Introducing UberText 2.0: A Corpus of Modern Ukrainian at Scale

Dmytro Chaplynskyi
Lang-uk
Kyiv, Ukraine
chaplinsky.dmitry@gmail.com

Abstract

This paper addresses the need for massive corpora for a low-resource language and presents the publicly available UberText 2.0 corpus for the Ukrainian language and discusses the methodology of its construction. While the collection and maintenance of such a corpus is more of a data extraction and data engineering task, the corpus itself provides a solid foundation for natural language processing tasks. It can enable the creation of contemporary language models and word embeddings, resulting in a better performance of numerous downstream tasks for the Ukrainian language. In addition, the paper and software developed can be used as a guidance and model solution for other low-resource languages. The resulting corpus is available for download on the project page. It has 3.274 billion tokens, consists of 8.59 million texts and takes up 32 gigabytes of space.

1 Introduction

In this paper, we introduce UberText 2.0, which is the new and extended version of UberText, a corpus of modern Ukrainian texts designed to meet various NLP needs.

Modern development of word embeddings (Bojanowski et al., 2017), transformers (Devlin et al., 2019), neural machine translators (NLLB Team et al., 2022), speech-to-text models (Radford et al., 2022), and question answering systems (Yang et al., 2019) opens new horizons for natural language processing. Most of the models mentioned above rely heavily on the availability of corpora for a target language. While it is not usually a problem to obtain such a dataset for languages such as English, Chinese or Spanish, for low-resource languages, the absence of publicly available corpora is a severe barrier for researchers.

Different approaches can be used to overcome this problem. Researchers might use multilingual transformers to achieve sub-optimal performance for low-resource languages (Rust et al., 2021). Alternatively, they might rely on the publicly available multilingual corpora, such as Wikipedia (Al-Rfou' et al., 2013), Common Crawl (Grave et al., 2018), or Oscar (Srinath et al., 2021), or collect their own corpus using web crawling technologies. The latter approach requires a lot of data engineering to combat noisy data and extract relevant texts in the target language. While many authors follow this path, it shifts the attention from the target task, requires specific skills, and takes time to collect the data rather than make use of it.

To enable researchers to work on large language models or perform a data mining on texts, we release a high-quality corpus for the Ukrainian language at scale and a model solution that can be applied to other low-resource languages.

The core concept of our corpus is that the same data, once collected and processed, can be later used to produce various deliverables suitable for different computational linguistics tasks. The corpus size, the additional layers (like POS tags and lemmas), and its availability for direct download make it an invaluable dataset. At the same time, the data model behind it and its flexible architecture allows exporting the corpus version pinpointed to a particular task or research need.

The pipeline behind the corpus simplifies data collection, pre- and post-processing, and export of the deliverables, helping set up a regular release cycle so that end users can use the fresh copy of the data or update their models built on the previous versions when needed. Such deliverables can include:

- Raw texts with markup and complete metadata
- Cleansed and filtered texts
- Tokenized version of the corpus (with or without punctuation)
- Lemmatized version of the corpus
- Lemma frequencies, n-grams, and other lists

Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP), pages 1–10
May 5, 2023 ©2023 Association for Computational Linguistics

as well as other deliverables or subcorpora, obtained by filtering original texts based on such metadata as date, author, or source.

2 Background

The Ukrainian language is a morphologically rich language of the synthetic type, spoken by more than 40 million people. Historically, it took shape in different centers, which influenced modern Ukrainian as we know it. While it is one of the most widely spoken Slavic languages, it can still be considered a low-resource language and is underrepresented in modern NLP research. The reason is the lack of publicly available corpora tailored to different needs. It can be speculated with a high degree of confidence that similar issues exist in other languages. We want to address this gap for the Ukrainian language and propose a model solution that can be reused for other languages.

Existing corpora are scattered across quite a wide range. On one end, we have relatively small, well-balanced corpora such as Brown (Francis and Kucera, 1979), BRUK (Starko et al., 2016-2023), or any national corpus collected by a dedicated team. On the other end, we have gigantic corpora, such as OSCAR (Abadji et al., 2022) and Common Crawl [1], which have been collected fully automatically. In between these two extremes, there are many corpus projects that may be used either as the main data source or as supplementary material, depending on the task at hand.

In our opinion, each corpus should have a clear contract with the end user that specifies the guarantees and promises it fulfills, the availability of the data, the functionality offered on top of the data (e.g., a corpus manager or extra layers), frequency of updates, and the methodology behind the data collection and processing. This will allow the researcher to pick the right tool for the job and understand the limits of this tool. To meet the requirements of modern computational linguistics, we establish the following contract for the corpus:

- Massive
- Freely available for download under a permissive license
- Built from modern language data and sufficiently representative
- Maintains a decent level of text quality and internal quality control procedures
- Has additional layers, e.g., lemmatization, POS tags, et cetera. This approach allows for various corpus mining tasks, building the lemma frequency dictionaries by POS tags.

