Ukrainian News Corpus As Text Classification Benchmark*

Dmytro Panchenko[1] [0000−0001−5454−5661], Daniil Maksymenko[1] [0000−0003−3223−5130], Olena Turuta[1] [0000−0002−1089−3055], Mykyta Luzan[1] [0000−0001−9150−7971], Stepan Tytarenko[1] [0000−0002−2821−1518], and Oleksii Turuta[1] [0000−0002−0970−8617]

Kharkiv National Univesitsity of Radio Electronics, Kharkiv, Ukraine
`oleksii.turuta@nure.ua`

Abstract. One of the crucial problems of natural language processing for languages such as Ukrainian is lack of datasets both labeled (for pretraining of word embeddings or large deep learning models) and unlabeled (for benchmarking existing approaches).

In this paper we describe a framework for simple classification dataset creation with minimal labeling effort. We create a dataset for Ukrainian news classification and compare several pretrained models for Ukrainian language in different training settings.

We show that ukr-RoBERTa, ukr-ELECTRA and XLM-R tend to show the highest performance, although XLM-R tends to perform better on longer texts, while ukr-RoBERTa performs substantially better on shorter sequences.

We publish this dataset on Kaggle (https://www.kaggle.com/c/ukrainian-news-classification/) and suggest to use it for further comparison of approaches for Ukrainian text classification.

Keywords: Ukrainian language processing · Text classification · Transformer models · Text dataset.

1 Introduction

Recently, natural language processing went through the phase of rapid development similar to the computer vision revolution in 2010s. This progress can be mainly attributed to the development of Transformer [2] and BERT architectures [3]. Such success has been possible to achieve only because of the transfer learning mechanism.

However, unsupervised pretraining of such models requires a lot of data and computational power. As a result, most of the top pretrained architectures only exist for the most popular languages such as English, Chinese, etc. Only a few of such models exist for Ukrainian language.

[*] Results of the "Ukrainian News Classification" contest [1] hosted by TechTalents

The most prominent out of them are ukr-RoBERTa [4] and ukr-ELECTRA [5]. However, both of the models lack proper evaluation: ukr-ELECTRA is benchmarked on POS tagging and NER tasks only, while ukr-RoBERTa does not have any metrics calculated on public datasets at all.

This lack of diverse evaluation that is usually present in scientific papers on NLP for English language is a result of insufficient amount of publicly available, properly arranged and cleaned datasets for Ukrainian language.

There were several attempts to create a benchmark dataset for the most common and straightforward NLP task – text classification.

In [6] [7] authors create a hotel review sentiment analysis dataset that can serve as a benchmark for text classifiers in Ukrainian. They also note lack of Ukrainian data and complexity of data collection. In the end, they resolve to augmenting their dataset with Russian texts translated to Ukrainian with machine translation algorithms.

Another notable resource with Ukrainian datasets [8] suggest a vast collection of unlabeled data as well as datasets and pretrained models for NER. However, it lacks Ukrainian datasets for seq2one tasks.

Alternative approach for solving NLP tasks in Ukrainian is to use multilingual models. There are two transformer models that were trained on a variety of languages including Ukrainian: Multilingual BERT [3] and XLM-R [9]. Such models are usually trained on a combined corpus that includes texts in dozens of languages (more specifically, mBERT and XLM-R are trained on the collection of 104 largest Wikipedia datasets in different languages). After that, they are evaluated on the crosslingual benchmarks such as XNLI [10]. However, XNLI does not include Ukrainian language, so these models were not tested on Ukrainian data specifically.

In this paper we suggest to benchmark pretrained models for Ukrainian text classification task. In order to do that, we develop a generic framework that allows us to collect a lot of Ukrainian data relatively easily and without any need for manual annotation.

After that, we apply our methodology to create a dataset for news classification (though it can be applied to several other domains). We use this dataset to evaluate and compare several open-sourced transformers that are available and applicable for Ukrainian language. In the last section, we analyze the results and create recommendations for the potential model selection for similar applied tasks.

