TEDB System Description to a Shared Task on Euphemism Detection 2022

Peratham Wiriyathammabhum
peratham.bkk@gmail.com

Abstract

In this report, we describe our Transformers for euphemism detection baseline (TEDB) submissions to a shared task on euphemism detection 2022. We cast the task of predicting euphemism as text classification. We considered Transformer-based models which are the current state-of-the-art methods for text classification. We explored different training schemes, pretrained models, and model architectures. Our best result of *0.816* F1-score (*0.818* precision and *0.814* recall) consists of a euphemism-detection-finetuned TweetEval/TimeLMs-pretrained RoBERTa model as a feature extractor frontend with a KimCNN classifier backend trained end-to-end using a cosine annealing scheduler. We observed pretrained models on sentiment analysis and offensiveness detection to correlate with more F1-score while pretraining on other tasks, such as sarcasm detection, produces less F1-scores. Also, putting more word vector channels does not improve the performance in our experiments.

1 Introduction

A shared task on euphemism detection (Gavidia et al., 2022; Lee et al., 2022) is the first installment of a natural language processing (NLP) shared task on a particular figurative language detection, euphemism. Figurative languages, including metaphors, synecdoches, idioms, puns, hyperbole, similes, onomatopoeia, and others, are word uses where the meaning deviates from the literal meaning to convey a complicated, creative and evocative message without directly stating it. In addition, figurative language might use contexts such as relations to other things, actions, social experiences, or images. Figurative languages are ubiquitous since they are filled in countless of our everyday activities without notice (Lakoff and Johnson, 2008).

Table 1: An Example instance from the shared task dataset. The first sentence is more offensive literally. The phrase “collateral damage" should be replaced with politeness. The second sentence was revised by using the phrase “advanced age" to provide more politeness than some possible words or phrases like old, near expiration, or wrinkly.

Sentence	Label
All the deaths were just <collateral damage> in their cause.	[non-euphemistic]
In spite of his <advanced age>, Rollins remains one of jazz’s most talented improvisers.	[euphemistic]

Euphemisms are mild or indirect words or phrases being used in place of offensive or unpleasant ones. Moreover, euphemisms are used to mark profanity or politely refer to sensitive and taboo topics such as death, disability, or sickness. The applications of euphemisms involve social interactions such as politics or doctor-patient discourses. Euphemisms can also be dangerous since terrorists can use euphemisms for language manipulation and separate message and meaning (Matusitz, 2016). Also, politely calling terrorism results in semantic deviance and attention away from reality for media and government officials which makes citizens lower their guard while in danger.

Previous works (Gavidia et al., 2022; Lee et al., 2022) utilize RoBERTa models (Liu et al., 2019) for sentiment and offensive ratings because politeness is the aim of euphemisms. Euphemisms should make the sentences more positive in sentiment and less offensive (Bakhriddionova, 2021). Our systems build upon these findings and explore transformer-based models which are pretrained for sentiment analysis or offensive detection.

Our best submission ranks 6th on the leaderboard. The codes for our systems are open-sourced and available at our GitHub repository[1].

[1] https://github.com/perathambkk/euphemism_

Proceedings of the 3rd Workshop on Figurative Language Processing (FLP), pages 1 - 7
December 8, 2022 ©2022 Association for Computational Linguistics

2 Models

2.1 Pretrained Transformers

Huggingface library (Wolf et al., 2020) is an extensive platform for transformer models (Vaswani et al., 2017). Huggingface provides many checkpoints for the pretrained transformer suitable to many tasks as a model hub. TweetEval (Barbieri et al., 2020) is a social NLP benchmark where standardized evaluation protocols and strong baselines and employed on seven Twitter classification tasks. The strong baselines later became pretrained model checkpoints loadable via Huggingface.

Diachronic specialization was shown to be lacking in language models (Loureiro et al., 2022) where changes or evolution in time can break current (synchronic - a language at a moment in time without any histories.) language model performances entirely. For example, pre-COVID19 language models will have no knowledge about the pandemic events completely. Diachrony and synchrony are two complimentary viewpoints that were theorized by linguist Ferdinand de Saussure more than a hundred years ago (De Saussure, 2011). The paper shares many time-specific language model checkpoints (TimeLMs).