3 Related work

Most existing corpora for the Ukrainian language do not meet all the criteria outlined above, particularly when it comes to the scale and availability of the data for direct download.

Corpora unavailable for download:

- Zvidusil created by Kotsyba et al. (2018) corpus contains 2.8 billion tokens collected primarily in an automated fashion. The last update to the corpus was made in 2017.
- General Regionally Annotated Corpus of Ukrainian (GRAC-16) collected by team of Shvedova et al. (2017-2023) has almost 1.9 billion tokens, is updated twice a year, and has extensive meta-information on the texts.
- The Ukrainian Text Corpus (KUM) by Darchuk (2017) contains about 120 million tokens and is only accessible through a limited corpus manager.
- The Ukrainian Web Corpus of Leipzig University [2] only provides samples of up to 1 million words.
- The Corpus of the Chtyvo Library [3] contains 6.6GB of OCRed texts of mediocre quality.
- Araneum Ucrainicum Beta, corpus by Benko (2014) has around 5,249 million of tokens, only available for the registered users through the corpus manager [4]
- ukTenTen: Ukrainian corpus from the Web has about 3,280 million of tokens, available for subscribed users through a corpus manager[5]

Corpora available for download of smaller size:

- Brown-UK by Starko et al. (2016-2023), a well-balanced national high-quality corpus, is available for download, with around one million words.
- UberText 1.0[6], is the previous version of the corpus presented in this paper. It has around 665 million tokens, and consists of shuffled

[1] https://commoncrawl.org

[2] http://corpora.informatik.uni-leipzig.de/

[3] http://korpus.org.ua/

[4] http://aranea.juls.savba.sk/guest/

[5] https://www.sketchengine.eu/uktenten-ukrainian-corpus/

[6] https://lang.org.ua/en/corpora/

sentences. UberText 1.0 wasn't updated since 2016.

4 The Corpus

To address the issues of availability and scale and allow researchers to train large language models for Ukrainian, we release a new version of UberText. The new version shares some sources and texts with UberText 1.0, but all of them were re-crawled and pre-processed.

The total size of the corpus after post-processing and filtering is:

- 8,592,389 texts
- 156,053,481 sentences
- 2,489,454,148 tokens
- 32 gigabytes of text

In addition to releasing texts, we have developed and open-sourced a software solution[7] that helps manage the data sources and update the corpus database, perform quality assurance tasks, calculate statistics, pre- and post-process texts, and export data in various formats.

4.1 Corpus composition

UberText 2.0 has five subcorpora:

- *news* (short news, longer articles, interviews, opinions, and blogs) scraped from 38 central, regional, and industry-specific news websites;
- *fiction* (novels, prose, and some poetry) scraped from two public libraries;
- *social* (264 public telegram channels), acquired from the project TGSearch;
- *wikipedia* — the Ukrainian Wikipedia as of January 2023;
- *court* (decisions of the Supreme Court of Ukraine), received upon request for public information.

Table 1 presents statistical information on the subcorpora.

All the entries of the corpus are stored as separate documents in a document-oriented database and have a title (where possible), the text itself, and meta-information: author, source or publisher, URL of the original article or text, main picture, date of publication, tags or categories of the text, and more. Some subcorpora have additional meta-fields specific to the domain, e.g., court decisions have information on the judge and the geographic region.

The original texts' markup (headers of various levels, ordered and unordered lists, emphases, etc) is preserved where possible by converting the HTML of the article to the markdown format using html2text library[8]. Markdown allows keeping some structure of the text (for example, headers and subheaders). Also, it is human-readable and can be easily stripped afterward with the help of Markdown library[9].

4.2 Data collection

UberText 2.0 utilizes the Scrapy framework and ecosystem to crawl texts from the web. A dedicated spider is written for each source to capture only the text of an article and meta-information about it but not the boilerplate of the webpage. Extra effort is made to exclude repetitive elements from the article texts, like "subscribe to our social networks" or "also read" calls to action, during the crawling stage.

Such subcorpora as *court*, *wikipedia*, and *social* are also collected using the Scrapy spiders to keep things consistent and manageable even though their data is obtained or downloaded in machine-readable formats in bulk. A custom fork of a gensim's Wikipedia reader was created[10] for better parsing the Ukrainian Wikipedia dump, primarily to deal with accented characters and to process Wikipedia section names in Ukrainian correctly.

The Wikipedia dump was downloaded from the Wikimedia download page[11]; a dump of public telegram channels was received from the TgSearch[12] project; court decisions were obtained from "Court on the Palm" project[13], in the RTF format with a CSV index. Court decisions were initially published by the State Judicial Administration of Ukraine on the National Open Data Portal[14]. Figure 2 demonstrates the manager of the spiders used in the project.

