ISSN 2224-087X. Електроніка та інформаційні технології. 2024. Випуск 25. С. 67–80
Electronics and information technologies. 2024. Issue 25. P. 67–80

УДК 519.765:519.767:004.89 DOI: https://doi.org/10.30970/eli.25.6

DATA AUGMENTATION IN TEXT CLASSIFICATION WITH MULTIPLE CATEGORIES

B. Pavlyshenko, M. Stasiuk

System Design Department,
Ivan Franko National University of Lviv,
50 Drahomanova St., 79005 Lviv, Ukraine
bohdan.pavlyshenko@lnu.edu.ua,
mykola.stasiuk@lnu.edu.ua

In the modern world, the amount of text data that is being generated every day is enormous. However, because of the differences in various language usage in day-to-day life, the amount of data generated in English is much greater than for example, Ukrainian. Moreover, there are a huge amount of languages that may become extinct in the near future. Because of this, there is a request for the methods and techniques that will make it possible to preserve endangered languages and will allow us to use them effectively in the machine learning approaches. One of the developed methods for creating new data based on already existing information is called augmentation.

The purpose of this article is to investigate the effect of data augmentation on the multi-class text classification task, which is performed by different transformer models: BERT, DistilBERT, ALBERT, and XLM-RoBERTa. Data for the models' training and testing were taken from the HuggingFace. Data themselves were modified using different augmentation techniques: on the word level synonym, antonym, and contextual word embeddings augmentation were used; on the sentence level abstractive summarization and lambada augmentations were utilized. Instead of direct training and evaluation, training infrastructure, provided by the HuggingFace portal was used. Different metrics of model training efficiency were considered: learning time, the output of validation and training loss functions, accuracy, recall, f1-score, and precision.

The result of this investigation allows comparing the efficiency of every observed model in multi-class text classification tasks. At the same time, the efficiency of different text augmentation was estimated. This is valuable for assessing the most corresponding transformer model in connection with augmentation to obtain the best efficiency in the classification with multiple categories.

Keywords: augmentation, multi-class text classification, BERT, ALBERT, DistilBERT, XLM-RoBERTa.

Introduction.

Data augmentation is a technique used in machine learning to artificially expand the size of a training set by creating modified versions of existing data [1, 2]. Overall augmentations can be performed on different levels: character, word, sentence, etc. In many cases, augmentations can not only generate new data but also help to prevent the overfitting of the machine-learning models. Alongside the advantages, augmentations have their challenges. They require additional computational capabilities. Also, if the original data have biases, generated data also will have the same biases. It is one of the approaches, used to deal with the

threat of a vast majority of languages going extinct [3], or effectively utilizing low-resource languages [4] in machine learning models.

The most common models that are used in modern Natural Language Processing (NLP) are transformer models [5]. Transformers are the models that are capable of efficient parallel processing, meaning they can process sequences in parallel ways, making them more efficient than other models. Furthermore, because of the self-attention mechanism, transformers can handle long-term dependencies in a better way. Finally, they can handle various length inputs, making them more suitable for the NLP tasks. At the same time, transformer models have high computational costs. Moreover, they depend on the data, requiring a huge amount of it to be able to show good performance. Lastly, transformers may face the overfitting problem. This architecture allowed the invention of the Large Language Models (LLM) in the future, which are widely used in the world [6-8]. Commonly used transformers are BERT[9], RoBERTa [10], DistilBERT[11], and ALBERT [12].

BERT is an open-source NLP pre-trained model that was introduced shortly after the general transformer architecture. It was the first deeply bidirectional, unsupervised language representation model to be pre-trained using only a plain text corpus of unlabeled text. RoBERTa, is one of the models, built on top of BERT. Both RoBERTa and BERT use masked language models, but they employ them in different ways. In BERT, masking is performed once during the preparation phase, with each sentence being masked in 10 different ways. In RoBERTa, masking is done dynamically during training whenever a sentence is added to a batch, so the number of different masked versions is not limited as it is in BERT. DistilBERT is a distilled version of BERT. It uses roughly the same architecture as BERT but with some changes, such as fewer encoder blocks, the removal of token-type embeddings, and the pooling functionality. ALBERT, or A Lite BERT, is another BERT-based model introduced around the same time as DistilBERT. It has a smaller model and can be trained faster, but it does not achieve these gains by sacrificing performance, unlike DistilBERT. The difference between the two models lies in the way they are structured.

This paper aims to analyze the effect of the augmentations on the efficiency and the time consumed for model training with all described transformers in multi-class text classification tasks.

Methods and materials.

As data for the research, the dataset [13] was utilized from the HuggingFace portal [14]. Only a part of the dataset, or 5000 records, was used to conduct experiments with the same data across all the models. Only the labeled part was utilized for the experiment. To have data evenly distributed across the classes, the original dataset was shuffled and then reassembled for future augmentations.

