University *of Ljubljana*
Faculty *of Computer and Information Science*

Slovenian Instruction-based Corpus Generation

Andraž Čeh, Tilen Miklavič, Tom Sojer

Abstract

This study focuses on enhancing the capabilities of Large Language Models (LLMs) to interpret and execute instructions in the Slovene language accurately. As AI assistants become increasingly prevalent across various sectors, there's a need for these technologies to support a wide range of languages, including less commonly supported ones like Slovene. The project focuses on assembling and fine-tuning a dataset of Slovene conversations to enhance the performance of a multilingual LLM in understanding and responding in Slovene.

Keywords
data gathering, corpus, finetuning

Advisors: Slavko Žitnik

Introduction

The NLP field has for a very long time focused on training different models for different tasks. Some of these tasks may include text classification and question answering. Since training these models is resource-intensive and by extension also expensive, a new field has emerged. Large language models (LLMs) have gained popularity in recent years. Just like other models, these too are trained on some corpus of text, but their goal is to create a more general-purpose text-to-text model, which could then further be finetuned to serve specific needs. The finetuning part can extend a model to be used for different languages. This idea can only be realized by generating a large dataset of a specific language which would be as general as possible. This is what we delved into in our project.

The initial idea behind the project was to address the gap in natural language processing (NLP) capabilities for Slovene, a less commonly supported language in the realm of large language models. The primary objective was to create a substantial Slovene language corpus that can be used to train and finetune the LLaMA 2 model, optimizing it for a range of NLP applications specific to Slovene.

Recognizing the limitations of existing LLMs in processing and understanding Slovene accurately, our project seeked to enhance model performance through targeted data collection. The project aimed to assemble a diverse dataset from a variety of sources, including news websites, forums, and other digital media that are rich in Slovene text. This approach ensures a comprehensive linguistic representation, capturing nuances and idiomatic expressions unique to Slovene.

Related work

Corpus generation is a basis for all the text-based language models. The process of cleaning and optimising the data that was used for training the model Llama 2 model is explained by Touvron et al [1]. They talk about increasing the pretraining corpus. For fine-tuning, they have used supervised fine-tuning, reinforced learning with human feedback and Ghost Attention.

In order to make large language models more accessible to the public, researchers worked towards developing BLOOM, a 176 billion parameter LLM that was built by hundreds of researchers [2]. This decoder-only transormer language model was trained on the ROOTS corpus which consists of 1.6 TB of text and spans 46 natural languages as well as 13 programming languages. The corpus is a collection of 498 Hugging face datasets [3]. After data was collected from various sources like the web, Github and large PDF archives, the obtained dataset was preprocessed. This step included deduplication, filtering non-natural language text and removing personal information. BLOOM showed that it is excellent at creating language-diverse content. It also represents a great asset as a code assistant. It has become a useful option due to its multi-lingual nature and the fact that it is open source.

One of the biggest Slovene corpuses spans collections of newspaper articles, magazines, web pages textbooks and fiction. Gigafida 2.0 is an upgrade to previous versions and contains collections that were made from 2012 on, which previous versions do not include. It contains more than 38 thousand text sources and 1.8 billion tokens [4]. The lingusitic

annotation phase involved tokenization, sentence segmentation, morphosyntactic descriptions and lemmatization. After obtaining all the data, the authors also performed deduplication and near-deduplication where text was removed if it exceeded the threshold of the number of n-grams that had already been detected.

Slovene text collection has also been performed by Bañón Marta et al [5], where authors collected data from 1500 domains, preprocessed and a corpus of size around 17 GB was generated. They removed boilerplate content, performed deduplication. Machine translated domains as well as short sequences of text were discarded. Each obtained item in the XML document was accompanied with some metadata to provide a dataset that is useful for corpus linguistic studies and for training large language models. This work was a reference point for our assigment and some of the methods that was used in the study was also used on our part. For instance, they authors annotated each text item with a score that assignes it a score of fluency. We have also done this in order to filter out disfluent sequences.

Methods

Our data processing pipeline consists of four main stages: data collection, preprocessing, generating an instruction-following dataset, and model fine-tuning. The following figure illustrates this data processing pipeline.

