Identifying sexism in text sentences
NLP Course Project

Davide Brescia, Daniele Marini and **Iulian Zorila**
Master's Degree in Artificial Intelligence, University of Bologna
{ davide.brescia, daniele.marini3, iulian.zorila }@studio.unibo.it

Abstract

Identifying signs of discrimination in texts can be an excellent tool for preventing harmful behavior. In our paper we will focus solely on identifying signs of sexism and classifying them. To be more specific we are going to complete 3 different tasks: Task A - Identification of sexism (e.g. sexism, non-sexism), Task B - Sexism categories (e.g. threats, plans to harm and incitement, derogation, animosity and prejudiced discussions), Task C - Sexism subcategories (e.g. descriptive attacks, incitement and encouragement of harm and aggressive and emotive attacks). After choosing a baseline model to carry out each of these tasks we applied regularization and data augmentation techniques to obtain the best possible results, which were evaluated using the macro F1 metric. The best model that completed Task A was RoBERTa pretrained on hate speech, for Task B hateBERT with learning rate scheduler with an F1 score of 0.62 and lastly for Task C again RoBERTa pretrained on hate speech but using the learning rate scheduler and data augmentation, reaching an F1 score of 0.83 and 0.38 respectively.

1 Introduction

Sexism is a form of discrimination based on a person's gender or sex. Sexism can lead to violence and creates an oppressive environment that prevents most women from fully participating in public life. Creating a model that is able to identify sexist texts could lead to several positive outcomes for society. Such a model could be used to prevent and counter sexist language in different situations, such as in public and private communications, media, and social networks. This could help promote gender equality and create a more inclusive and respectful environment.

Using the dataset proposed in SemEval-2023 Task 10: Explainable Detection of Online Sexism (Kirk et al., 2023) we are going to perform a text classification task using deep learning techniques.

For each of the tasks proposed by SemEval, we used three initial baselines: LSTM, BERTTiny and DistilRoBERTa (Section 2.1). Then with the model that performed best we worked to apply different techniques in order to improve performances such as: bigger model, learning rate scheduling, pretrained model with similar task and data augmentation (Section 2.2). Then we compared these models and performed evaluation on the best one (Section 4).

2 System description

Given that all three tasks proposed by the competition are classification tasks, it was considered useful to employ a consistent pipeline for each task. Specifically, for each of the tasks presented by the dataset, two stages were considered:

- **Baseline Phase**: This phase involved the use of simple models to ensure the correct functionality of the code, and to initially evaluate which models work best for the given task.
- **Advanced Phase**: Following the identification of the most effective baseline model, advanced techniques were then applied to further improve the results.

At the end of each task, the best-performing model will be identified and an evaluation of it will be carried out, finding strengths and weaknesses.

2.1 Baseline Phase

The initial models used for each of the tasks are:

- A model using **LSTM** with GloVe 27B twitter. In order to apply correctly the LSTM in PyTorch we have repurposed the code of the following websites (Saini; Ng and Fu)

