);
https://github.com/patrickquick (P. Quick)
0000-0001-7914-3191 (L. Grotti)

[9]. Furthermore, the often-low agreement between human annotators has also been reported to hinder models' performance [15]. Finally, the biggest challenge to date remains the generalisability of models for HS classification: i.e., a model's ability to perform in cross-domain (i.e., coming from different domains, such as Facebook and Twitter) and cross-genre (i.e., text belonging to different genres, e.g., social media posts and journal articles) tasks [15, 8].

To deal with these issues, scholars have adopted different approaches: machine learning algorithms have been improved through the addition of linguistic features [16, 17], word lists [18], and syntactic features [19]. In recent years, researchers have also experimented with complex ensembles of both neural [17] and non-neural [1] models.

In the latter case, they [17] leveraged the large amount of data on which large language models are trained and the increased availability of sentence and sub-word embeddings. In short, large language models possess numerical representations for both smaller (sub-embeddings) and larger (sentence-embeddings) language units. This enables them to capture not only misspellings and rare linguistic forms but also the overall meaning of sentences. Furthermore, due to their pretraining process, these models already have an understanding of language structure, enhancing their performance in capturing the nuances of a text [9, 15][1].

3. Task and Method

3.1. Data

Data set	¬HS Tweets	HS Tweets
Development	3456	2144
Test (in)	700	700
Test (out)	2513	487

Table 1
Count of Hate Speech labels in task data.

The task organizers [20] provided development data consisting of 5,600 Italian-language tweets from the Policycorpus XL corpus, a manually-annotated HS corpus [21]. The testing data consists of one subset of in-domain data and one subset of out-of-domain data. The in-domain data consists of 1,400 Italian-language tweets from Policycorpus XL, and the out-of-domain data consists of 3,000 Italian-language tweets from the Italian subset of the ReligiousHate corpus, a manually-annotated religious HS corpus [22].

[1]For an extensive explanation of how these factors have improved the performance of HS detection systems, see Yin and Zubiaga 2021

The development data has been anonymized, with a tweet's identifying information and the user's identifying information both mapped to pseudo-random integers. Placeholders have replaced all instances of URLs, and hashtags have been preserved. The data is marginally imbalanced, with a lesser presence of tweets annotated as HS. The accompanying contextual information is comprised of attributes of Tweet objects and User objects [23].

The structure and anonymization of both the in-domain and out-of-domain testing data follows that of the development data. Upon reconciliation with the gold data, we can see that the in-domain subset is balanced, whereas the out-of-domain subset is significantly unbalanced.

3.2. Task

The task proposal for HaSpeeDe 3 focuses on Italian-language HS detection on Twitter and consists of two tracks with two sub-tasks each. Both Task A and B are binary classification problems to determine whether a tweet contains HS or not. The two sub-tasks in Task A are: (i) Textual, where participants can only use the provided textual content of the tweets from Policycorpus XL for development, and (ii) Contextual, where participants can employ the textual content of the tweets along with the accompanying contextual information.

Task B deals with test data from different domains. The evaluation includes two sub-tasks: XPoliticalHate, where the test set consists of tweets from Policycorpus XL, and XReligiousHate, where the task focuses on recognizing religious hate using tweets from the ReligiousHate corpus. Participants are allowed to use external data, including from other hate domains, and are not restricted to the textual and contextual data provided in Policycorpus XL for development.

The final results in the HaSpeeDe 3 task are evaluated and ranked by computing the F1-score over each class to arrive at an averaged F1-score.

3.3. Models

Fine-tuning, a common technique in NLP, is a form of transfer learning that involves training a pre-trained model on new data to adapt it for a specific downstream task [24]. As mentioned in Section 2, there are notable benefits in fine-tuning models when it comes to HS detection tasks. Furthermore, this approach has been widely used in Italian HS detection, see, e.g., Eric et al. (2020), Tamburini (2020), Nozza et al. (2022). Thus, we fine-tune two large pre-trained language models:

- **UmBERTo-commoncrawl-cased** (Run 1) is a RoBERTa-based model using the OSCAR (Open

Super-large Crawled ALMAnaCH coRpus) Italian large corpus. The model is used for both Named Entity Recognition (NER) and Part Of Speech (POS) tagging and reached excellent performance on different datasets.

- **bert-base-italian-cased** (Run 2) is a BERT-based model which was trained on two million tokens and over 13GB of data. The model was pre-trained on a combination of data which includes the OPUS corpus as well as a Wikipedia dump. Note that for ease of readability, we will now refer to this model as BERT-ita.