Augmented datasets were created based on the original dataset, with the ratio of one original record to three augmented. In this research, word-level and sentence-level augmentations were considered. Augmentations were done with the nlpaug library [15].

Utilized word-level augmenters:

- synonym augmenter: leverage semantic meaning to substitute a word to the synonym;
- antonym augmenter: leverage semantic meaning to substitute a word to the antonym;
- contextual word embeddings augmenter: leverage contextual word embeddings to find a specified amount of top similar words for augmentation.

Utilized sentence-level augmentations:

- lambada augmenter - apply an operation to textual input based on abstractive summarization with the utilization of the lambada method;
- abst_sum augmenter - apply an operation to textual input based on abstractive summarization.

The experiments were conducted on the NVIDIA GeForce GTX 1080 Ti GPU. All used models and tokenizers are available on the HiggingFace portal and are accessible by name:

- bert-base-uncased;
- distilbert-base-uncased;
- xlm-roberta-base;
- albert-base-v2.

For batch formation, the DataColator class from the HuggingFace portal was used. Trainer and TrainingArgumets classes from the same portal were utilized for easier feature-complete training. All classes are available in the 'transformers' package and all experiments were conducted with the exploitation of the PyTorch framework.

Initially, all models were trained in 3 epochs, with a static value of training and evaluation batch size of 8, weight decay was set to 0.01, learning rate $1e^{-4}$, and evaluation and save strategies were set to epoch.

To estimate the effect of data augmentations, different transformer models were trained with all four datasets. Evaluation of models' performance was done by different metrics: validation and training losses, accuracy, precession, F1-score, and recall.

Results and discussion.

In Table 1 we can see the time consumed by different augmentations while modifying the original dataset. Overall it is noticeable that word-level augmentations take less time than word-level. It is possible to categorize obtained results into three groups by consumed time:

- the fastest augmentations: this group contains synonym, antonym, and a combination of synonym and antonym augmentations. These methods took the least time among all the augmentations with significant differences with other groups at least 25 times;
- the medium augmentations: to this group, we can refer contextual word embeddings augmentation with all combinations and the lambada method. Consumed time in this group is around 2200 seconds with an amplitude of nearly 250 seconds;
- the slowest augmentations: to this group belongs abst_sum augmentation and the combination of abst_sum and lambada techniques. The consumed time of this group was nearly 7 times longer than any augmentation from the medium group.

Table 2 demonstrates the time consumed by every model to be trained and evaluated on datasets, modified with different augmentations, both word- and sentence-leveled. Firstly, we will discuss the training time with word-level modified datasets. As we can see, the time for models to be trained on datasets without enhancement is significantly smaller than for augmented datasets. It is because of the size differences in those datasets, as after augmentation initial dataset's size of 5000 records was increased to 20000 records.

Table 1. Dataset's processing time with different augmentations.

Augmentations	**Dataset**	
	Train	Test
	Time, s.ms	
Word-level augmentations		
synonym	41.79	39.82
antonym	48.17	48.92
contextual word embeddings	2143.24	2119.08
synonym + antonym	81.27	80.76
synonym + contextual word embeddings	2332.39	2327.75
antonym + contextual word embeddings	2194.89	2173.19
synonym + antonym + contextual word embeddings	2475.28	2460.61
Sentence-level augmentations		
abst_sum	15955.55	15765.23
lambada	2323.03	2483.48
abst_sum + lambada	17533.63	17735.32

For the models, trained with processed data, we can see that training times are changing in a particular pattern for all models. Training with an antonym-augmented dataset was completed faster for every model, than with synonym-augmented. Then, training with a dataset, modified by contextual word embeddings augmentations was faster, than with antonym-amplified.

Training with a dataset, enhanced with a combination of synonym and antonym augmentations took longer to complete than with any of the single-, double- or triple-augmented by word-level modifications datasets. Training with datasets, modified with a combination of contextual word embeddings and a synonym or antonym augmentations was faster than training with datasets, expanded with a single synonym or antonym, or with a combination of antonym and synonym augmentations. Still, it was slower than training with a single contextual word embeddings-augmented dataset.

The training time required for models to complete with datasets changed with sentence-leveled augmentation, in general, was smaller than the required time for word-level-augmented datasets with one exception. The time consumption for models training with datasets, amplified with abst_sum and the combination of the anst_sum and lambada augmentations was smaller than any other experiment, with single abst_sum-augmented dataset training was the fastest and the mixture was second fastest among all the experiments. At the same time training time for models, trained only with lambada augmentations was the longest among all the experiments.

Table 2. Time of model's training and evaluation with augmented datasets.