Specifically, we employed two RoBERTa language model checkpoints from the papers (TweetEval and TimeLMs), one for sentiment analysis ('cardiffnlp/twitter-roberta-base-sentiment-latest') and another for offensiveness detection ('cardiffnlp/twitter-roberta-base-offensive'), as in (Gavidia et al., 2022; Lee et al., 2022). We finetuned them for euphemism detection as text classification.

2.2 Convolutional Neural Networks Backend

Convolutional Neural Networks (CNN) were primarily introduced for visual tasks, firstly, handwritten digit recognition, given its properties in translation invariance for 2D data (LeCun et al., 1998). KimCNN (Kim, 2014) proposed a little modification that enables on-top finetuning of CNN over pretrained word vectors for sentence classifications. The results in the paper were from a simple CNN with a little parameter tuning and static vectors.

We further performed some modifications by concatenating hidden state outputs from all RoBERTa layers as a word vector and instead finetuning the whole model end-to-end. We also used checkpoints from finetuning the pretrained transformers as RoBERTa starting points. We also attempted to combine two word vectors for a multichannel KimCNN and finetuning the model with both word vectors for sentiment analysis and offensiveness detection end-to-end in contrast to freezing one word vector channel as in the original paper.

3 Experimental Setup

Our input consists of a three-sentence utterance, the sentence before, after, and the sentence containing the euphemistic term. We did not observe any improvements from removing any special characters including the '<' and '>' symbols around the euphemistic term given in the dataset. We used the maximum input length of 150 tokens since we found that it is the number that fits well as our heuristics with the GPU memory for many reasonable batch sizes ($4-20$ in our cases). Also, it seems to cover most data instances given the histogram plotting in Figure 1. We sampled the model at the end of each epoch. The dataset has 1572 training instances and 393 test instances.

All of our experiments were done in the Google Colab setting on NVIDIA Tesla T4 GPUs. We used the batch size in the range of $4 - 20$ and the learning rate for an AdamW optimizer (Loshchilov and Hutter, 2018) in the set of $\{2.5e-5, 2e-5, 1e-5, 7.5e-6\}$ for all experiments. We considered linear annealing scheduler and cosine annealing scheduler with restart. The cycle number is in the set of $\{5, 8\}$. Also, adding a warm-up step does not make any difference so we set the warm-up step to zero in all experiments.

3.1 Early Stopping Criterion for Empirical Risk Minimization

We employed the early stopping with zero patience training strategy schema (Prechelt, 1998; Bengio, 2012). We varied the training epoch until the training metric saturated with manual monitoring, and then stopped right at the end of that epoch. We tried to split the training data into training and development sets but empirically we found that the data set size is too small to perform accurate estimations/cross-validations on just an efficient held-out schema. For these reasons, we relied solely on our heuristics on the training set instead.

Theoretically simply speaking, given a small data for finetuning, it is not easy to estimate the

shared_task_emnlp2022

Table 2: Test F1-scores of different pretrained transformers on euphemism detection. (The number in **bold** is for the best score, and in *italic* is for the second best.)

Pretrained Transformer	Test F1-score
‘cardiffnlp/xlm-twitter-politics-sentiment’	0.4693
‘Hate-speech-CNERG/dehatebert-mono-english’	0.6821
‘mrm8488/t5-base-finetuned-sarcasm-twitter-classification’	0.6969
‘finiteautomata/bertweet-base-sentiment-analysis’	0.7349
a strong finetuned vanilla baseline: ‘roberta-base’	0.7776
‘sagteam/covid-twitter-xlm-roberta-large’	0.7776
‘cardiffnlp/twitter-roberta-base-offensive’	*0.7838*
another strong finetuned vanilla baseline: ‘bert-base-cased’	0.7941
‘cardiffnlp/twitter-roberta-base-sentiment-latest’ (TimeLMs)	**0.8064**