We also encourage other researchers in this field to use this dataset in further evaluation of their models in this domain.

2 Dataset

For our purposes we construct a dataset of Ukrainian news scraped from several data sources listed in section 3.1. Data preparation framework is described in section 2.2 below. We use this dataset to benchmark a variety of models for text classification. These models are either trained or fine-tuned to a downstream task formalized in section 2.3.

2.1 Data sources

In order to create a large enough benchmark dataset we need to gather and label a huge number of texts. If we also want to test our models under different conditions (e.g. text style, length, mixture of different languages, etc.), we need to create a separate labeling for each of these settings. Unfortunately, data collection is quite expensive and difficult. Especially for Ukrainian language, since it has a limited amount of data sources, and most of the media sources that are easy to collect (e.g. social network posts, news, etc.) are contaminated with Russian and English pieces.

To tackle this problem, we created a pipeline that allowed us to collect a text classification dataset without any labeling and with reasonably small data preparation efforts. This dataset can be further extended in the same way as it was initially collected if needed.

For this purpose, we scraped several Ukrainian news websites: BBC News Україна [11], НВ [12], Українська правда [13], Економічна правда [14], Європейська правда [15], Українська правда Життя [16] and Уніан [17].

Distribution of the scraped data is shown on figure 1. Complete raw dataset consists of 94994 texts.

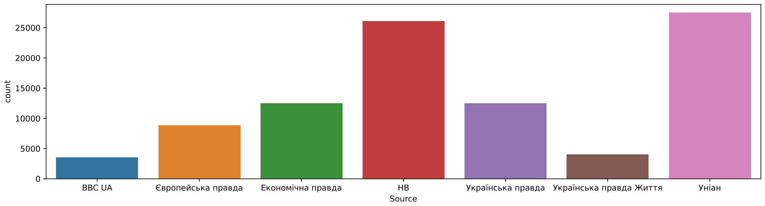


Fig. 1. Distribution of raw data

After that, we used data source as a classification objective. Implications of this choice and hypothesis on what a model can learn from such target variable are considered in section 2.3.

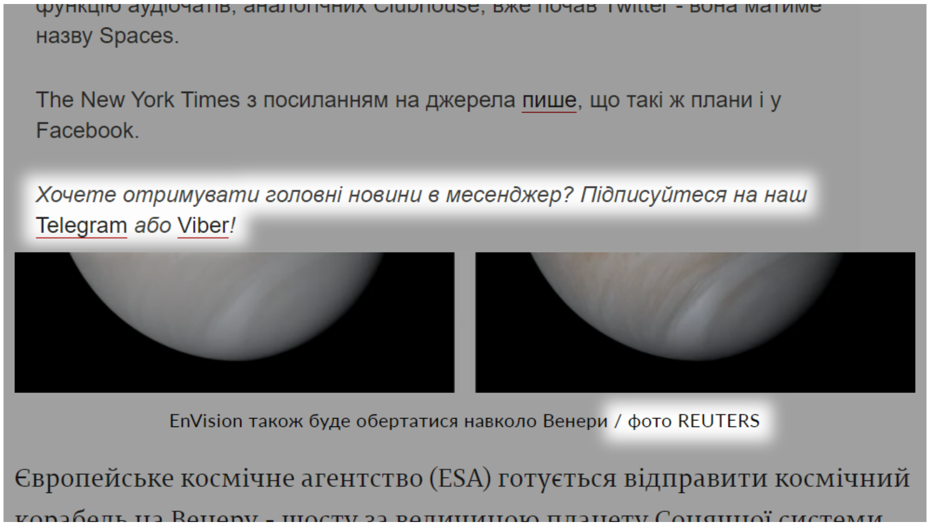
2.2 Data preparation

There are several aspects of the raw dataset that require preprocessing and cleaning. Even a simple bag-of-words [19] based model, trained on this initially obtained dataset, achieves 0.9 F1-score, while more complex deep learning-based approaches show near perfect accuracy after several epochs of training due to numerous implicit data leakages [20] in text.