Augmentations	**Transformer model**			
	BERT	RoBERTa	DistilBERT	ALBERT
	Time, s.ms			
none	233.54	348.62	126.62	221.27
Word-level augmentations				
synonym	795.56	1201.28	432.92	704.49
antonym	789.63	1186.19	428.62	689.34
contextual word embeddings	760.41	1157.22	414.43	661.99
synonym + antonym	815.53	1213.92	441.63	717.50
synonym + contextual word embeddings	778.64	1162.22	423.12	679.60
antonym + contextual word embeddings	776.77	1160.17	422.28	675.76
synonym + antonym + contextual word embeddings	796.96	1173.09	432.30	697.20
Sentence-level augmentations				
abst_sum	715.52	1110.14	393.94	612.96
lambada	866.89	1273.83	468.48	843.78
abst_sum + lambada	727.54	1118.94	398.89	636.34

The figures in the study are organized in a next way: in each figure 6 plots are present. Two plots on the top of the figure present training and validation losses. The next 4 plots show the evaluation metrics, namely precision, accuracy, recall, and f1-score changes during the training.

Fig. 1 shows the training results of the BERT model. As we can see, with the original data BERT after the third epoch was overfitted as we can see from the decrease in training loss and simultaneous increase in validating losses. It can be seen that experiments with datasets, changed with almost every augmentation, except synonym and lambada, suffered the same issue. From the metrics point of view, only the lambada augmentation allowed to facilitate the model's performance by nearly 5 percent. All other augmentations did not improve the effectiveness of the models. We can see a drop in every metric value with datasets, enhanced in any other way.

Fig. 2 depicts the results of the DistilBERT model training. From the training and validation loss, it is visible that similarly to the BERT model's result, only the dataset modified with the lambada technique allowed the model to solve the overfitting issue. At the same time, most of the augmentations did not improve the model's performance. The model, trained with original data has shown better performance, than models trained on augmented datasets. Meanwhile, the performance of the model, trained on the lambada augmented dataset, was the best, with evaluation metrics after the third epoch of at least 99.5%.

Fig. 3 reveals the results of the RoBERTa model training. As we can see, the initial results of RoBERTA are worse than from any other model. As RoBERTa has the most comprehensive model among all utilized, these results were not expected. Additional experiments were conducted with the RoBERTa model. The learning rate was decreased to $1e^{-5}$, and the batch size was decreased to 4.

The results of training with changed parameters are presented in Fig. 4. The time required for models' training increased by approximately 1.5 times. With these parameters, the RoBERTa model's efficiency was improved and was around 90%. Moreover, the lambada augmentation was able to solve overfitting issues and improve the model's performance to approximately 99.7%.

One of the possible explanations for this behavior is, that simpler models, like BERT, DistilBERT, and ALBERT can't differentiate between the words and their synonyms, so models see them as one semantic entity. Consequently, simple augmentations are more efficient with simpler models. On the other hand, RoBERTa is a much more complicated model, with a greater number of parameters. This model is capable of distinguishing between words and their synonyms or antonyms. In that case, the training should be performed with proper optimization of training parameters.

Fig. 5 presents the results of ALBERT model training. Loss graphs present that the model, trained only with contextual word embeddings and antonym-augmented datasets can be utilized with ALBERT. Still, only word embedding augmentation has allowed models to overcome the overfitting issue. For the model, trained with contextual word embeddings modified dataset an additional training epochs can be considered, as the losses continue to decrease. However, unlike the RoBERTa model, the ALBERT model, which overcame the overfitting showed decent performance, with the evaluation metrics values of the model, trained on an antonym-augmented dataset being around 90%.