4.3 Data model

MongoDB[15] was selected to efficiently store the massive number of texts together with numerous

[7] https://github.com/lang-uk/lang.org.ua/tree/master/languk/corpus

[8] https://github.com/Alir3z4/html2text/

[9] https://python-markdown.github.io

[10] https://gist.github.com/dchaplinsky/f7bf86837837778f75b704ef57e3811c

[11] https://dumps.wikimedia.org/backup-index.html

[12] https://tgsearch.com.ua

[13] https://conp.com.ua

[14] https://data.gov.ua/organization/derzhavna-sudova-administratsiia-ukrayiny

[15] https://www.mongodb.com

Figure 1: Data flow diagram and processing pipeline

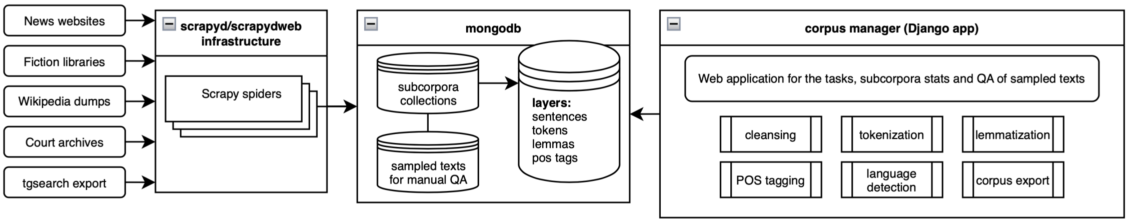

News websites
Fiction libraries
Wikipedia dumps
Court archives
tgsearch export
scrapyd/scrapydweb infrastructure
Scrapy spiders
mongodb
subcorpora collections
sampled texts for manual QA
layers: sentences tokens lemmas pos tags
corpus manager (Django app)
Web application for the tasks, subcorpora stats and QA of sampled texts
cleansing
tokenization
lemmatization
POS tagging
language detection
corpus export

Table 1: Subcorpora of UberText 2.0 with time spans and additional statistics on the number of articles and tokens. The number of texts and tokens are measured before filtering, except when explicitly stated otherwise

Subcorpora	**time span**	**# of sources**	**# of texts**	**# of tokens**
news	2000-2023	38	7,208,299	2,172,526,177
fiction	n/a	2	23,796	253,321,894
court	2007-2021	1	111,658	285,252,442
wikipedia	2004-2023	1	2,819,395	499,603,082
social	2018-2022	264	885,314	63,472,353
total	-	-	11,048,462	3,274,175,948
total after filtering	-	-	8,592,389	2,489,454,148

meta-fields. Also, MongoDB has native support for efficient data compression algorithms, which helps reduce storage requirements and make the whole system scalable.

Each subcorpus has its collection, and the schemaless nature of MongoDB allows for different sets of meta-fields.

Separate collections are created to store the additional layers (such as the normalized, tokenized, and lemmatized versions of the texts and information on the UD POS tags and features) for each text. Figure 1 demonstrates the general architecture of the system and the data model.

A data model like this enables the "collect and process once/reuse many times" concept. It also makes it possible to release incremental updates to the corpus for which only the newly added texts need to be processed.

4.4 Pre-processing

Once a new batch of texts is collected and added to the corpus database, the corpus editor can launch a set of pre-processing jobs:

- Markdown removal and normalization of the texts (unification of hyphens, apostrophes, and Ukrainian diacritics, fixes for encoding issues and word wraps, etc.)
- Language detection
- Segmentation into sentences
- Tokenization (with preserved punctuation)
- Lemmatization
- POS tagging.

The results of these jobs are saved to the corresponding layers and linked to the original texts. Markdown removal is accomplished with the help of the markdown python library[16]. Normalization, sentence segmentation, tokenization and lemmatization are covered by the nlp-uk-api wrapper[17] over nlp-uk[18] groovy library.

Language detection is performed by CLD3 library[19] to allow filtering out non-Ukrainian texts at later stages.

Finally, POS-tagging is done with a fork of the UDPipe[20] tuned for Ukrainian and the corresponding model[21]. Since the tag and the features for one word of text are much longer than the word itself, the tagging results are converted into a more compact textual format to reduce the storage requirements[22].