We apply next data preparation pipeline in order to deal with this issue:

1. *Whitespace normalization.* Leading and trailing whitespace characters were truncated. Sequences of more than one whitespace character were compressed.

2. *Source title removal.* All mentions of any of the source title in any grammatical form (e.g. “BBC”, “Бібісі” or “Служба новин BBC”) were replaced with special token [SOURCE] both in article titles and texts. Modified version of Norvig’s typo corrector [21] was used to deal with incorrect spelling of data source titles.
3. *Duplicate removal.* For each cluster of duplicated or similar texts only one instance was left. The most prominent examples of such clusters are template articles about currency exchange rates or new coronavirus cases in Ukraine, that only differ from each other in numbers and minor details (e.g. list of red zone regions). Obviously, a model could memorize such texts instead of learning their semantics. Thus, all such cases were considered to be data leakages.
4. *Language cleaning.* Language of texts was automatically detected using langdetect [22]. All texts of non-Ukrainian origin were removed from the dataset.
5. *Template data leakages.* We conducted a semi-automated search of typical patterns that only occur in texts from a particular data source thus unequivocally identifying it by form. All such occurrences were removed from the dataset.



функцію аудіочатів, аналогічних Clubhouse, вже почав Twitter - вона матиме назву Spaces.

The New York Times з посиланням на джерела пише, що такі ж плани і у Facebook.

Хочете отримувати головні новини в месенджер? Підписуйтеся на наш Telegram *або* Viber*!*

EnVision також буде обертатися навколо Венери / фото REUTERS

Європейське космічне агентство (ESA) готується відправити космічний корабель на Венеру - шосту за величиною планету Сонячної системи

Fig. 2. Examples of data leakages via template phrases in BBC and Unian articles respectively

In order to do template data leakage search, first, we created a TF-IDF matrix [23] of all terms and the most popular bigrams in the dataset. Then, we used chi-square feature selection to find top-20 tokens for each class. All selected tokens were manually reviewed, and some typical sentences or phrases containing them were identified as data leakage and cleaned. This process was repeated several times, until no suspicious token showed up in the top-20.

The most prominent examples of such template data leakages were clickbait phrases (e.g. “Visit our YouTube channel for more details”) and references (e.g. “Image credits . . . ”). Examples of such leakages with context are shown in figure 2. Each type of template data leakage was either masked with [SOURCE] token or otherwise removed.

Although we aknowledge that such changes alter the natural data distribution, we deem them necessary in order to make this task representative to real-world problems where models need to learn complex semantic relations instead of searching for a set of predefined clues.

After the aforementioned preprocessing, the processed dataset consists of 82554 texts (approximately 12000 texts were completely eliminated due to various reasons).

The dataset is further split into training and test subsets. Complete training set contains 57789 titles and texts. Test set consists of 24765 samples. Subsets have similar target variable distribution.

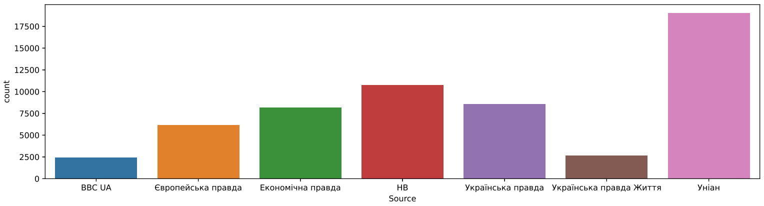


Fig. 3. Training set class distribution

2.3 Task formulation

We use data sources as a classification objective. This way we expect models to learn some kind of a mixture of style classification (since each data source has unique stylistics and textual attributes) and topic modeling (because some of the news websites in our dataset are focused on particular sets of topics, though each major topic is represented by at least several sources).