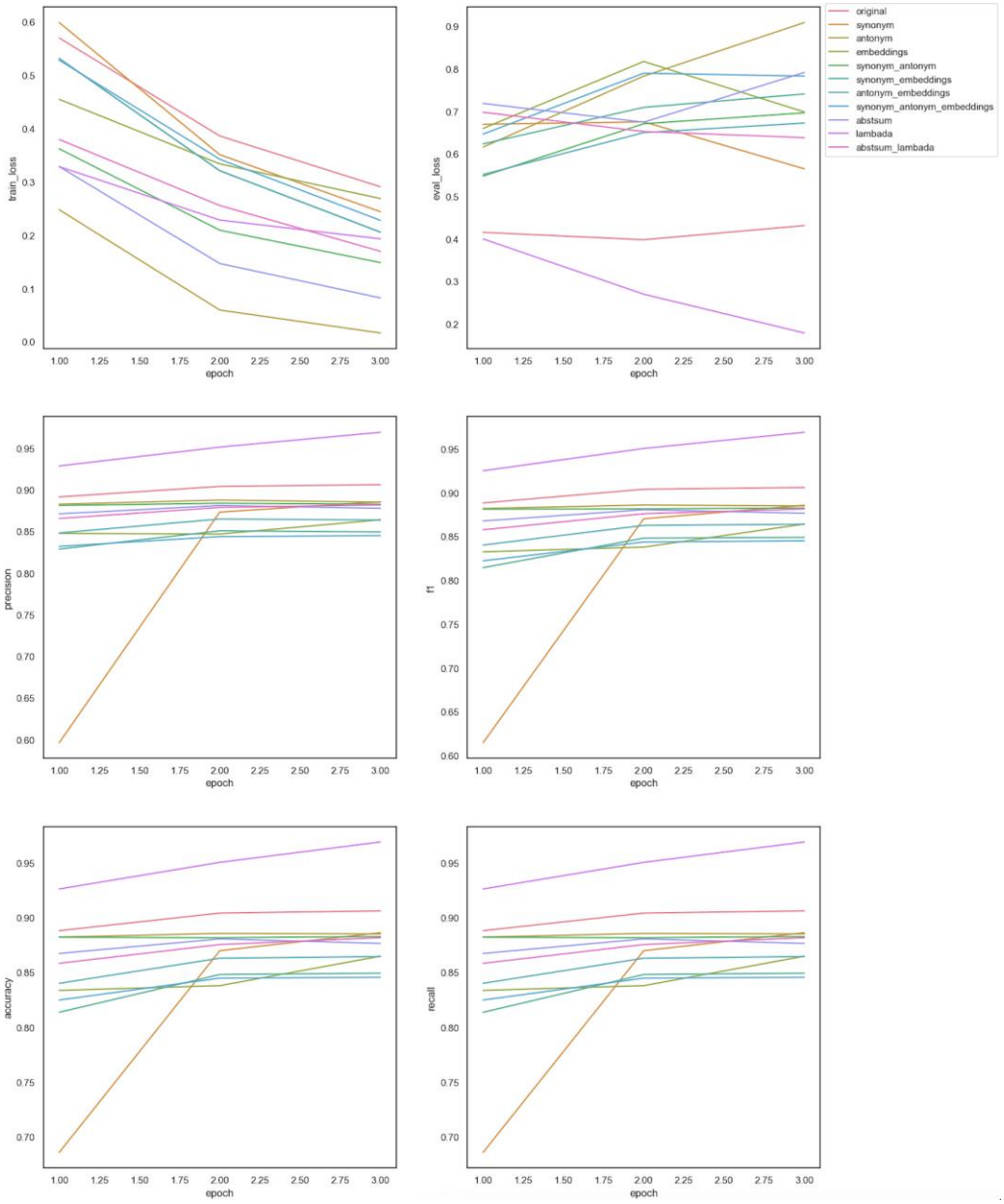

Fig. 1. BERT model's training result (learning rate: $1e^{-4}$, batch size: 8).