[16] https://python-markdown.github.io

[17] https://github.com/arysin/nlp_uk_api

[18] https://github.com/brown-uk/nlp_uk

[19] https://github.com/google/cld3

[20] https://github.com/mova-institute/udpipe

[21] mova.institute analyzer

[22] https://github.com/lang-uk/lang.org.ua/blob/master/languk/corpus/ud_converter.py

4.5 Post-processing and export

Once all texts in the corpus are processed and results are stored in the corresponding layers, the corpus editor can initiate the export of deliverables. The post-processing is being done during the execution of the export job and might include the following:

- Filtering by the subcorpora, individual source of the text, or any other filtering over the meta-information. For example, we might export only the texts published over the last two years.
- Additional filtering by the detected language of the text and/or its length. Some texts (especially from Wikipedia) might be too short or unfinished, and some (especially from news websites) might be in Russian or English. To improve the quality of the exported corpus, we usually filter by the combined text length of the title and text (> 100 characters) and only accept the texts where CLD3 is confident in the language.
- Selection of the layer and transformation to a desired format. Some tasks might require tokenized texts with no markup and no punctuation, split by sentence. Some can benefit from the unaltered texts with the markup. Some require unique sentences only or lemmatized texts.
- Compression of the output stream (bzip2 or lzma2).

Figure 5 reflects the corpus export settings available.

Finally, there is a separate class of export tasks: frequency dictionaries built on n-grams of tokens or lemmas. These require additional calculation during the export and rely on the pre-computed layers. Figure 6 shows the settings available for the frequency dictionary export task.

The existing architecture of the corpus software allows for adding more layers, filters, and output formats without the need to rebuild the whole corpus. That helps deliver massive amounts of data tailored to particular research needs in a very short time. For example, the complete export of all subcorpora currently takes around 24 hours on a very modest hardware.

4.6 Data quality

Maintaining the desired quality of the data in a massive corpus is hard, especially when it is collected from sources the corpus editors do not control. Of course, the amount of data collected can smooth some issues. Still, extra measures can be applied to improve the quality of data. In UberText 2.0, we use the following:

- Texts are collected using custom spiders written for each data source. That allows us to filter out boilerplate texts of webpages or overused fragments like "join us on Patreon." with the help of handcrafted CSS and XPATH selectors. In the case of the *social* subcorpus, we apply additional filtering to exclude Telegram channels that are only posted in Russian or considered to be propaganda by the media-monitoring organizations [23].
- When the text source crawling is complete, the spider automatically samples texts, including the oldest, the shortest, and the longest ones, texts with no title or body, and a random sample. Later volunteers manually review those sampled texts and report the issues found to the GitHub repository. Figures 3 and 4 show the stats of the data sources and available text samples under each source.
- The developer of the spider additionally verifies that the spider works correctly before starting a major update of the corpus. This helps account for design or page structure changes.
- During the post-processing stage, texts that are not in Ukrainian or are too short are dropped.

4.7 Release cycle

When we created UberText 1.0, it took much manual labor to prepare the initial deliverables. The old corpus architecture did not allow for quick updates of the texts from the sources or the export of texts into a different format. Therefore, the work on the new corpus version started with the architecture and pipeline revamp. With these changes, we can update the corpus database and the list of deliverables quickly. We aim for the annual update of the corpus and its deliverables. This way, the end-users might refer to a particular version of the corpus to make their research reproducible. New deliverables may be added between the releases to fulfill particular research needs.

We also plan to add more data sources, for example, websites and social media, to keep up with the quickly changing vocabulary of the Ukrainian

[23] Detector Media

language. This will help to increase the size of the corpus and capture the effect of historical changes on the Ukrainian language.

5 Intended usage and cooperation

We successfully used developers' preview of the corpus in various tasks:

- building the first flair embeddings (Akbik et al., 2018) of the Ukrainian language[24] and training compact downstream models like POS[25] and NER[26] on these embeddings;
- training fastText vectors of a high quality (Romanyshyn et al., 2023);
- training lean language models for a Ukrainian speech-to-text project[27];
- training models for punctuation restoration[28];
- training GPT-2 models of different sizes for the Ukrainian language and fine-tuning for various tasks using instructions (Kyrylov and Chaplynskyi, 2023);
- fine-tuning *paraphrase-multilingual-mpnet-base-v2* sentence transformer on the sentences mined from the corpus to achieve better performance on WSD task (Laba et al., 2023).

We cooperate with teams of researchers to train transformer models like GPT-2 proposed by Radford et al. (2019), BERT by Devlin et al. (2019), RoBERTa by Liu et al. (2019) and ELECTRA by Clark et al. (2020) and are open to further collaborations.

We also share the texts of the corpora with the GRAC project[29] to improve the coverage of this vital corpus and make modern texts accessible to linguists, translators, and students through a user-friendly corpus manager[30].

6 Conclusions and Future Work

To build a massive corpus of high-quality texts for a low-resource language, researchers must have a clear contract of what the corpus guarantees and does not guarantee, a methodology, data sources, and a clear pipeline. Proper pipeline implementation will allow for updating the corpus and its deliverables with minimum manual labor. While implementation of such a pipeline and required infrastructure is more related to data engineering and programming rather than to NLP, the impact on the natural language processing for a target language can be enormous. When collected and made available, a good corpus is a solid foundation for myriads of computational linguistics tasks, multiplying the impact on the industry.