This multiclass classification problem is evaluated with macro-averaged F1-score:

$$F_1 = \frac{1}{n} \sum_{i=0}^{n} \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i}$$

This way we penalize minor classes on the same scale as major, thus making models struggle with class imbalance.

We also suggest benchmarking each model in several different settings.

First option is to train the models either on the full training set or on the tiny subset consisting of 8256 samples. It allows us to simulate model performance under training data constraints: such a situation might occur in real-world applications when the cost of data labeling is high, e.g. when data labeling requires highly-specialized professionals [18].

Second option that we suggest is to train the models either on full articles or just titles. This way we can determine models' performance on texts of different length.

Distribution of text and title length is shown in figure 4. Hereinafter, texts and titles are also referred to as long and short texts respectively.

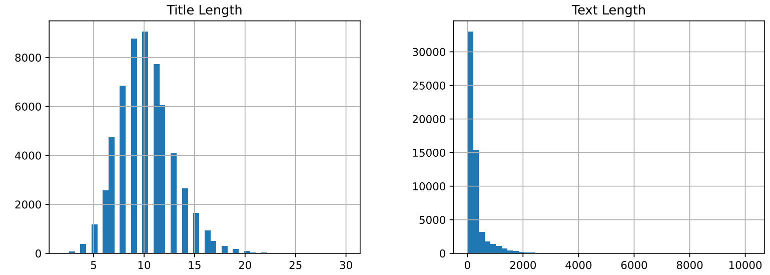


Fig. 4. Distribution of number of words in titles and texts respectively

To sum up, there are a total of four training settings under these two conditions. We test each of the models under each of these settings.

3 Model zoo

We test five different transformer models: Multilingual BERT [3], Slavic BERT [24], ukr-RoBERTa [4], ukr-ELECTRA [5] and XLM-R [9]. Sections 3.1-5 briefly describe these models.

Also, we train a simple but robust baseline - NB-SVM on TF-IDF features implemented according to [25]. Surprisingly its results are comparable to those of transformers in some of the settings. These results are discussed in greater detail in section 5.

3.1 Multilingual BERT

Multilingual BERT (mBERT) is an extension of a classic BERT proposed by [3] that is trained on the combined corpora for 104 languages including

Ukrainian. It can be used for further transfer learning on various downstream tasks including news classification. It usually shows much lower performance than models pretrained for each specific language. For our experiments we specifically use uncased mBERT-base.

3.2 Slavic BERT

Slavic BERT is a result of unsupervised transfer learning from mBERT on the combined corpus of Wikipedia pages written in several Slavic languages. Though that corpus doesn't include Ukrainian, it is possible that pretraining on the same morphemes might improve its performance for the Ukrainian downstream dataset as well.

3.3 ukr-RoBERTa

ukr-RoBERTa is a version of RoBERTa [26] model pretrained specifically on the large-scale corpus consisting of Ukrainian Wikipedia, Ukrainian OSCAR deduplicated dataset [27] and youscan's internal dataset collected from the social networks. Authors do not report any results for this model on public benchmarks, though they mention that they got 2 percent f1-score improvement on their internal datasets comparing to mBERT. Measuring this model's performance on a public dataset is crucial for its effective usage in applied science.

3.4 ukr-ELECTRA

ukr-ELECTRA is an ELECTRA [28] architecture based model pretrained on the Ukrainian Wikipedia pages and Ukrainian OSCAR deduplicated dataset. It is expected that it should perform better than ukr-RoBERTa, since ELECTRA architecture generally outperforms RoBERTa on most of the tasks. However, Ukrainian version of ELECTRA was pretrained on a smaller dataset, so as our experiments show, they compare differently than their English counterparts.