For the Textual tasks (both in- and out-of-domain) our experimental setup consists of three stages: To begin with, we apply two basic preprocessing steps, which consist of substituting the pseudo-random user identifiers (e.g., '@12020569') with '@USER' and removing the hash symbol (i.e., '#') from hashtags. Such steps are applied to avoid excessively long tweets[2] and remove unnecessary noise.

Then, both models are fine-tuned and cross-validated across five epochs using PyTorch Trainer and the Transformers library. The development data is shuffled and divided into five folds. For each fold, the models are fine-tuned on 80% of the development data and evaluated on the remaining 20% across 10 epochs with an EarlyStopping patience of 3. We employ cross-validation to ensure that the obtained results are not dependent on a particular data split but rather generalize well across multiple folds.

During this stage, we also tune the learning rate ($1^{e\text{-}3}$, $2^{e\text{-}5}$, and $5^{e\text{-}05}$)[3]. We do not tune batch size as the test data was not available at this stage and increasing batch size may have improved development set performance but worsened generalizability on unseen data (see He et al., 2019). Once the stability of the results has been established through cross-validation, the models are fine-tuned on 85% of the training data[4] (after shuffling) and the resulting model is saved and used to output predictions on both test sets.

For the Contextual task, additional features are incorporated into the model using GradientBoostingClassifier. This ensemble algorithm sequentially trains weak models, resulting in a strong model that is a weighted combination of the weak models. Unlike other algorithms, GradientBoostingClassifier employs decision trees as weak learners and is optimized through gradient descent. To do so, the output labels for both BERT-based models together with the additional features are used as input features for GradientBoostingClassifier.

To further assess the performance of our models, we build four baselines: one LogisticRegression and one DummyClassifier for each task. For Textual and Contextual tasks, the models were trained on the textual data and evaluated on the in-domain test data. For the cross-domain task, they were trained on the same textual data but evaluated on the out-of-domain test set instead. It is worth mentioning that no additional data was used at any stage of our experiments.

4. Results and Discussion

In this section, we describe the results obtained for HaSpeeDe 2023. For each model, we report precision, recall, and F1 score (for both classes). We submitted results for every sub-task except for Task B's XpoliticalHate sub-task. All results are compared with the respective baselines.

4.1. Baselines

As a reference point, Table 2 first presents the baseline results for both in- and out-of-domain tasks for each class. The DummyClassifier performs slightly above random chance for the in-domain task, with an average F1 score of 0.52. However, the out-of-domain results are—as expected—poorer, reaching an average F1 score of 0.42.

LogisticRegression, on the other hand, achieves competitive results, with average F1 scores of 0.86 for in-domain data and 0.52 for out-of-domain data. However, upon further inspection, we can observe how LogisticRegression is fairly limited. With a precision of 0.80 for the non-hate speech (¬HS) class, the model exhibits a relatively high rate of false positives. Additionally, the recall of 0.96 for ¬HS implies a high rate of true positive instances captured but at the expense of potentially overlooking some true negatives. These results suggest that the model may be overly biased towards predicting instances as ¬HS, potentially missing some actual HS instances. Similarly, while a high precision of 0.953 is achieved for the HS class, the model showcases a fairly low recall of 0.75. In turn, this pattern implies the model's inability to identify a significant portion of actual HS instances, resulting in false negatives.

It is worth mentioning that the high performance of LogisticRegression in the in-domain task is likely related to the balanced nature of the data (700 HS v. 700 ¬HS). When out-of-domain, unbalanced test data (see Table 1) is used, performance drastically drops.

[2] The presence of multiple user tags in some of the tweets caused a mismatch in Tensor size and consequently a RuntimeError.

[3] These learning rates are found in Nozza et al. (2022), HuggingFace's fine-tuning guide, and in the standard training parameters, respectively.

[4] Such a configuration is selected to mirror the task's original train-test split of 5600-1400, see Celli et al. (2021).

	Class	Precision	Recall	F1
		In-domain		
Dummy	¬HS	0.518	0.504	0.511
	HS	0.517	0.530	0.523
LogReg	¬HS	0.800	0.963	0.874
	HS	0.953	0.759	0.845
		Out-of-domain		
Dummy	¬HS	0.827	0.485	0.612
	HS	0.152	0.474	0.230
LogReg	¬HS	0.844	0.926	0.883
	HS	0.237	0.119	0.158

Table 2
Results for baseline models for in-domain data and out-of-domain data.