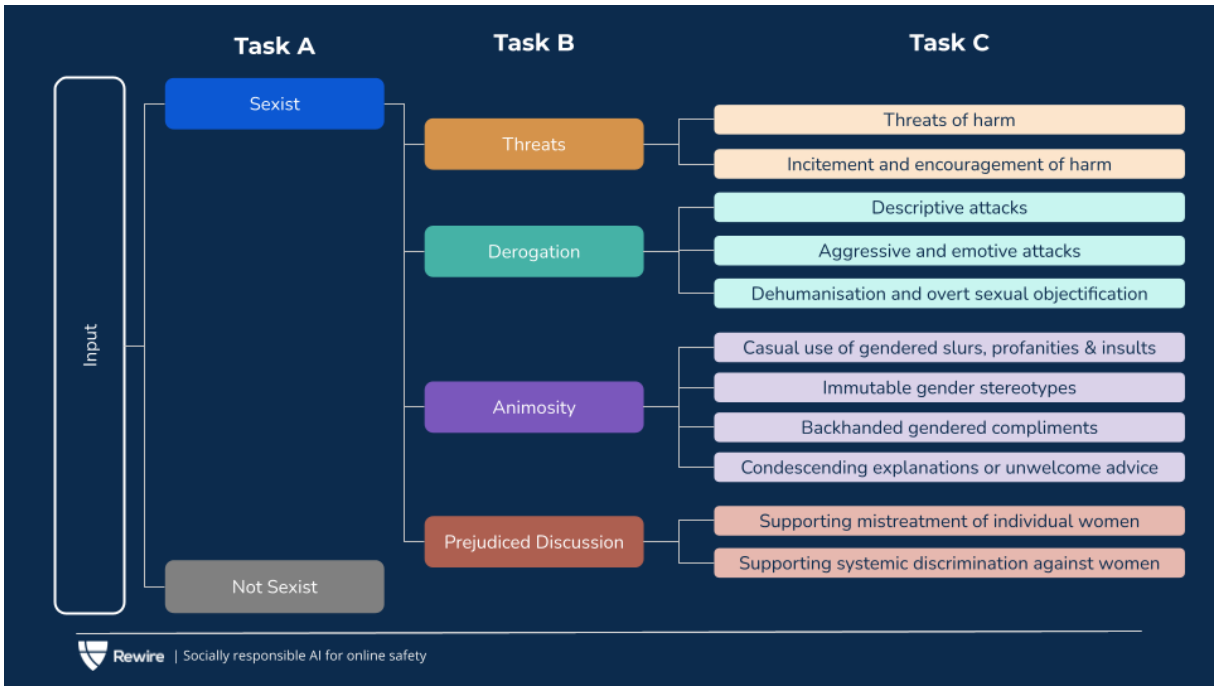

Figure 1: SemEval-2023 Task 10: Explainable Detection of Online Sexism - Tasks Details

- **BertTiny** (Jiao et al., 2019), a lightweight and efficient version of the BERT model that is designed for resource-constrained devices and applications. It achieves its smaller size by reducing the number of layers and the size of the hidden layers, while still maintaining a high level of performance.

- **DistilRoberta** (Sanh et al., 2019), a smaller and faster version of the RoBERTa model that achieves similar levels of performance while using fewer parameters. It achieves its smaller size by using a combination of knowledge distillation and parameter pruning techniques during pre-training.

2.2 Advanced Phase

In an effort to improve the performance of our models, we tried several techniques:

- **Larger models**: we have tried using models with larger capacities such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019)).

- Different **Optimizers**: Adam (Kingma and Ba, 2014), AdamW (Loshchilov and Hutter, 2017), SGD (Ruder, 2016), Adagrad (Ward et al., 2021) and RMSprop.

- **Pretrained models** on a similar topic: Roberta pretrained on TwitterEval (Basile et al., 2019) and Bert pretrained on more than 1 million posts from banned communites from Reddit (Caselli et al., 2021).

- **Data augmentation** techniques with `nlpaug` module (Ma, 2019), employing Contextual Word Embeddings. More informations in Paragraph 2.2.1.

- **Learning Rate Schedulers**: StepLR, OneCycleLR, CosineAnnealingLR and ReduceLROnPlateau.

- **Combine successful techniques**: whenever models proved to be performing a good idea to combine them together

Despite our efforts, however, we observed only marginal improvements in the performance metrics (Section 4). As trying every possible combination for all the tasks was not feasible from a computational standpoint, we began by focusing on task A and testing all of its combinations. We then selected the best combinations and applied them to the other tasks. However, the results indicated that the regularization techniques we employed were frequently ineffective in improving the performance of our models.

We believe that the limited amount of data (especially in Task B and Task C) and a strong

class imbalance in our dataset may have contributed to the lack of improvement in our model performance. We observed that the only exception to this trend was the application of data augmentation techniques on baseline models and RoBERTa pretrained on hate speech, which showed a substantial improvement in the performance of the classification (section 2.2.1).