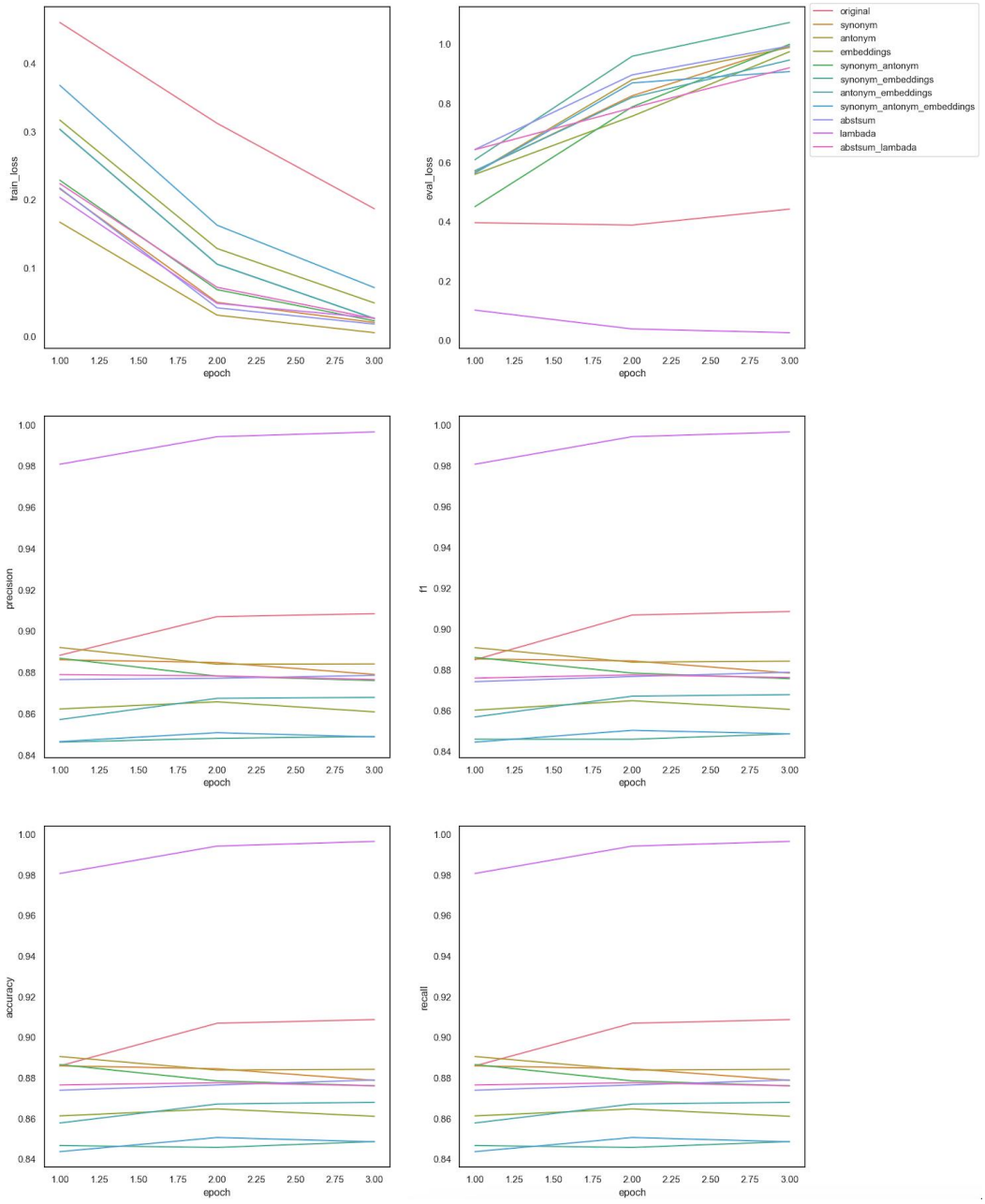

Fig. 2. DistilBERT model's training result (learning rate: $1e^{-4}$, batch size: 8).

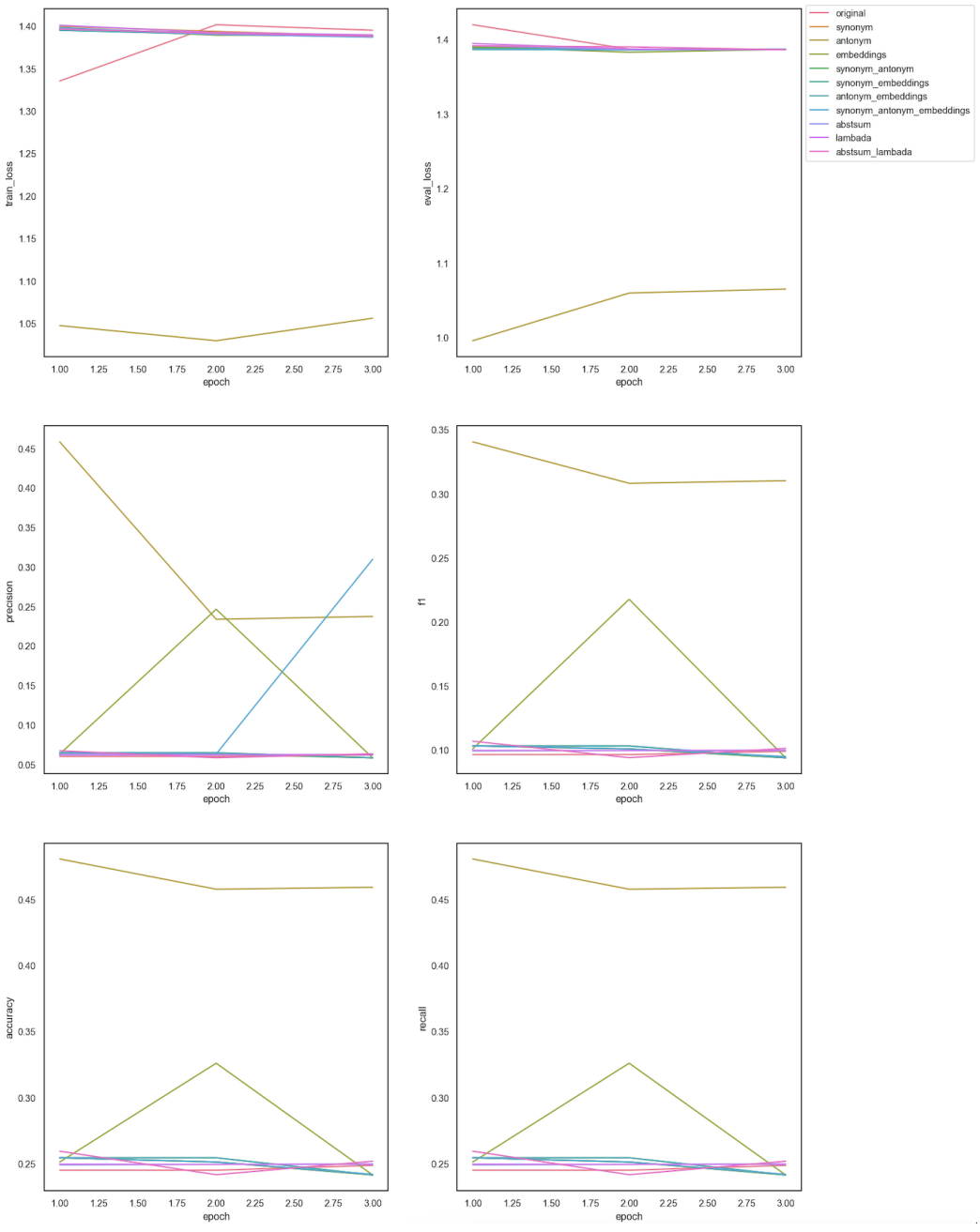

Fig. 3. RoBERTa model's training results (learning rate: $1e^{-4}$, batch size: 8).