4.2. Task A

Our models achieve competitive results in both Task A's sub-tasks[5], as shown in Table 3 (Textual) and Table 4 (Contextual) for each class. Starting from the former, both models perform above both baselines. However, there seems to be a substantial difference between UmBERTo's (Run 1) and BERT-ita's (Run 2) performance: while the first reaches an F1 average of 0.89, the second reaches 0.86, with a difference between the scores of over .03. Indeed, even if the second run's results are close to the LogisticRegression baseline, the model's predictions (i.e., precision and recall) are more balanced across the two classes. Thus, the F1 for the HS class is higher for BERT-ita compared to the baseline.

The reason for the discrepancy between the two models' performance is likely related to the size of the pre-training data: UmBERTo was trained on over 70GB (against the 13GB of BERT-ita). As such, the model likely has more sub-embeddings and sentence-embeddings available, which in turn allows for better results.

For the Contextual sub-task (Table 4), we included a set of extra features (i.e., 'anonymized description', 'retweet count', 'favorite count', 'is reply', 'is retweet', 'is quote', 'statuses count', 'followers count', and 'friends count') to the output labels through GradientBoostingClassifier. Both models once again reach competitive results. While the first run's results are not affected by the inclusion of contextual features, BERT-ita (Run 2) significantly benefits from their addition. The model performs on the same level as UmBERTo, with an F1 of 0.902 for ¬HS and 0.892 for HS. The inclusion of contextual information during the training stage likely enables BERT-ita to capture more diverse linguistic patterns and generalize better to the classification task.

[5]Note that the overall F1 average for each model and sub-task can be found in Table 7 below.

	Class	Precision	Recall	F1
Run 1	¬HS	0.861	0.949	0.903
	HS	0.943	0.847	0.892
Run 2	¬HS	0.824	0.930	0.874
	HS	0.920	0.801	0.856

Table 3
Results for Task A Textual sub-task.

	Class	Precision	Recall	F1
Run 1	¬HS	0.861	0.949	0.903
	HS	0.943	0.847	0.892
Run 2	¬HS	0.860	0.949	0.902
	HS	0.943	0.846	0.892

Table 4
Results for Task A Contextual sub-task and Task B in-domain sub-task.

4.3. Task B

Though we did not formally submit results to Task B's sub-task XPoliticalHate, we met the requirements of the task by submitting results for the Contextual sub-task of Task A, for which the same test data was used. We will thus report results for both sub-tasks of Task B, referring to Table 4 for the results of Task B's sub-task XPoliticalHate.

Our model performs competitively in the XPoliticalHate sub-task, which made use of in-domain test data, while our model for the sub-task XReligiousHate performed poorly in the context of out-of-domain test data. We made no consideration regarding the XPoliticalHate sub-task, as we did not take any additional steps.

The models' performance on out-of-domain data (Table 5) is much lower than the average F1 score (0.57) but still higher than the baseline (0.52). Such low scores may relate to the imbalance between the two classes in the test data and to limitations in transfer learning. As noted by Ada et al. (2019), performance on the source task may not reflect performance on the target task. Also, the model may overfit on the data on which it was fine-tuned [30].

	Class	Precision	Recall	F1
Run 1	¬HS	0.849	0.950	0.897
	HS	0.330	0.127	0.184
Run 2	¬HS	0.848	0.942	0.893
	HS	0.306	0.131	0.184

Table 5
Results for Task B out-of-domain sub-task.

Overall, our models have consistently outperformed the baselines, demonstrating significant improvements across the board. However, it is worth noting that Table 6 reveals that some runs were below the competition's averages. In particular, Run 2 in Task A (Textual) and

Task B (XReligiousHate) failed to meet our expectations, as discussed in detail in Sections 4.2 and 4.3. These underperforming results can be attributed to the previously highlighted factors.

Task	Sub-task	Model	F1 avg
A	Textual	Run 1	0.89759
		Run 2	0.86516
		Avg	0.88263
	Contextual	Run 1	0.89759
		Run 2	0.89687
		Avg	0.88616
B	XPoliticalHate	Run 1	0.89759
		Run 2	0.89687
		Avg	0.88866
	XReligiousHate	Run 1	0.54011
		Run 2	0.53841
		Avg	0.57439

Table 6
F1 averages for our models and the average for all models submitted to the task.

5. Conclusion

In this paper, we introduced two fine-tuning techniques to detect Italian-language HS in Twitter's posts and replies. We were asked to address the issue in two different tasks with two sub-tasks each. Two models were fine-tuned and 5-cross-validated: UmBERTo and BERT-ita. Task A was comprised of a Textual and a Contextual sub-task: here, UmBERTo performed competitively in both sub-tasks, reaching above the baseline and competition average. However, the model did not benefit from the addition of contextual features. BERT-ita, on the other hand, performed above the baselines but significantly lower than the task average. In contrast to UmBERTo, BERT-ita's results improved significantly, reaching the first model's performance.