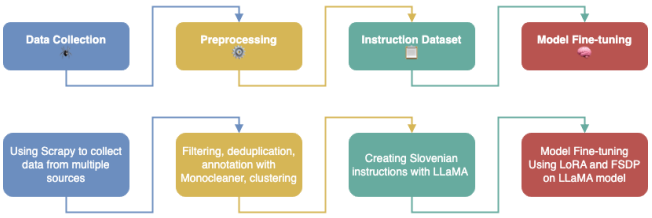

Figure 1. Data processing pipeline

0.1 Corpus generation

To crawl Slovene text from the web, we first needed to decide the appropriate tool that will help us. We have looked into several existing solutions such as Puppeteer, Playwright and Scrapy. There were many different possibilities, but because of ease of use, configuration, an active community and extensive documentation, we went with Scrapy [6]. We used this open source Python framework to generate different spiders with their own properties, depending on the domain that was crawled. For instance, we created a separate spider for the "slo-tech.si", "rtvslo.si" and "24ur.com" domains. Each uses different selectors from which it obtains text data and it writes into separate files. In order to finetune our model, we output the content into xml files with annotations for each obtained content item.

0.2 Preprocessing

After obtaining a larger dataset from different online sources, we needed to do some preprocessing, such as filtering short sequences, deduplicating the content and assigning values based on text length and language fluency. This process was critical for enhancing the efficiency of the finetuning phase, as it reduced redundancy and helps in training the model more effectively on unique data instances. Firstly, we ran the data through steps of excluding short texts and deduplication. Sequences of 10 characters or less were removed as these do not show significant value and context for finetuning. Deduplication is also needed which on one hand reduces the size of dataset used for training and on the other hand avoids overfitting of the model.

In order to further preprocess the obtained dataset, we annotated each text sequence with two tags. Firstly, we added tags that show the fluency of each text sequence. We used **monocleaner**, a Python tool that aims to detect disfluent sentences in a monolingual corpus [7]. Monocleaner can process Slovene language which is great to assign some value to a sequence based on how fluent in a language it is. The tool uses an xml file as an input and adds annotations in form of "lm-score". So more fluent text will have a higher score and we can assign it a higher weight as an annotation.
Secondly, we used tags that describe the length of each sequence in form of "quality-score". Those with very short and long sequences of characters were described as bad and those with around 200 characters were described as good.

This way, we filtered out sequences that seemed disfluent so they are not included in the finetuning phase. We used content that was described as appropriately long and fluent. The output was then used in the next phase.

0.3 Analysing

From gathered corpus we have made some analysis, many of which are talked in different parts of this paper. We have also performed k-means clustering, since we wanted to see how diverse our data is and if at any point it will become very scewed. At the start of the analysis, we had 1132852 words in the dat frame. When we have chosen 5 clusters to sort out corpus in, we have gotten the following representation:

- 20974
- 9607
- 9306
- 7038
- 6067

We can see that the clustering produces fairly good results, from the point that no clusters were too different from one another. There is one exception to this. Cluster 1 is definitelly larger than the following four. This might be, because of the prevailing sort of content we have gotten from one domain.

Since there was only data from siol.net news portal in this corpus, it is possible that the large amount of data was for example comments, which could be classified in the same cluster.

0.4 Generating Instruction-following Dataset

To enhance the model's instruction-following capabilities, we generated a dataset of Slovenian instruction-following examples. This dataset is similar to the Alpaca dataset, but tailored to the linguistic and contextual needs of Slovenian users. We created approximately 50,000 high-quality examples using advanced NLP models.

As part of this process, we utilized the LLaMA 3 model to generate instructions based on the content. By feeding the model with preprocessed text sequences, we were able to generate context-appropriate questions and instructions. This method ensured that the instructions were relevant to the content, thereby improving the model's ability to follow and execute tasks as instructed in Slovenian. This approach allowed the model to understand better and perform tasks as instructed in Slovenian, making it more effective for local applications.