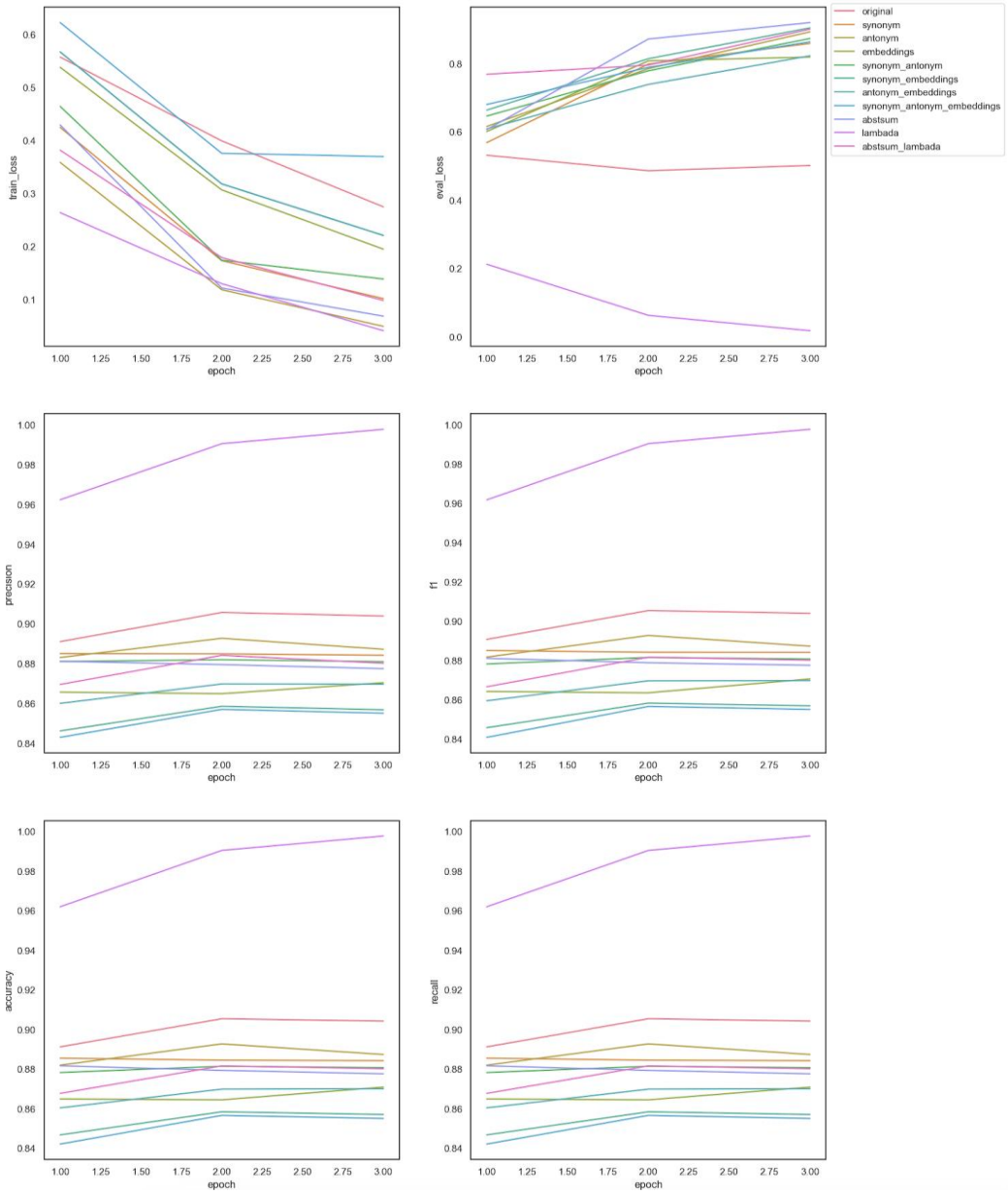

Fig. 4. RoBERTa model's training results (learning rate: $1e^{-5}$, batch size: 4).