3.5 XLM-R

XLM-R is a RoBERTa-based model that is trained in the same manner as mBERT. XLM-R is the only model that has an open-sourced version of pretrained weights for a large architecture version. For the sake of finding the best text classification model for Ukrainian texts we use this version instead of the base one during our experiments since it is expected to give the top performance for downstream tasks.

4 Experiments

We conduct a set of four experiments for each model:

1. Small training set; training on titles only.
2. Small training set; training on full texts.
3. Large training set; training on titles only.
4. Large training set; training on full texts.

For each of these experiments superficial tuning is performed. We select the learning rate scheduler on a small validation subset selected from the training set. After that, we retrain the model with top hyperparameters set on the whole training set before submitting the prediction.

In order to compare models in realistic conditions, instead of training each model for the same number of training steps, we train each of them in a fixed budget of 24 hours per single P100 GPU.

Benchmark results are shown in the table below:

Model	Short texts / small training set	Long texts / small training set	Short texts / large training set	Long texts / large training set
NB-SVM baseline	0.533	0.790	0.636	0.900
mBERT	0.626	0.853	0.685	0.910
Slavic BERT	0.620	0.840	0.708	0.907
ukr-RoBERTa	**0.675**	0.903	**0.745**	0.940
ukr-ELECTRA	0.623	0.909	0.721	0.948
XLM-R	0.624	**0.915**	0.689	**0.950**

5 Results

The highest score among all conducted experiments is achieved by XLM-R trained on the large version of the full-text training set: it reaches 0.95 F1.

There are a few interesting observations:

1. Both mBERT and Slavic BERT perform rather poorly in terms of F1-score. While it was expected for mBERT, it is somewhat surprising that Slavic BERT did not show any accuracy improvement, performing even worse than mBERT in three settings out of four.
2. ukr-RoBERTa shows huge performance improvement over mBERT (5-6% for short texts and 3-6% for long texts). It also shows a smaller gap between short text and long text settings. We attribute it to the fact that it was trained on the dataset that includes scraped social media posts which generally tend to be shorter than other types of texts.
3. ukr-ELECTRA shows similar metrics, being slightly worse on short texts and slightly better on long texts.
4. XLM-R generally outperforms all models on long texts while having significantly lower performance on short texts. It is worth mentioning that XLM-R has 24 encoder blocks instead of 12, so it has almost 3x memory bandwidth and latency compared to other benchmarked transformer models.

5. Despite expectations, the NB-SVM baseline shows quite high f1-score in the large training set mode. While trained on the large dataset, it is only 7% worse than average transformer model in short text setting and it performs almost on par with mBERT and Slavic BERT in long text setting. We assume that it is due to the fact that when we train models on the small dataset, efficiency of the transfer learning approach is much more significant than in the case of the large dataset.

These results show that ukr-RoBERTa could be a model of choice for short-length texts, while XLM-R or ukr-ELECTRA is the best choice for longer texts depending on the computational budget for inference.

It is worth mentioning that NB-SVM model which requires neither GPU for training nor expensive hardware for real-time inference achieves comparable performance if the training dataset is large enough. It is only 5% below the best model while taking fifteen minutes to implement and train which is acceptable in a lot of applied cases.

6 Conclusion

In the scope of this paper we presented a simple and effective framework that allows us to create a text classification dataset with minimal effort.

Using this approach we created a dataset for news classification that consists of almost 60 thousand training samples and allows benchmarking models in several different settings for deeper understanding of models pros and cons. The dataset is hosted at Kaggle (https://www.kaggle.com/c/ukrainian-news-classification/) and is available for benchmarking of novel machine learning algorithms for Ukrainian language.

We tested several existing open-source models on this dataset and evaluated these models in a fair setting. As a result, we showed that ukr-RoBERTa and ukr-ELECTRA are the top-performing medium-sized models, while XLM-R performs better for long texts if there are no computational constraints.