0.5 Finetuning

In adapting our LLaMA model with 7 billion parameters for the Slovene language, we implemented two critical techniques: Low-Rank Adaptation (LoRA) [8] and Fully Sharded Data Parallel (FSDP) [9]. LoRA targets the efficient adaptation of large pre-trained language models. It modifies a select subset of the model's parameters through rank-decomposition matrices. This strategy significantly cuts the number of parameters needing updates, reducing computational costs and speeding up the adaptation process. It also maintains or even enhances performance.

We also enhanced our training capabilities using FSDP [9] on the HPC Sling supercomputer. FSDP optimizes training by distributing the model's parameters across multiple GPUs. This distribution effectively shards the parameters, lowering the memory demands on each GPU. This method is crucial for handling the extensive computational and memory requirements of training large-scale models. It allows for more efficient scaling and better use of the supercomputer's resources.

We applied the Slovenian dataset, comprising approximately 50,000 instruction-following examples, to enhance our model's precision in following instructions in the Slovenian language.

Results

After generating a larger corpus of text and preprocessing we obtained a dataset of size, shown on table 1. The second column shows the size of obtained data immediately after crawling, and the last column shows the size after performing deduplication. We can see that this step greatly reduced the number of text sequences and we can infer that we excluded

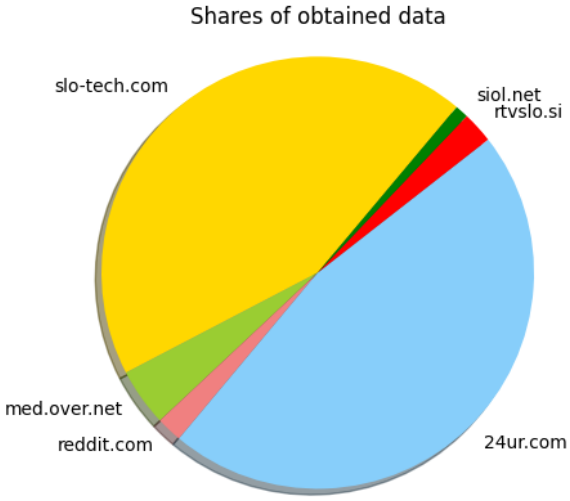

Figure 2. Data source share size in comparison.

a lot of static content that might be present on each of the scraped url. The slo-tech and reddit forums seems to have more content using the same selectors, which made our crawling harder to perform. The size of the initially obtained corpus was **6.2** GB. The size of the preprocessed corpus was **1.8** GB, which means that we significantly reduced the overall size and increased the quality of the dataset.

	Obtained [MB]	Preprocessed [MB]
slo-tech.si	4160	793
24ur.com	976	846
reddit.com	840	35
med.over.net	188	79
rtvslo.si	60	42
siol.net	24	18

Table 1. Dataset size based on domain

We also show a share of each domain from which we performed crawling in figure 2. These are the results after preprocessing was done and we can see that there are two domains that are prevalent by size. Because we would like to have data to be as general as possible, the more interesting domain seems to be "24ur.com".

Below is an example of one preprocessed sequence of text. We annotated each item with a domain, url and timestamp during crawling and added the fluency and quality score during preprocessing. The fluency score is expectedly high, as there are no syntactic and semantic errors. As the sequence is appropriately long but not as desired, we added a "near good" quality score.

- **Content**: Toni Kroos se je vrnil v nemško reprezentanco, kjer sodeluje tudi s Thomasom Müllerjem. Danes bosta tekmeca v polfinalu lige prvakov.

- **Timestamp**: 2024-04-30T08:24:02.483571

- **URL**: https://siol.net/sportal/nogomet/toni-kroos-luka-doncic-bayern-real-632806
- **Domain**: siol.net
- **Fluency Score**: 0.9993332608434458
- **Quality Score**: Near Good

Because this sequence has a good overall score, we used it in the question generation part of the assigment. The content was passed as an output or an answer, and the obtained result is the instruction. This instance of the obtained instruction shows how the generated questions are quite well written.