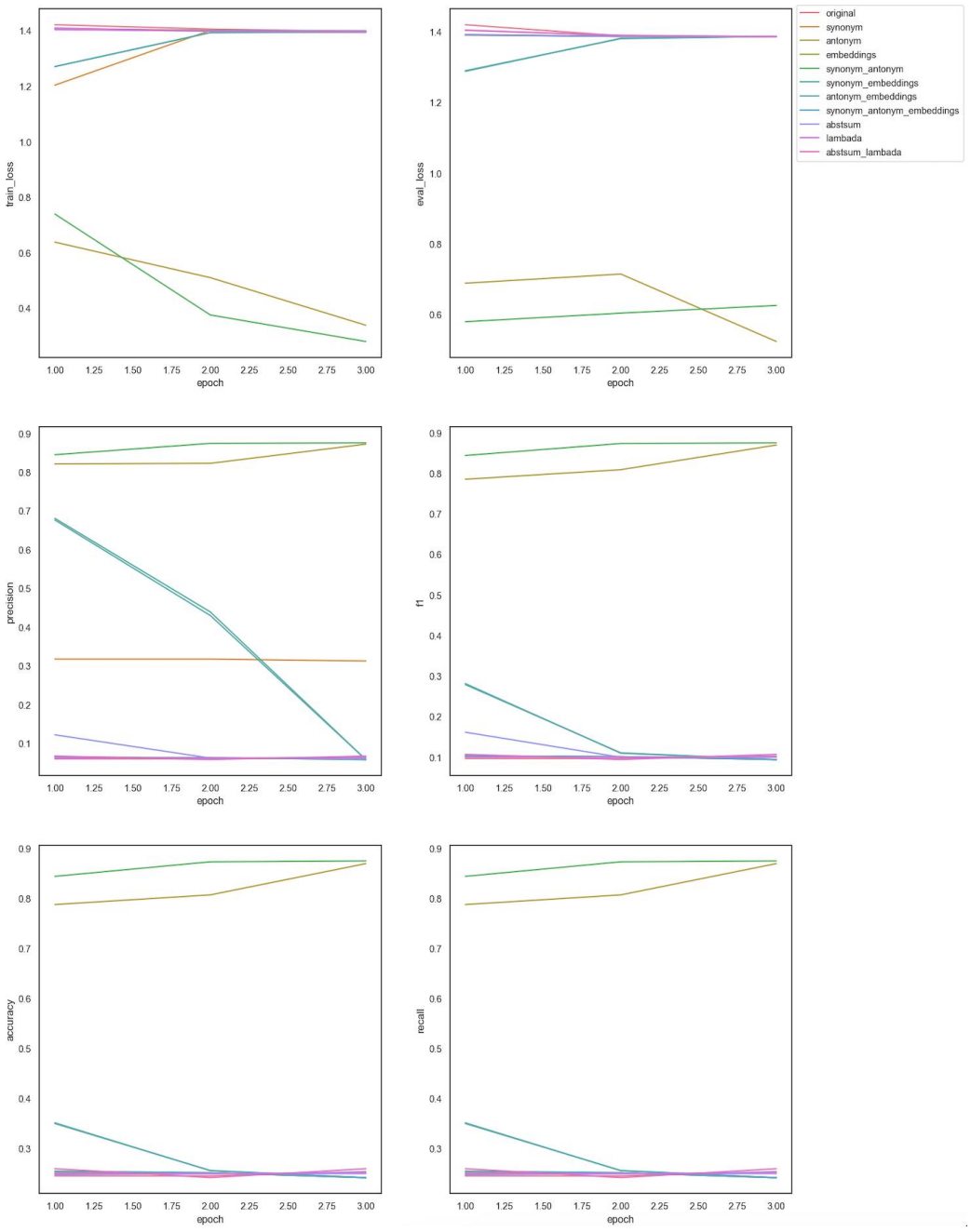

Fig. 5. ALBERT model's training results (learning rate: $1e^{-4}$, batch size: 8).

Conclusion.

This research paper investigates the effect of the word- and sentence-level augmentations and their combination on different transformer models, namely BERT, RoBERTa, ALBERT, and DistilBERT in multi-class classification tasks.

Based on our findings, we can divide the augmentation into three groups by the time consumed for dataset modification. The first group, the fastest to complete contains synonym, antonym, and a combination of synonym and antonym augmentations; the medium group, which was in the middle of time consumption consists of contextual word embeddings augmentation, all combinations with this technique and lambada method; to the slowest group, we can refer the abst_sum augmentation and the combination of abst_sum and lambada techniques.

Our analysis shows that the augmentations influence on the models' training time. Time, consumed by every model training with datasets enhanced with all augmentations changes in a pattern, that repeats itself for every model. Models are trained the fastest with the dataset, augmented with abst_sum augmentation. At the same time, the longest to train took the dataset, modified with a combination of synonym and antonym augmentations.

It was found that most of the augmentations were not proficient enough to allow models to solve the overfitting issue and improve the performance. For the BERT model, only the lambada augmentation had a positive effect on the performance. With the DistilBERT model, the same lambada augmentation showed the best performance, overcoming the overfitting issue. Initially, none of the augmentations allowed the RoBERTa model to solve the overfitting issue nor enhance the performance of the model. However, after proper optimization of the training parameters, performance was improved. Moreover, some augmentations allowed the models to resolve the overfitting issues. The ALBERT model was better than the RoBERTa, but only in connection with antonym and antonym-synonym augmented datasets.