Corpora for low-resource languages can also be included in the datasets used to train multilingual word embedding models, such as XLM-RoBERTa proposed by Conneau et al. (2020).

To continue the effort made for UberText, we are planning to:

- set up a regular annual release cycle for UberText;
- collaborate with more researchers, contributing the corpus for various NLP tasks for the Ukrainian language;
- train and release modern word embeddings and models for downstream tasks.

Limitations

When working on the corpus and the software pipeline, we found some obstacles that might affect the reproducibility of the results for other low-resource languages. While the software created is available for reuse under a permissive license, it relies on other programming components, which might not be available for the target language. For example, text segmentation, tokenization, and lemmatization might be very language-specific. We use the nlp-uk package, which wraps the LanguageTool library[31]. A similar wrapper should be developed or integrated for languages other than Ukrainian. The same applies to the UDPipe library[32] and the model used for automatic POS tagging. Other solutions, like SpaCy[33], can be integrated instead. Also, as mentioned above, creating and maintaining a corpus of such scale requires additional knowledge in data retrieval and data engineering.

Ethics Statement

Our paper aims to bring greater visibility to the Ukrainian research community and foster connections within the ACL community. Furthermore, we

[24] https://huggingface.co/lang-uk/flair-uk-forward

[25] https://huggingface.co/lang-uk/flair-uk-pos

[26] https://huggingface.co/lang-uk/flair-uk-ner

[27] https://huggingface.co/Yehor/kenlm-ukrainian

[28] https://huggingface.co/dchaplinsky/punctuation_uk_bert

[29] http://uacorpus.org/Kyiv/en/

[30] https://parasol.vmguest.uni-jena.de/grac_crystal/#dashboard?corpname=grac16

[31] https://languagetool.org

[32] https://ufal.mff.cuni.cz/udpipe

[33] https://spacy.io

acknowledge the potential broader impact of our research on other low-resource languages and believe that our ideas, methodology, and open-source code are applicable and could be utilized to benefit other languages and communities. We recognize the scarcity of academic papers in the ACL Anthology related to the Ukrainian language or produced by Ukrainian researchers.

We take copyright concerns seriously and have made every effort to ensure that the collection of the texts for our corpus does not violate the law. The texts were collected from various web resources and we have preserved their authorship whenever possible. We believe that our use of these texts falls within the bounds of fair use and Ukrainian copyright law, which specifies that certain objects are not protected by copyright. For example, news or other facts of the nature of ordinary press information, official documents of a political, legislative, administrative, and judicial nature, such as laws, decrees, resolutions, decisions, state standards, drafts, and official translations, are not protected by copyright. Additionally, we are willing to remove any texts from our corpus upon request from the authors or right owners.

Acknowledgments

We want to thank Andriy Rysin for his contribution to the nlp-uk-api project and support of the Ukrainian language in the LanguageTool project, Kyrylo Zakharov for his work on spiders and help with access to the decisions of the Supreme Court of Ukraine and archive of public Telegram channels, Maria Shvedova and Vasyl Starko for their inspirational work on GRAC and BRUK corpora, Mariana Romanyshyn, Volodymyr Kirilov, Nataliia Romanyshyn, Yehor Smoliakov, Yuriy Paniv, and Stefan Schweter for their feedback and ongoing collaborations on developers' preview of the corpus data.

References

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2022. Towards a cleaner document-oriented multilingual crawled corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4344–4355, Marseille, France. European Language Resources Association.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Rami Al-Rfou', Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.

Vladimír Benko. 2014. Aranea: Yet another family of (comparable) web corpora. volume 8655.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

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.

Nataliia Darchuk. 2017. Mozhlyvosti semantychnoyi rozmitky korpusu ukrainskoyi movy (kum). *Naukovyi chasopys Natsionalnoho pedahohichnoho universytetu im. M.P. Drahomanova*, abs/1911.02116:18—28.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

W. N. Francis and H. Kucera. 1979. Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Natalia Kotsyba, Bohdan Moskalevskyi, and Mykhailo Romanenko et al. 2018. Laboratorija ukrajins'koji.

Volodymyr Kyrylov and Dmytro Chaplynskyi. 2023. GPT-2 metadata pretraining towards instruction finetuning for Ukrainian. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop*, pages 32–40, Dubrovnik, Croatia. Association for Computational Linguistics.