2.2.1 Data Augmentation

We focused on **Contextual Word Embeddings** augmentation to increase our training set, using two different models to generate words with similar meaning based on the context: BERT base model (cased) and RoBERTa base.

Vocabulary: the main difficulty regards the particular words written by users within the dataset, that is, slang terms, racial slurs, multiple insults and overall informal language, which makes harder the task of generating different but meaningful examples.

Methods: there are various methodologies of augmentation, from basic to complex, from less to more resource demanding, for instance EDA (Easy Data Augmentation) (Wei and Zou, 2019) performs simple operations of synonim replacement, random insertion, deletion and swapping, while LAMBADA (language-model-based data augmentation) (Anaby-Tavor et al., 2019) requires training a classifier, which implicitly solves the classification problem, and fine-tuning a Seq2Seq model (e.g. GPT2). We decided to strike in the middle and rely on contextual embeddings to augment the training set.

Degrees of freedom: besides trying single methodologies, which depending on the complexity can have few or many hyperparameters, it is also possible to combine them together in a sequential pipeline, complicating significantly the evaluation of the genereted data and making it harder to understand which method contributed the most and which not. Therefore to easen up the augmentation part we opted for an individual application, without employing compositions.

We choose the following **Configuration**:

- Insertion turned out to preserve better the semantic meaning of the sentence, while changing the syntax, compared to word substitution.
- Augmentation probability has been set to 0.3, otherwise if `aug_p` ≥ 0.5 the sentences result to be too noisy, losing the original meaning and confusing the model.
- Skipped words from data augmentation process are `[USER]` and `[URL]`.
- Models used are `bert-base-cased` and `roberta-base`, but later **decided to use the data generated by the latter**, as it proved to be more effective with the majority of used classification models.

3 Data

The dataset used is associated with the competition SemEval-2023 Task 10: Explainable Detection of Online Sexism (Kirk et al., 2023). As can be read from the official competition page (Kirk et al.), there are 3 distinct tasks each of classification:

- **Task A: Binary Sexism Detection**: In this task, systems are required to predict whether a post is sexist or not, by performing a two-class (or binary) classification.
- **Task B: Category of Sexism**: This task involves the classification of posts that are identified as sexist into one of four categories: threats, derogation, animosity, or prejudiced discussions, by performing a four-class classification.

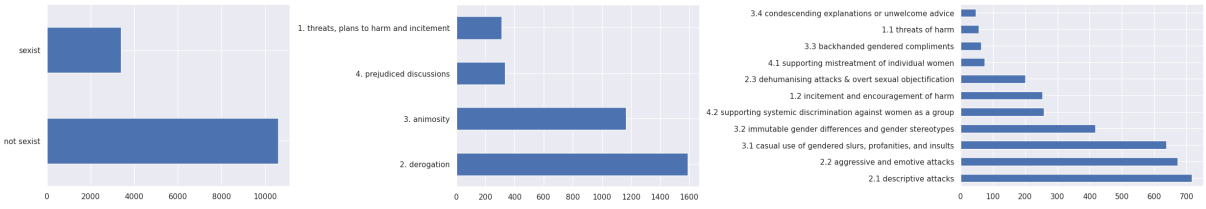

Figure 2: Dataset Distribution. The first graph depicts data in Task A, the second in Task B and the last in Task C.

	DistilRoBERTa w/ LR Scheduler	RoBERTa pretrained on Hate Speech	RoBERTa pretrained on Hate Speech w/ LR Scheduler	HateBERT	HateBERT w/ LR Scheduler	LSTM w/ Augmentation	BERTTiny w/ Augmentation	DistilRoBERTa w/ Augmentation	RoBERTa pretrained on Hate Speech w/ LR Scheduler and Augmentation	HateBERT w/ LR Scheduler and Augmentation
Task A	0.8	**0.83**	0.82	0.82	0.82	0.74	0.73	0.78	0.82	0.82
Task B	0.56	0.6	0.6	0.59	**0.62**	0.35	0.43	0.54	0.61	0.59
Task C	0.32	0.31	0.28	0.33	0.36	0.1	0.19	0.35	**0.38**	0.34

Table 1: Macro F1 scores of advanced models

- **Task C: Fine-grained Vector of Sexism**: For posts that are identified as sexist, this task involves the prediction of one of the 11 fine-grained vectors by performing an 11-class classification.