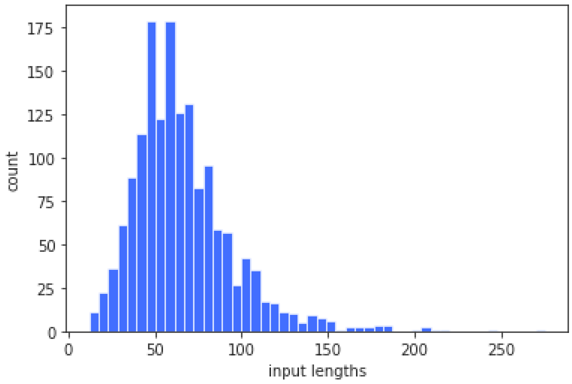

Figure 1: **The distribution of the input length derived from the shared task training set.**

model performance using a held-out validation set. Leave-one-out cross-validation (LOOCV) is appropriate but might need much more computation costs. Even k-fold cross-validation with a high value of k, which is a less extreme case of LOOCV, still needs a lot of computation costs. Additionally, if we split a small data, our model might fit the train split, but not the validation split. That model is very unlikely to perform well on the validation split, especially when the training is still underfitting the task, given a small data to train and a data-hungry model with a large capacity. Therefore, it will certainly have a weak upper bound of its error against a model that fits the whole training data.

This gives us an intuition of training our models just to shatter the whole training data and stop training in a basic train-test setting (empirical risk minimization). In our other simple intuition, it would be weird to withhold some training data from a given small data, implicitly lower the model capacity by (randomly) filtering out some data for an inaccurate generalizability estimation, and let the model predict them wrongly. Also, using more data to train lowers the model variance error term in the bias-variance decomposition framework.

3.2 Finetuning Pretrained Transformers

We compare many available huggingface hub’s pretrained checkpoints we feel suitable for the task, which are multilingual Twitter politics sentiment analysis (Antypas et al.), hate speech detection (Aluru et al., 2020), Twitter sarcasm detection (Ghosh et al., 2020; Raffel et al., 2020), Twitter English sentiment analysis (Nguyen et al., 2020; Loureiro et al., 2022), Multilingual Russian-English Twitter COVID-19 report detection (Sboev et al., 2021), and offensiveness detection (Barbieri et al., 2020). The transformer models include BERT (Kenton and Toutanova, 2019), RoBERTa (Liu et al., 2019), XLM (Conneau et al., 2018), XLM-RoBERTa (Conneau et al., 2020) and T5 (Raffel et al., 2020) which are finetuned for the target task and their model parameters are shared on huggingface hub.

From the test F1-scores in Table 2, in which we even report the best result from all model hyperparameter settings in our experiment not reported here for brevity, we tend to confirm the hypothesis in the aforementioned previous works (Gavidia et al., 2022; Lee et al., 2022; Bakhriddionova, 2021) which state that euphemism relates with sentiment and offensiveness because the top-2 best scores in the table are sentiment analysis and offensiveness detection. Also, multilingual pretraining seems not to be helpful in this case of English euphemism detection. The ‘cardiffnlp/twitter-roberta-base-sentiment-latest’ RoBERTa-base model seems to outperform the ‘finiteautomata/bertweet-base-sentiment-analysis’ BERTweet model as in the TimeLMs paper (Loureiro et al., 2022) too. Therefore, we further build our models based on these top-2 best scorer pretrained TweetEval/TimeLMs RoBERTa models (Gururangan et al., 2020). We are aware that these top-2 models were among pre-

Table 3: Test F1-scores of different TweetEval pretrained transformers (Barbieri et al., 2020) on euphemism detection. (The number in **bold** is for the best score, and in *italic* is for the second best.)

Pretrained Transformer	Test F1-score
'cardiffnlp/twitter-roberta-base-stance-climate'	0.7238
'cardiffnlp/twitter-roberta-base-sentiment'	0.7238
'cardiffnlp/twitter-roberta-base-stance-feminist'	0.7306
'cardiffnlp/twitter-roberta-base-stance-abortion'	0.7446
'cardiffnlp/twitter-roberta-base-emotion'	0.7588
'cardiffnlp/twitter-roberta-base-emoji'	0.7615
'cardiffnlp/twitter-roberta-base-stance-hillary'	0.7651
'cardiffnlp/twitter-roberta-base-hate'	0.7665
'cardiffnlp/twitter-roberta-base-irony'	*0.7688*
'cardiffnlp/twitter-roberta-base-stance-atheism'	*0.7688*
a strong finetuned vanilla baseline: 'roberta-base'	0.7776
'cardiffnlp/twitter-roberta-base-offensive'	**0.7838**
another strong finetuned vanilla baseline: 'bert-base-cased'	0.7941

trained language models using the most data in TweetEval/TimeLMs.