Yurii Laba, Volodymyr Mudryi, Dmytro Chaplynskyi, Mariana Romanyshyn, and Oles Dobosevych. 2023. Contextual embeddings for Ukrainian: A large language model approach to word sense disambiguation. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop*, pages 11–19, Dubrovnik, Croatia. 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.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Nataliia Romanyshyn, Dmytro Chaplynskyi, and Kyrylo Zakharov. 2023. Learning word embeddings for Ukrainian: A comparative study of fastText hyperparameters. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop*, pages 20–31, Dubrovnik, Croatia. Association for Computational Linguistics.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Maria Shvedova, Ruprecht von Waldenfels, Sergiy Yarygin, Andriy Rysin, Vasyl Starko, and Tymofij Nikolajenko et al. 2017-2023. GRAC: General regionally annotated corpus of Ukrainian.

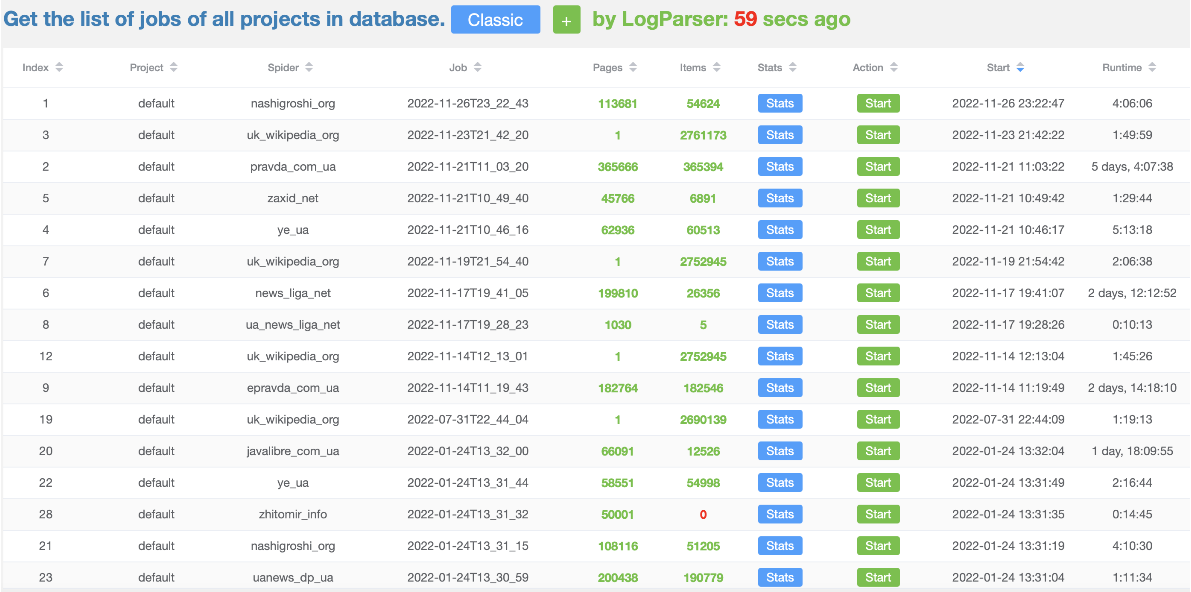
Mukund Srinath, Shomir Wilson, and C Lee Giles. 2021. Privacy at scale: Introducing the PrivaSeer corpus of web privacy policies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6829–6839, Online. Association for Computational Linguistics.

Vasyl Starko, Andriy Rysin, Olha Havura, and Nataliia Cheilytko et al. 2016-2023. BRUK: Braunskyi korpus ukrainskoi movy.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.

A Screenshots of the system

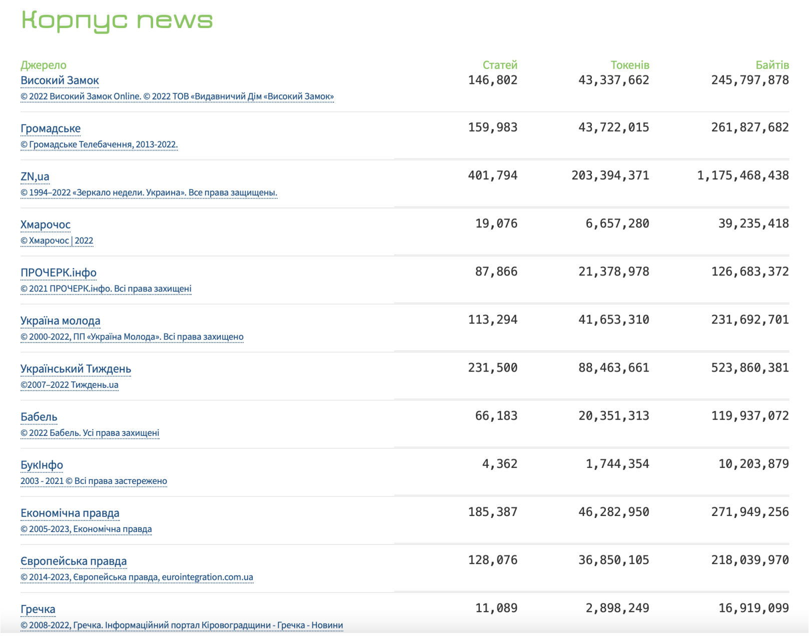
Figure 2: ScrapydWeb webapp to manage corpus spiders