With around 14,000 labeled sentences, the dataset proposal shows a noticeable problem of data imbalance, as depicted in Figure 2.

4 Experimental setup and results

The techniques employed for each task in the competition, along with the corresponding results, will be outlined below.

	LSTM	BERTTiny	DistilRoBERTa
Task A	0.73	0.74	**0.78**
Task B	0.16	0.35	**0.55**
Task C	0.03	0.21	**0.28**

Table 2: Macro F1 scores of baseline models

4.1 Task A

Since this is precisely a binary classification task, the results are quite high. The best-performing baseline model was DistilRoBERTa, which achieved an accuracy of 83% and an F1 Score of 0.78. We then applied several regularization techniques and found that using **RoBERTa pretrained on hate speech dataset** yielded the best results, with an accuracy of 87.81% and an F1 score of 0.83. Notably, even though the LSTM model had limited capacity, it still performed decently. Our findings suggest that regularization techniques can slightly improve the results of the baseline model, and using a models pretrained on hate speech can further enhance the accuracy and F1 score of the classification task.

The points to focus on in our best model in the task A are:

1. The usage of slang challenges the model, as these words are extremely rare and the model is not always able to associate the right meaning, e.g. "*True. Landwhales have the pick of normal guys because Chad gets all the Top Stacies.*" is classified as non-sexist, instead it is a sexist phrase. In this specific case, the model fails to correctly identify the term "*Landwhales*" which is the core of the statement and represents a negative term referring to a woman within this context. In contrast, with more common terminologies (such as common insults) the model makes fewer errors.

2. The model usually classifies offensive phrases as sexist, and conversely, sexist phrases that perhaps do not contain insults or negative terms are more likely to be misclassified as non-sexist phrases. This is justified by the argument that the model was pre-trained on a hate speech dataset. To be more detailed, sentence "*Because they're afraid of confrontation, a woman will always escape*" is sexist and does not contain any particular offensive terms, yet it is classified as non-sexist by the model. In contrast, the sentence "*If my daughter got pregnant from this shit, That judge has a lawsuit coming!*" is non-sexist but contains derogatory terms, in which case the model makes the opposite error.

3. The model has a higher chance of classifying sexist samples as nonsexist (Figure 3). This can be attributed to the unbalanced dataset problem.

4.2 Task B

As expected, the performance of the classifiers in Task B was lower than that in Task A, considering the increasing difficulty of the problem. The best-performing baseline model for both tasks

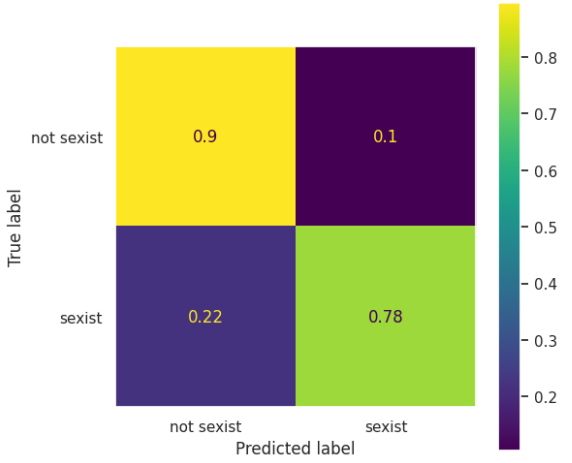

Figure 3: Confusion Matrix of the best model - Task A

was DistilRoBERTa, with an accuracy of 57.56% and an F1 score of 0.55. After applying several regularization techniques to enhance the performance of the models, we found that **hateBERT**, a model trained on the hate speech dataset, achieved the best overall performance with the One Cycle learning rate scheduling technique. This model reached an accuracy of 65.23% and an F1 score of 0.62. Furthermore, we observed that even with a slightly more complex classification problem in Task B, there was a substantial difference between the performance of larger and smaller models. This emphasizes the importance of using larger models for more complex tasks. Finally, we found that the application of regularization techniques was more effective for these tasks because the models were able to better generalize and consequently increase metric scores.