For Task B, we did not submit any results for the XPoliticalHate sub-task. As such, the results obtained for Task A (Contextual) were assumed to be valid for this sub-task given the test data was the same. Finally, both our models performed well below the competition average for the out-of-domain task.

Future work should look at the potential benefits of including additional training data for the out-of-domain task. Also, the addition of contextual features could be tested in combination with different language models.

References

[1] M. K. Aljero, N. Dimililer, A novel stacked ensemble for hate speech recognition, Applied Sciences 11 (2021) 1–15. URL: http://dx.doi.org/10.3390/app112411684. doi:10.3390/app112411684.

[2] F. Del Vigna, A. Cimino, F. Dell'Orletta, M. Petrocchi, M. Tesconi, Hate Me, Hate Me Not: Hate Speech Detection on Facebook., ITASEC (2017) 86–95.

[3] A. A. Siege, Online Hate Speech, Cambridge University Press, 2020. URL: https://doi.org/10.1017/9781108890960[.

[4] K. P. De Maiti, D. Fišer, N. Ljubešić, Nonstandard linguistic features of Slovene socially unacceptable discourse on Facebook, Znanstvena založba Filozofske fakultete (2020).

[5] M. Sanguinetti, F. Poletto, C. Bosco, V. Patti, M. Stranisci, An Italian Twitter Corpus of Hate Speech against Immigrants, Language Resources and Evaluation (2018) 1–8.

[6] M. Sanguinetti, G. Comandini, E. D. Nuovo, S. Frenda, M. Stranisci, C. Bosco, T. Caselli, V. Patti, I. Russo, HaSpeeDe 2 @ EVALITA2020: Overview of the EVALITA 2020 Hate Speech Detection Task, EVALITA Evaluation of NLP and Speech Tools for Italian - December 17th, 2020 (2020) 93–101. URL: http://dx.doi.org/10.4000/books.aaccademia.6897. doi:10.4000/books.aaccademia.6897.

[7] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, R. Kumar, SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval), Proceedings of the 13th International Workshop on Semantic Evaluation (2019). URL: http://dx.doi.org/10.18653/v1/s19-2010. doi:10.18653/v1/s19-2010.

[8] I. Markov, W. Daelemans, Improving Cross-Domain Hate Speech Detection by Reducing the False Positive Rate, Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda (2021). URL: http://dx.doi.org/10.18653/v1/2021.nlp4if-1.3. doi:10.18653/v1/2021.nlp4if-1.3.

[9] W. Yin, A. Zubiaga, Towards generalisable hate speech detection: a review on obstacles and solutions, PeerJ Computer Science 7 (2021) e598. URL: http://dx.doi.org/10.7717/peerj-cs.598. doi:10.7717/peerj-cs.598.

[10] T. Caselli, V. Basile, J. Mitrović, M. Granitzer, HateBERT: Retraining BERT for abusive language detection in English, in: Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021), Association for Computational Linguistics, Online, 2021, pp. 17–25. URL: https://aclanthology.org/2021.woah-1.3. doi:10.18653/v1/2021.woah-1.3.

[11] T. Caselli, V. Patti, N. Novielli, P. Rosso, Evalita 2018: Overview on the 6th evaluation campaign of natural language processing and speech tools for italian, EVALITA Evaluation of NLP and Speech

Tools for Italian (2018) 3–8. URL: http://dx.doi.org/10.4000/books.aaccad. doi:10.4000/books.aaccademia.4437.

[12] R. Kumar, B. Lahiri, A. Ojha, Aggressive and offensive language identification in hindi, bangla, and english: A comparative study, SN Computer Science 2 (2021). doi:10.1007/s42979-020-00414-6.

[13] C. Bosco, F. Dell'Orletta, F. Poletto, M. Sanguinetti, M. Tesconi, Overview of the evalita 2018 hate speech detection task, EVALITA Evaluation of NLP and Speech Tools for Italian (2018) 67–74. URL: http://dx.doi.org/10.4000/books.aaccademia.4503. doi:10.4000/books.aaccademia.4503.

[14] A. Arango, J. Pérez, B. Poblete, Cross-lingual hate speech detection based on multilingual domain-specific word embeddings, CoRR abs/2104.14728 (2021). URL: https://arxiv.org/abs/2104.14728. arXiv:2104.14728.