At the same time, NB-SVM shows comparable results. This observation along with the fact that crosslingual model is one of the top performers means that pretrained transformer models for Ukrainian language still have a long way to go. Collecting larger datasets for the unsupervised pretraining and pretraining of larger models (e.g. RoBERTa-large) seem to be the most promising fields of development.

References

1. https://www.kaggle.com/c/ukrainian-news-classification/
2. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

3. Jacob Devlin, Ming-Wei Chang, Kenton Lee, & Kristina Toutanova. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
4. Vitalii Radchenko. We Trained the Ukrainian Language Model. https://youscan.io/blog/ukrainian-language-model/
5. Stefan Schweter, Ukrainian ELECTRA model https://github.com/stefan-it/ukrainian-electra https://doi.org/10.5281/zenodo.4267880
6. Babenko, Dmytro. Determining sentiment and important properties of Ukrainian-language user reviews : Master Thesis : manuscript rights / Dmytro Babenko ; Supervisor Vsevolod Dyomkin ; Ukrainian Catholic University, Department of Computer Sciences. – Lviv : [s.n.], 2020. – 35 p. : ill.
7. Babenko, D., & Dyomkin, V. (2019). Determining Sentiment and Important Properties of Ukrainian Language User Reviews. http://ceur-ws.org/Vol-2566/MS-AMLV-2019-paper39-p106.pdf
8. NER annotation corpus https://lang.org.ua/en/corpora/
9. Alexis Conneau and Kartikay Khandelwal and Naman Goyal and Vishrav Chaudhary and Guillaume Wenzek and Francisco Guzmán and Edouard Grave and Myle Ott and Luke Zettlemoyer and Veselin Stoyanov (2019). Unsupervised Cross-lingual Representation Learning at Scale. CoRR, abs/1911.02116.
10. Conneau, A., Rinott, R., Lample, G., Williams, A., Bowman, S., Schwenk, H., & Stoyanov, V. (2018). XNLI: Evaluating Cross-lingual Sentence Representations. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics.
11. https://www.bbc.com/ukrainian
12. https://nv.ua/
13. https://www.pravda.com.ua/
14. https://www.epravda.com.ua/
15. https://www.eurointegration.com.ua/
16. https://life.pravda.com.ua/
17. https://www.unian.ua/
18. Shen, Ying et al. "Improving Medical Short Text Classification with Semantic Expansion Using Word-Cluster Embedding." ArXiv abs/1812.01885 (2018): n. pag.
19. Zhang, Yin & Jin, Rong & Zhou, Zhi-Hua. (2010). Understanding bag-of-words model: A statistical framework. International Journal of Machine Learning and Cybernetics. 1. 43-52. https://doi.org/10.1007/s13042-010-0001-0
20. Kaufman, Shachar & Rosset, Saharon & Perlich, Claudia. (2011). Leakage in Data Mining: Formulation, Detection, and Avoidance. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 6. 556-563. https://doi.org/10.1145/2020408.2020496
21. Peter Norvig. How to Write a Spelling Corrector. url: http://norvig.com/spell-correct.html.
22. Shuyo, N. (2010). Language Detection Library for Java.
23. (2011) TF–IDF. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8
24. Arkhipov, A. (2019). Tuning Multilingual Transformers for Language-Specific Named Entity Recognition. In Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing (pp. 89–93). Association for Computational Linguistics.
25. Wang, C. (2012). Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 90–94). Association for Computational Linguistics.

26. Yinhan Liu and Myle Ott and Naman Goyal and Jingfei Du and Mandar Joshi and Danqi Chen and Omer Levy and Mike Lewis and Luke Zettlemoyer and Veselin Stoyanov (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. CoRR, abs/1907.11692.
27. Ortiz Suárez, P., Sagot, B., & Romary, L. (2019). Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In 7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7). Leibniz-Institut für Deutsche Sprache.
28. Kevin Clark, Minh-Thang Luong, Quoc V. Le, & Christopher D. Manning (2020). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In International Conference on Learning Representations.