Get the list of jobs of all projects in database. Classic + by LogParser: 59 secs ago

Index	Project	Spider	Job	Pages	Items	Stats	Action	Start	Runtime
1	default	nashigroshi_org	2022-11-26T23_22_43	113681	54624	Stats	Start	2022-11-26 23:22:47	4:06:06
3	default	uk_wikipedia_org	2022-11-23T21_42_20	1	2761173	Stats	Start	2022-11-23 21:42:22	1:49:59
2	default	pravda_com_ua	2022-11-21T11_03_20	365666	365394	Stats	Start	2022-11-21 11:03:22	5 days, 4:07:38
5	default	zaxid_net	2022-11-21T10_49_40	45766	6891	Stats	Start	2022-11-21 10:49:42	1:29:44
4	default	ye_ua	2022-11-21T10_46_16	62936	60513	Stats	Start	2022-11-21 10:46:17	5:13:18
7	default	uk_wikipedia_org	2022-11-19T21_54_40	1	2752945	Stats	Start	2022-11-19 21:54:42	2:06:38
6	default	news_liga_net	2022-11-17T19_41_05	199810	26356	Stats	Start	2022-11-17 19:41:07	2 days, 12:12:52
8	default	ua_news_liga_net	2022-11-17T19_28_23	1030	5	Stats	Start	2022-11-17 19:28:26	0:10:13
12	default	uk_wikipedia_org	2022-11-14T12_13_01	1	2752945	Stats	Start	2022-11-14 12:13:04	1:45:26
9	default	epravda_com_ua	2022-11-14T11_19_43	182764	182546	Stats	Start	2022-11-14 11:19:49	2 days, 14:18:10
19	default	uk_wikipedia_org	2022-07-31T22_44_04	1	2690139	Stats	Start	2022-07-31 22:44:09	1:19:13
20	default	javalibre_com_ua	2022-01-24T13_32_00	66091	12526	Stats	Start	2022-01-24 13:32:04	1 day, 18:09:55
22	default	ye_ua	2022-01-24T13_31_44	58551	54998	Stats	Start	2022-01-24 13:31:49	2:16:44
28	default	zhitomir_info	2022-01-24T13_31_32	50001	0	Stats	Start	2022-01-24 13:31:35	0:14:45
21	default	nashigroshi_org	2022-01-24T13_31_15	108116	51205	Stats	Start	2022-01-24 13:31:19	4:10:30
23	default	uanews_dp_ua	2022-01-24T13_30_59	200438	190779	Stats	Start	2022-01-24 13:31:04	1:11:34

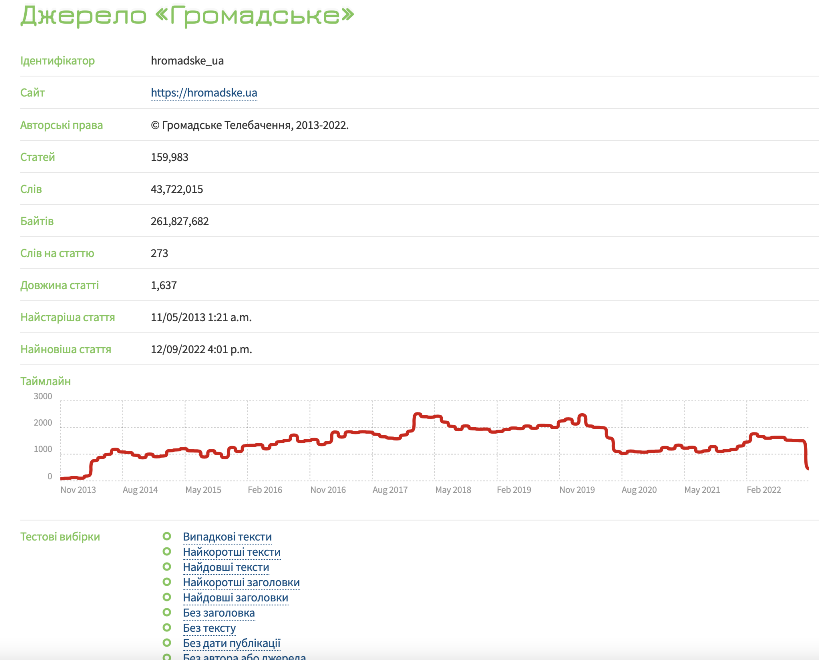
Figure 3: Internal corpus manager and QA tool