The things to emphasize about our best model of task B are:

1. The "derogation" class (language which explicitly derogates, dehumanises, demeans or insults women) is the class that the model predicts most often (Figure 4). This is related to the fact that "derogation" is the majority class in this task (Figure 2).

2. The class "prejudiced discussions" (language which denies the existence of discrimination, and justifies sexist treatment) is the class with the highest degree of misclassification (f1 score equal to 0.35). Again we believe that this poor performance is related to the presence of few samples for this class within the dataset.

3. For this task we have a relatively low number of sentences within the test set, as we can well imagine performing an evaluation considering such a low number of values means having a lot of inaccuracy in the final results.

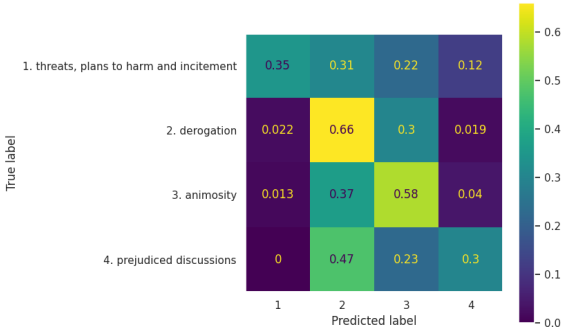

Figure 4: Confusion Matrix of the best model - Task B

4.3 Task C

This was the hardest problem among the previous two ones, having a total of 11 classes of sexism, grouped by categories, with heavy data imbalance (Figure 2, rightmost graph). Indeed, the models had a hard time classifying such examples. In particular, once again for the baseline DistilRoBERTa has overtaken the other two, reaching an F1 score of 0.28 and 46.37% accuracy. As for the advanced section **RoBERTa pretrained on hate speech** with the addition of the OneCycle learning scheduler and data augmentation was able to get to 0.38 F1 and 48.02% accuracy.

Things to consider about the best model of this task are:

1. The worst performing classes are: "3.4 condescending explanations or unwelcome advice" (Offering unsolicited or patronising advice to women on topics and issues they know more about) with F1 Score equal to zero and "4.1 supporting mistreatment of individual women" (Expressing support for mistreatment of women as individuals. Support can be shown by denying, understating, or seeking to justify such mistreatment) with F1 Score equal to 0.22.

2. Here again the problem of the previous task is highlighted: looking at Figure 5 the model

overly predicts the 2.1 (descriptive attacks) and 2.2 (aggressive and emotional attacks) subclasses along with 3.1 (casual use of gendered slurs, profanities, and insults) and 3.2 (immutable gender differences and gender stereotypes) which are the majority subclasses (Figure 2, last graph).

3. Once again, the low number of samples in the test set makes the evaluation of this model very approximate and inaccurate.

