FRI Natural language processing course 2024
Project report

University of Ljubljana
Faculty of Computer and
Information Science

Slovenian Instruction-based Corpus Generation

Žan Horvat, Bine Markelj, Anže Glušič

Abstract

This paper presents a comprehensive approach to constructing a specialized (conversational) corpus for the Slovene language, aimed at enhancing the performance of large language models through fine-tuning with locally relevant data. We crawled three of the biggest Slovenian forums to gather a substantial and diverse dataset. This data was then preprocessed through a custom pipeline to ensure quality and relevance. Using this refined dataset, we tried fine-tuning a multi-lingual LLM to create a conversational agent tailored for Slovene interactions.

Keywords

natural language processing, large language models, slovenian corpus, conversational corpus, model fine-tuning

Advisors: Slavko Žitnik

Introduction

Slovene falls into the category of languages that are underrepresented in the digital landscape and has not been the focus of (extensive) NLP research. Recognizing this, the paper aims to address the problem by leveraging the rich linguistic content available in Slovenian online forums for corpus generation that can be used for fine-tuning large language models. To achieve this, we undertook a systematic approach that involved crawling data from three prominent Slovenian forums: slo-tech.com, alter.si, and forum.finance.si. These forums were selected for their rich and diverse content, which provides a broad spectrum of conversational data. The gathered data was then processed through a robust preprocessing pipeline designed to clean and organize the data effectively for fine-tuning purposes. Following this, we tried fine-tuning an existing pre-trained multilingual LLM with our curated dataset and evaluated performance.

Related Work

Multiple articles have explored different optimal ways to create useful corpora. Authors of a Bloom LLM [1] set a good example for datasets used to fine-tune LLMs. They used a very large ROOTS corpus, that is a collection of 498 different Hugging Face datasets. This corpus amounted to 1.6 terabytes of text. They emphasize the issues when compiling terabytes of crawled data into corpus. This task can be very hard, especially, because of its size the process has to be completely automatic. A big challenge is therefore preserving the quality of the dataset and filtering of problematic data, that can introduce heavy biases into the corpus, which would impact all the models trained on it.

Of big importance is also the source of the data. Bloom LLM [1] creators balanced the goal of getting as much data as possible, while keeping the quality. For this reason, they chose to crawl only the sites about subjects that they had expert knowledge on. Some of the trustworthy sources they listed were science journals, articles and different expert conferences. A challenging task also proved to be language diversity, since most of the data is in English. Additionally, as a preprocessing step they also made sure to remove all the identifiable personal information, not compromising on anyone's privacy.

Similarly [2] also emphasizes the great importance of quality data. But the researchers decided on an approach, where they scrubbed the minimal amount of data, to not create additional biases and remove certain populations. They also used approaches to determine data's toxicity and report it in the corpus, which can warn researchers using the dataset to train their models. Additionally, they created 3 danger categories and analyzed if any of the data falls into them, such data is then balanced by giving counterarguments and explanations, why this action is bad and harmful, so the model will be able to do the same.

Articles: [3] and [2] report great success of fine-tuning LLMs with large data corpora. They also both show the importance of RLHF. Even a small portion of training being the RLHF technique (human evaluation) greatly improves the model's accuracy and its truthfulness. That way the model gets more aligned with our intent (for example of making it a conversational agent). Such supervised training can have up to 100 times fewer parameters and return better outputs than far larger models without the human evaluation. Such human evaluation is also great for making the model safer. [2] authors report human annotators determining the safety hazard of every answer, which allowed the researchers to analyze areas of weakness and safety issues of the model.

Methods

The success of fine-tuning a large language model (LLM) for specific tasks heavily relies on the quality and relevance of the training data. For the Slovenian Instruction-based Corpus Generation this project, we aimed to create a robust dataset of conversational data in Slovene. This chapter outlines the process of dataset construction and categorization, including the selection of data sources, data crawling, preprocessing pipeline, and the final organization of the dataset.

Data Sources

Since our task is fine-tuning a conversational agent, we wanted to find the data, that would follow natural conversations. We decided, that our best option are therefore popular Slovenian forums. To build a comprehensive corpus, we identified three popular Slovenian forums as primary data sources:

- slo-tech.com: A prominent forum focusing on technology, programming, and IT-related discussions.
- alter.si: diverse community forum with discussions ranging from daily life topics to specialized areas such as health, travel, and hobbies.
- forum.finance.si: forum dedicated to financial topics, including investments, economics, and personal finance, include upvotes for messages.