Корпус news

Джерело	Статей	Токенів	Байтів
Високий Замок © 2022 Високий Замок Online. © 2022 ТОВ «Видавничий Дім «Високий Замок»	146,802	43,337,662	245,797,878
Громадське © Громадське Телебачення, 2013-2022.	159,983	43,722,015	261,827,682
ZN,ua © 1994–2022 «Зеркало недели. Украина». Все права защищены.	401,794	203,394,371	1,175,468,438
Хмарочос © Хмарочос | 2022	19,076	6,657,280	39,235,418
ПРОЧЕРК.інфо © 2021 ПРОЧЕРК.інфо. Всі права захищені	87,866	21,378,978	126,683,372
Україна молода © 2000-2022, ПП «Україна Молода». Всі права захищено	113,294	41,653,310	231,692,701
Український Тиждень ©2007–2022 Тиждень.ua	231,500	88,463,661	523,860,381
Бабель © 2022 Бабель. Усі права захищені	66,183	20,351,313	119,937,072
БукІнфо 2003 - 2021 © Всі права застережено	4,362	1,744,354	10,203,879
Економічна правда © 2005-2023, Економічна правда	185,387	46,282,950	271,949,256
Європейська правда © 2014-2023, Європейська правда, eurointegration.com.ua	128,076	36,850,105	218,039,970
Гречка © 2008-2022, Гречка. Інформаційний портал Кіровоградщини - Гречка - Новини	11,089	2,898,249	16,919,099

Figure 4: Details about corpus source and text samples



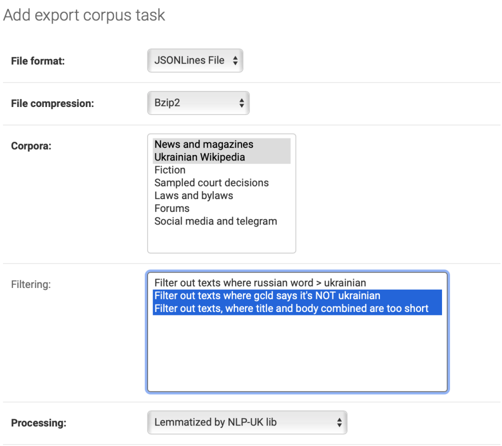
Джерело «Громадське»

Ідентифікатор	hromadske_ua
Сайт	https://hromadske.ua
Авторські права	© Громадське Телебачення, 2013-2022.
Статей	159,983
Слів	43,722,015
Байтів	261,827,682
Слів на статтю	273
Довжина статті	1,637
Найстаріша стаття	11/05/2013 1:21 a.m.
Найновіша стаття	12/09/2022 4:01 p.m.

Таймлайн

Тестові вибірки
- Випадкові тексти
- Найкоротші тексти
- Найдовші тексти
- Найкоротші заголовки
- Найдовші заголовки
- Без заголовка
- Без тексту
- Без дати публікації
- Без автора або джерела

Figure 5: Corpus export task options



Add export corpus task

File format: JSONLines File

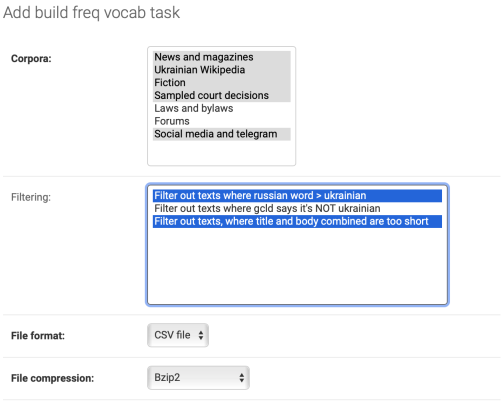
File compression: Bzip2

Corpora:
[✓ News and magazines]
[✓ Ukrainian Wikipedia]
Fiction
Sampled court decisions
Laws and bylaws
Forums
Social media and telegram

Filtering:
Filter out texts where russian word > ukrainian
[✓ Filter out texts where gcld says it's NOT ukrainian]
[✓ Filter out texts, where title and body combined are too short]

Processing: Lemmatized by NLP-UK lib

Figure 6: Lemma frequency export options



Add build freq vocab task

Corpora:
[✓ News and magazines]
[✓ Ukrainian Wikipedia]
[✓ Fiction]
[✓ Sampled court decisions]
Laws and bylaws
Forums
[✓ Social media and telegram]

Filtering:
[✓ Filter out texts where russian word > ukrainian]
Filter out texts where gcld says it's NOT ukrainian
[✓ Filter out texts, where title and body combined are too short]

File format: CSV file

File compression: Bzip2