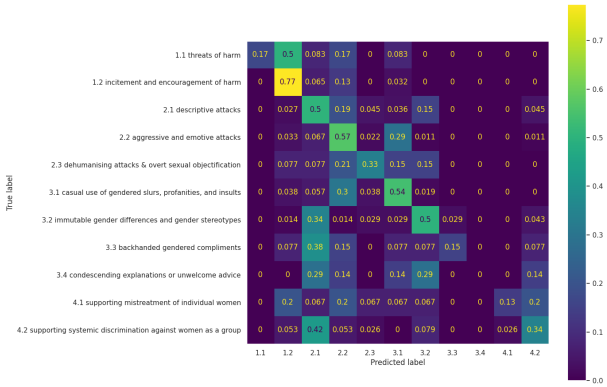

Figure 5: Confusion Matrix of the best model - Task C

5 Discussion

It is necessary to make an important premise about our dataset, in some sentences the boundary is blurred, and the classification can be correct either way by considering more than one point of view or more than one line of reasoning. Moreover, it is extremely difficult for a non-expert to distinguish between classes in tasks C and B, even A in some cases. This makes this classification task even more difficult by making the irreducible error very high. Many sentences also are **extrapolated without context**, which makes classification further complex

We noted that the **smaller models performed reasonably well in the binary classification task**. However, in Task C, a more complex classification problem, they had lower performance compared to a higher capacity model. This highlights the importance of using more complex models for more challenging tasks.

Moreover, we observed that the **impact of regularization techniques**, particularly LR Scheduling, **was more significant as the complexity of the task increased**. In some cases, applying these techniques resulted in improving model performance. These findings demonstrate the importance of using appropriate regularization techniques for different types of classification tasks.

We also noticed in our discussion that **changing the optimizer or testing different loss functions did not have a significant impact on the final results**. This suggests that, in our specific experimental setup, the choice of optimizer and loss function had a minimal effect on model performance.

We observed that the **data augmentation techniques employed improved the results slightly in general, but worked best in task C for a specific model**. This could be due to the fact that our text augmentation technique depends on the model used to generate similar words, i.e. `roberta-base`.

The **results of the augmented data, strongly depend on the task type and the model used for classification**. For instance in task A, a binary classification problem, there is more data at our disposal and is trivial to obtain good results, hence the augmentation process hardly improves them, when training the baseline models. However when it comes to task B and C, since we consider just the sexist examples, which are far less then the non-sexist ones and the problem gets harder (four and eleven categories respectively), the performance increased significantly for some models (LSTM, BERTTiny and RoBERTa pretrained on twitter) and decreased for others (hateBERT with learning rate scheduler and DistilRoBERTa).

In particular data augmentation worked remarkably well for **RoBERTa pretrained on hate speech** with learning rate scheduler in Task C, going from 0.28 to 0.38, but overall it always helped for this specific model the augmentation process as we can see from table 1. On the other hand for hateBERT the F1 score got even worse compared with the original data. We suspect that this particular augmentation technique adds more noise than actual useful information to the model.

6 Conclusion

Overall, our study highlights the importance of selecting appropriate models and regularization techniques for classification tasks of varying complexity. Future research in this area could explore the use of other regularization techniques or investigate the impact of different hyperparameters on model performance and uses specific data preprocessing techniques. In addition, it might be interesting to use a hierarchical approach to these classification tasks to check for variations in the performances. Furthermore the performance gains from data augmentation were not well suited for every model, and future work should explore more effective techniques for augmenting informal text datasets.

References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2019. Not enough data? deep learning to the rescue! *CoRR*, abs/1911.03118.

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

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. HateBERT: Retraining BERT for abusive language detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Hannah Rose Kirk, Wenjie Yin, Bertie Vidgen, and Paul Röttger. https://codalab.lisn.upsaclay.fr/competitions/7124.

Hannah Rose Kirk, Wenjie Yin, Bertie Vidgen, and Paul Röttger. 2023. SemEval-2023 Task 10: Explainable Detection of Online Sexism. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Ritchie Ng and Jie Fu. Long short-term memory (lstm) network with pytorch - deep learning wizard.

Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747.

Amar Saini. Nlp from scratch with pytorch, fastai, and huggingface.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Rachel Ward, Xiaoxia Wu, and Leon Bottou. 2021. Adagrad stepsizes: Sharp convergence over nonconvex landscapes.

Jason W. Wei and Kai Zou. 2019. EDA: easy data augmentation techniques for boosting performance on text classification tasks. *CoRR*, abs/1901.11196.