Our findings indicate that during the augmentation selection process, the complicity of the model should be taken into account. For simpler models, simple augmentations, such as with synonyms, may be proficient enough to improve the model's performance. This is because such transformers are not able to distinguish words and their synonyms. On the other hand, more complicated models differentiate these relations between words, which can sometimes lead to models overfitting and worsen the performance.

In the next step, we plan to investigate the augmentation capabilities of different LLMs, namely ChatGPT, Bard, LLaMA, etc. The results of this research may prove valuable for further research on multi-class classification tasks with one of the utilized transformers.

REFERENCES

[1] *Shorten C., Khoshgoftaar T. M., Furht B.* (2021). Text data augmentation for deep learning. *Journal of big Data*, *8*, 1-34.

[2] *Wei J., Zou K.* (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

[3] *Romaine S.* (2007). Preserving endangered languages. *Language and Linguistics Compass*, *1*(1-2), 115-132.

[4] *Magueresse A., Carles V., Heetderks E*. (2020). Low-resource languages: A review of past work and future challenges. *arXiv preprint arXiv:2006.07264*.

[5] *Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Polosukhin I.* (2017). Attention is all you need. Advances in neural information processing systems, 30.

[6] *Pavlyshenko B. M.* (2023). Analysis of Disinformation and Fake News Detection Using Fine-Tuned Large Language Model. arXiv preprint arXiv:2309.04704.

[7] *Pavlyshenko B. M.* (2023). Financial News Analytics Using Fine-Tuned Llama 2 GPT Model. arXiv preprint arXiv:2308.13032.

[8] *Pavlyshenko B. M*. (2022). Methods of Informational Trends Analytics and Fake News Detection on Twitter. arXiv preprint arXiv:2204.04891.

[9] *Devlin J., Chang M. W., Lee K., Toutanova K.* (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[10] *Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Stoyanov V.* (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[11] *Sanh V., Debut L., Chaumond J., Wolf T.* (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

[12] *Lan Z., Chen M., Goodman S., Gimpel, K, Sharma P., Soricut R.* (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

[13] *Zhang X., Zhao J., LeCun Y.* (2015). Character-level convolutional networks for text classification. *Advances in neural information processing systems, 28*.

[14] HuggingFace [Electronic resource]. Access mode: https://huggingface.co/

[15] *Ma E.* (2019). Nlp augmentation.

АУГМЕНТАЦІЯ ДАНИХ У КЛАСИФІКАЦІЇ ТЕКСТУ З КІЛЬКОМА КАТЕГОРІЯМИ

Б. Павлишенко, М. Стасюк

Кафедра системного проектування,
Львівський національний університет імені Івана Франка,
вул. Драгоманова 50, 79005 Львів, Україна
bohdan.pavlyshenko@lnu.edu.ua,
mykola.stasiuk@lnu.edu.ua

У сучасному світі кількість текстових даних, яка генерується кожного дня є надзвичайно великою. Однак, через різницю у використанні різних мов у повсякденному житті, кількість даних, згенерованих англійською, є набагато більшою, ніж, наприклад, українською. Більше того, є велика кількість мов, які можуть зникнути у близькому майбутньому. Через це, з'являється необхідність у методах та технологіях, які дозволяють зберегти вимираючі мови та зробляють ефективним використання цих мов у підходах машинного навчання. Одним з розроблених підходів для створення нових даних на основі вже існуючих є аугментація.

Мета цієї статті дослідити вплив аугментації даних на завдання багатокласової текстової класифікації, яке виконується різними моделями трансформерів: BERT, DistilBERT, ALBERT, and XLM-RoBERTa. Дані для тренування та тестування моделей

отримано з порталу HuggingFace. Зокрема, записи з даних модифікувалися різними аугментаційними техніками, а саме: на рівні слів використані аугментації антонімами, синонімами і контекстуальними вбудуваннями слів; на рівні речень використано аугментації абстрактного формування висновку та ламбада. Замість прямого тренування та оцінки, використано тренувальну інфраструктуру, яку надає портал HuggingFace. Розглянуто різні метрики ефективності моделей, такі як: точність, влучність, f1-оцінка та відклик.

Результати роботи дозволяють порівнювати ефективність роботи кожної розглянутої моделі у завданні багатокласової класифікації. Разом з тим, оцінено ефективність різних аугментацій текстових даних. Це є важливим у процесі визначення об'єднання трансофрмерної моделі та аугментації, яке дозволяє отримати найкращі результати у завдані класифікації тексту з множиною категорій.

Ключові слова: аугментація, багатокласова класифікація тексту, BERT, ALBERT, DistilBERT, XLM-RoBERTa.

Стаття надійшла до редакції 16.12.2023.
Прийнята до друку 23.02.2024.