- **Instruction**: Kdo se je vrnil v nemško reprezentanco in bo ob Thomasu Müllerju igral v polfinalu Lige prvakov?
- **Output**: Toni Kroos se je vrnil v nemško reprezentanco, kjer sodeluje tudi s Thomasom Müllerjem. Danes bosta tekmeca v polfinalu lige prvakov.

When it comes to the fluency score, we were surprised with the results we were getting. The fluency score was consistently high, even considering all the preprocessing we have done.

Some data about the calculated fluency score:

- **Average** 0.9991
- **Minimum** 0.9213
- **Standard diviation** 0.0035

We have some reservations about the method we were using for calculating the fluency score and if researching further, we would definitelly make some more calculations, especially with more complex and diverse texts. We would also like to include text, which was not preprocessed, to see the difference between the two.

One thing we did notice was the length of our corpus was not aligned with our set goal.

- **Short** 61
- **Near good** 26
- **Good** 13

The text we were getting was in the most part too short. It is true that we have set outselves some arbitrary threshold, which would be a good guidance for the length of the text we would be using but we were expecting the corpus to meet the length requirements at least to some greater degree. We think that this is moslty due to the structure of the spider we were using and the fact that we did not limit ourselves to the longer sections of the text from the beginning. We have discovered that the majority of the blog comments or news portal comments are short form text content and thus don't meet our standards for the optimum training length.

The last part of our project was fine tunning the large language model. We successfully fine-tuned the LLaMA model using our Slovenian instruction-following dataset, aiming to improve its task execution in Slovene. However, due to time constraints, we have not yet conducted a comprehensive evaluation to fully assess the improvements. Initial observations are promising, but further testing is needed to confirm these enhancements.

Discussion

Our project successfully addressed the gap in natural language processing for Slovene by creating a comprehensive dataset and fine-tuning the LLaMA model. We collected diverse text data using Scrapy, performed rigorous preprocessing to filter and annotate the dataset, and generated approximately 50,000 high-quality Slovenian instruction-following examples using the LLaMA 3 model. This process enhanced the model's ability to understand and execute tasks in Slovene. By implementing Low-Rank Adaptation (LoRA) and Fully Sharded Data Parallel (FSDP) techniques, we efficiently adapted the model to the Slovene language, reducing computational costs and improving scalability. Although the fine-tuned model demonstrated improved performance in following instructions and engaging in complex dialogues in Slovene, we were unable to conduct comprehensive validation and testing due to project deadlines. This limitation underscores the need for further evaluation to fully ascertain the model's capabilities. As goes for the corpus size, we have used only a few domains as our data source and this could be easily expanded for a bigger research. In such case, we would look into covering an even broader, more general dataset that would cover many different topics. Nonetheless, our work lays a solid foundation for future advancements in NLP applications for Slovene and similar languages.

References

[1] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sha-

ran Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[2] BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. Bloom: A 176b-parameter open-access multilingual language model, 2023.

[3] BigScience Workshop. Bloom (revision 4ab0472), 2022.

[4] Simon Krek, Špela Arhar Holdt, Tomaž Erjavec, Jaka Čibej, Andraž Repar, Polona Gantar, Nikola Ljubešić, Iztok Kosem, and Kaja Dobrovoljc. Gigafida 2.0: the reference corpus of written standard slovene. In *Proceedings of the twelfth language resources and evaluation conference*, pages 3340–3345, 2020.

[5] Marta Bañón, Malina Chichirau, Miquel Esplà-Gomis, Mikel L. Forcada, Aarón Galiano-Jiménez, Cristian García-Romero, Taja Kuzman, Nikola Ljubešić, Rik van Noord, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Peter Rupnik, Vít Suchomel, Antonio Toral, and Jaume Zaragoza-Bernabeu. Slovene web corpus MaCoCu-sl 2.0, 2023. Slovenian language resource repository CLARIN.SI.

[6] scrapy. https://github.com/scrapy/scrapy. accessed: 1.5.2024.

[7] monocleaner. https://github.com/bitextor/monocleaner/tree/main. accessed: 1.5.2024.

[8] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[9] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: Experiences on scaling fully sharded data parallel, 2023.