3.2.1 TweetEval Pretrained Language Models

However, when we additionally compared all TweetEval pretrained RoBERTa-base language models finetuned on the euphemism task using our training scheme in Table 3, we observed that a TweetEval sentiment analysis model does not perform well at all. Besides, it was pretrained using much less data than the one in TimeLMs (45k vs. 138.86M tweets). Still, in Figure 2, the TimeLMs sentiment classification model performs very well given lots of data. The sentiment classification task might have some correlations with euphemism detection when the model learns well, or just lots of data make it work.

The best result in Table 3 is from offensiveness detection with only 11k tweet data. The second best models are irony detection and stance detection in the target domain of atheism. The performances vary based on some degree of euphemisms in the pretrained data. Nevertheless, only the offensiveness detection language model performs better than a finetuned vanilla RoBERTa-base language model. Finally, this is only our evidence-based intuition based on some point estimations of the model performances on euphemism detection.

We observed high sensitivities in hyperparameter settings in these experiments. Changing some hyperparameters such as patience in early stopping, initial learning rate, learning rate scheduler cycle, or even the random seed can result in significant changes in the results as in typical transformer models which are known to be sensitive to perturbations (Dodge et al., 2020). Training the 'cardiffnlp/twitter-roberta-base-sentiment-latest' model until the training metric is saturated but using a linear scheduler for 10 epochs instead of the best 15 epochs and removing special characters can result in 0.6920 test F1-score, using a linear scheduler for 12 epochs and removing special characters can result in 0.7301 test F1-score, which both are significant degradation.

Table 4: Test F1-scores of different classifiers on euphemism detection using vanilla pretrained language models. (The number in **bold** is for the best score.)

Model	RoBERTa-base
Huggingface's classifier	0.5203
sklearn logreg	0.4376
PA classifier	0.4126
3-NN	**0.5446**
MLP	0.4545
Decision Tree	0.4910
Linear SVM	0.4125

Model	BERT-base-cased
Huggingface's classifier	0.4197
sklearn logreg	0.5062
PA classifier	**0.5239**
3-NN	0.4436
MLP	0.4927
Decision Tree	0.4315
Linear SVM	0.4125

Table 5: Validation F1-scores of different classifiers on euphemism detection using vanilla pretrained language models. The split ratio is 0.40. (The number in **bold** is for the best score.)

Model	RoBERTa-base
sklearn logreg	0.5954
PA classifier	0.5929
3-NN	0.6107
MLP	0.6438
Decision Tree	**0.6692**
Linear SVM	0.6260

Model	BERT-base-cased
sklearn logreg	0.5929
PA classifier	0.5700
3-NN	0.5954
MLP	0.6056
Decision Tree	**0.6743**
Linear SVM	0.6031

3.2.2 A Comparison to Vanilla Pretrained Language Models

We additionally conducted experiments on various classifiers using vanilla pretrained language models, like RoBERTa-base and BERT-base-cased, as fixed feature extractors. From Table 4 and Table 5, the validation F1-scores are not good estimations of any test F1-scores. They overestimate all model performances by some large margins of around $0.12 \sim 0.15$ by their best differences or more. Training a classifier on a fixed feature extractor yields us only at most around ~ 0.54 test F1-score. This is a large gap compared to the performance of most finetuned language models. Also, the classifier with the best validation score, a decision tree, performs poorly on the test set. We used default parameters for the classifiers and used the same early-stopping training scheme but with an initial learning rate of $2.5e-4$.

3.3 Finetuning KimCNNs

We employed the finetuned 'cardiffnlp/twitter-roberta-base-sentiment-latest' RoBERTa from the previous subsection for our KimCNN. We used 100 feature maps and $3, 4, 5$ weight length set input. We use a cross-entropy loss function and cosine annealing scheduler for this model type. Other hyperparameters were the same as in the previous subsection.