[15] P. Fortuna, S. Nunes, A Survey on Automatic Detection of Hate Speech in Text, ACM Computing Surveys 51 (2019) 1–30. URL: http://dx.doi.org/10.1145/3232676. doi:10.1145/3232676.

[16] C. Corazza, S. Menini, E. Cabrio, S. T. S. Villata, Cross-Platform Evaluation for Italian Hate Speech Detection, Le Centre pour la Communication Scientifique Directe - HAL - Université de Nantes (2019).

[17] I. Markov, I. Gevers, W. Daelemans, An Ensemble Approach for Dutch Cross-Domain Hate Speech Detection, Natural Language Processing and Information Systems (2022) 3–15. URL: http://dx.doi.org/10.1007/978-3-031-08473-7_1. doi:10.1007/978-3-031-08473-7/1.

[18] D. Njagi, Z. Zuping, D. Hanyurwimfura, J. Long, A lexicon-based approach for hate speech detection, International Journal of Multimedia and Ubiquitous Engineering 10 (2015) 215–230. doi:10.14257/ijmue.2015.10.4.21.

[19] T. Davidson, D. Warmsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, Proceedings of the Eleventh International Conference on Web and Social Media (2017) 512–521. URL: http://dx.doi.org/10.5555/3290605.3300749. doi:10.5555/3290605.3300749.

[20] M. Lai, S. Menini, M. Polignano, V. Russo, R. Sprugnoli, G. Venturi, Evalita 2023: Overview of the 8th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023), CEUR.org, Parma, Italy, 2023.

[21] F. Celli, M. Lai, A. Duzha, C. Bosco, V. Patti, Policycorpus XL: an italian corpus for the detection of hate speech against politics, in: E. Fersini, M. Passarotti, V. Patti (Eds.), Proceedings of the Eighth Italian Conference on Computational Linguistics, CLiC-it 2021, Milan, Italy, January 26-28, 2022, volume 3033 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021. URL: https://ceur-ws.org/Vol-3033/paper38.pdf.

[22] A. Ramponi, B. Testa, S. Tonelli, E. Jezek, Addressing religious hate online: from taxonomy creation to automated detection, PeerJ Comput. Sci. 8 (2022) e1128. URL: https://doi.org/10.7717/peerj-cs.1128. doi:10.7717/peerj-cs.1128.

[23] Twitter, Documentation, Twitter Developer Documentation, 2023. URL: https://developer.twitter.com/en/docs, Accessed: 13th June 2023.

[24] S. J. Pan, Q. Yang, A Survey on Transfer Learning, IEEE Transactions on Knowledge and Data Engineering 22 (2010) 1345–1359. URL: http://dx.doi.org/10.1109/tkde.2009.191. doi:10.1109/tkde.2009.191.

[25] L. Eric, R. Saini, G. Kovács, K. Murphy, TheNorth @ HaSpeeDe 2: BERT-based Language Model Fine-tuning for Italian Hate Speech Detection, EVALITA Evaluation of NLP and Speech Tools for Italian - December 17th, 2020 (2020) 142–147. URL: http://dx.doi.org/10.4000/books.aaccademia.6989. doi:10.4000/books.aaccademia.6989.

[26] F. Tamburini, How "BERTology" Changed the State-of-the-Art also for Italian NLP, Proceedings of the Seventh Italian Conference on Computational Linguistics CLiC-it 2020 (2020) 415–421. URL: http://dx.doi.org/10.4000/books.aaccademia.8920. doi:10.4000/books.aaccademia.8920.

[27] D. Nozza, F. Bianchi, G. Attanasio, HATE-ITA: Hate Speech Detection in Italian Social Media Text, Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH) (2022). URL: http://dx.doi.org/10.18653/v1/2022.woah-1.24. doi:10.18653/v1/2022.woah-1.24.

[28] F. He, T. Liu, D. Tao, Control batch size and learning rate to generalize well: Theoretical and empirical evidence, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/dc6a70712a252123c40d2adba6a11d84-Paper.pdf.

[29] S. E. Ada, E. Ugur, H. L. Akin, Generalization in transfer learning, CoRR abs/1909.01331 (2019). URL: http://arxiv.org/abs/1909.01331. arXiv:1909.01331.

[30] L. Shao, F. Zhu, X. Li, Transfer learning for visual categorization: A survey, IEEE Transactions on Neural Networks and Learning Systems 26 (2015) 1019–1034. doi:10.1109/TNNLS.2014.2330900.