We got the best result of 0.8158 test F1-score, approximately 0.01 improvement over the previous model, simply using a KimCNN backend. However, adding another word vector channel using 'cardiffnlp/twitter-roberta-base-offensive', finetuned in the last subsection, reduces the performance as shown in Table 6. We additionally conducted experiments on removing a large language model and used only static word embeddings. A vanilla KimCNN with either word2vec (Mikolov et al., 2013) or glove-twitter (Pennington et al., 2014), trained on euphemism detection, works quite well with 0.6807 and 0.6172 test F1-scores respectively.

Table 6: Test F1-scores of different settings for KimCNNs on euphemism detection. (The number in **bold** is for the best score.)

Model	Test F1-score
KimCNNs	**0.8158**
+ multichannel	0.7980
KimCNNs (word2vec)	0.6807
KimCNNs (glove-twitter)	0.6172

Also, we varied some hyperparameters and observed more stability and faster convergence by simply putting a KimCNN backend on top. The significant degradation in the previous subsection was no longer. The test F1-scores of those models are like 0.8130 or 0.8132 which are very close to the best score. We also observed lower scores and slower convergence from using the 'cardiffnlp/twitter-roberta-base-sentiment-latest' directly from the huggingface's hub for KimCNN. So, another pretraining step to the task by finetuning a model from some relevant task helps improve the overall performance.

4 Conclusion

This report describes our baseline systems for a shared task on figurative language processing 2022, euphemism detection. Our best result is from a single-channel KimCNN model using 'cardiffnlp/twitter-roberta-base-sentiment-latest', pretrained again for euphemism detection, as a feature extractor. We observed more stability and faster convergence from this training schema. Our results on pretrained transformer models are likely to confirm the previous works (Gavidia et al., 2022; Lee et al., 2022; Bakhriddionova, 2021) that euphemism relates with sentiment and offensiveness. Still, we also observed that finetuning a sentiment-based pretrained language model, which pretrained with a rather small dataset, does not perform well.

Limitations

We only sampled a relatively small portion of models and draw conclusions. We also conducted experiments only on one dataset for euphemism detection. We did not perform any strong statistical tests on the models, just point estimations.

The authors are self-affiliated and do not represent any entities.

Acknowledgments

We would like to thank anonymous reviewers for their constructive feedback, and suggestions for additional experiments.

References

Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2020. A deep dive into multilingual hate speech classification. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part V*, page 423–439, Berlin, Heidelberg. Springer-Verlag.

Dimosthenis Antypas, Alun Preece, and Jose Camacho-Collados. Politics, sentiment and virality: A large-scale multilingual twitter analysis in greece, spain and united kingdom. *Spain and United Kingdom*.

Dildora Oktamovna Bakhriddionova. 2021. The needs of using euphemisms. *Mental Enlightenment Scientific-Methodological Journal*, 2021(06):55–64.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650.

Yoshua Bengio. 2012. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Ferdinand De Saussure. 2011. *Course in general linguistics*. Columbia University Press.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Martha Gavidia, Patrick Lee, Anna Feldman, and Jing Peng. 2022. Cats are fuzzy pets: A corpus and analysis of potentially euphemistic terms. In *LREC 2022*.

Debanjan Ghosh, Avijit Vajpayee, and Smaranda Muresan. 2020. A report on the 2020 sarcasm detection shared task. *ACL 2020*, page 1.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

George Lakoff and Mark Johnson. 2008. *Metaphors we live by*. University of Chicago press.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Patrick Lee, Martha Gavidia, Anna Feldman, and Jing Peng. 2022. Searching for pets: Using distributional and sentiment-based methods to find potentially euphemistic terms. In *Proceedings of the Second Workshop on Understanding Implicit and Underspecified Language*.

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. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-collados. 2022. TimeLMs: Diachronic language models from Twitter. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 251–260, Dublin, Ireland. Association for Computational Linguistics.

Jonathan Matusitz. 2016. Euphemisms for terrorism: How dangerous are they? *Empedocles: European Journal for the Philosophy of Communication*, 7(2):225–237.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Lutz Prechelt. 1998. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Alexander Sboev, Ivan Moloshnikov, Alexander Naumov, Anastasia Levochkina, and Roman Rybka. 2021. The russian language corpus and a neural network to analyse internet tweet reports about covid-19.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.