Data Crawling

We developed web crawlers tailored to each forum to collect the necessary data. The crawlers were designed to navigate through forum threads, extracting posts and comments with additional metadata (such as upvotes). The data crawling process involved the following steps:

1. **Forum Structure Analysis**: understanding the HTML structure and navigation patterns of each forum to ensure efficient and complete data extraction.
2. **Crawler Development**: implementing web crawlers using TypeScript and Crawlee library to automate the data collection process.
3. **Data Extraction**: Collecting textual data from forum posts and comments, including metadata such as timestamps, user IDs, and thread titles to provide context.

```
{
    "ctx": "Jamstvo za ban ne vloge bo nekoliko
        vi je",
    "ctx_link": "https://www.finance.si/finance/
        jamstvo-za-bancne-vloge-bo-nekoliko-visje/
        a/163626",
    "upvotes": 0,
    "content": "\nJa, drzi., se opro  am - na
        netu je ze popravljeno Hvala, LP, M",
    "author": "anon-11946"
}
```

Listing 1. Example entry from *finance.si* forum

In the example from *finance.si* forum above:

- *ctx*: provides the context or title of the discussion thread.
- *ctx_link*: URL link to discussed article (every thread on finance.si is about a single article).
- *upvotes*: number of upvotes the post received, indicating its popularity (or quality).
- *content*: the actual content of user post.
- *author*: anonymized username of the post's author, which helps in preprocessing pipeline.

Preprocessing pipeline

The raw data extracted from the forums required thorough preprocessing to ensure its quality and relevance for fine-tuning the large language model (LLM). The preprocessing pipeline was designed to handle various aspects of data cleaning, normalization, and preparation to create a high-quality training dataset.

1. Data cleaning

The first step in the preprocessing pipeline was to remove irrelevant content from the raw data. By using regular expressions and HTML parsing techniques, we stripped away tags and unwanted sections to focus solely on the user-generated content. Additionally, any non-textual data, such as images, videos, and embedded links, were excluded, as they do not contribute to the text-based model training. We also removed all the special unicode characters, like paragraph and line endings. Lastly, we handle some site specific edge cases, like citations and quotes.

2. Data structure

We had to analyze the type of data we had. This knowledge helped us to determine how the usual conversation goes, which made it possible to automate the process. Since we have data from multiple forums, we noticed, that the initial post is most likely a question, while other users then give their answers. It was of crucial importance to determine, which forum post answers which question. We took advantage of the quotation systems these forums have to match questions with answers and correctly organise them into conversations. We also used a method, that would look for user mentions in messages and therefore match them. A difficult part proved to be determining, which posts were questions. We implemented a regex based method, that would look for known Slovenian indicators for questions.

3. Corpus structure

Our desired corpus structure was based on creating a corpus for conversational agent fine-tuning. We decided to keep the conversation length minimal and therefore always organize the data into 2 ”levels” – meaning there is 1 question and then an answer (or multiple answers) to it (not full long conversations). That allowed us to create a lot of small conversations. Keeping the conversation length small greatly improved the correct answer-question classification. We wanted to also evaluate the quality of each answer, we used different metrics to do that, based on the forum. We used upvotes and likes of posts to rank them higher – more likely to be useful answers. Another technique was analyzing user's forum roles, where our hypothesis was, that users with more posts and more reactions to their posts are likely more helpful and ranked higher.

All conversations were combined into a single JSON file where each conversation has an entry structure like shown in

```
{
        "index": 8,
        "source": "alter",
        "role": "user",
        "prompt": "Nokia PC Suite ali Nokia OVI
            Suite. Imam nokio N79. Kaj naj
            namestim? Nokia pc suite ali nokia ovi
             suite?",
        "answers": [
            {
                "role": "assistant",
                "message": "PC Suite je bolj za
                    generalno uporabi, Ovi suite
                    je bolj za multimedijske
                    vsebine..",
                "answer_rating": 3,
                "answer_hate": 0
            },
            {
                "role": "assistant",
                "message": "Jest imam kar oboje,
                    nekaj delam na pc suitu, nekaj
                     na oviju, kjer mi bolj sede
                    .",
                "answer_rating": 3,
                "answer_hate": 0
            }
        ]
    }
```

Listing 2. Example entry from final corpus.

In the example corpus entry above:

- *index*: is an ID of a conversation.
- *source*: shows, from which forum the conversation is.
- *role*: where the one asking the question is ”user” and the people answering are ”assistant”s.
- *prompt*: is the question message.
- *anwers*: represents a list of possible answers to the question. Each answer has the following properties:
  - *role*: is now here ”assistant”.
  - *message*: is answer to the question.
  - *answer_rating*: shows the quality of the answer – larger number is better.
  - *answer_hate*: shows the amount of hateful words used – larger number is worse.

4. Language filtering

We employed a strategy described in [2] to filter the dataset. Meaning, we did not delete data, but rather evaluated its toxicity. But unlike them, we were of course not able to use RLHF with human annotators, but relied on a predefined list of toxic words and penalized the answers for each use of them. This together with answer rating score can be used by other researchers to determine the weight of each answer for the model fine-tuning, based on their task.

5. Context preservation

Preserving the conversational context was another critical aspect of preprocessing. This involved maintaining the thread structure and the sequence of interactions. We ensured that replies were linked to their original posts by tracking thread IDs and parent-child relationships within forum discussions. This linkage preserved the natural flow of conversation, allowing the model to understand the context of each comment. We also further increased the context of messages, using post's titles and linked articles to give greater meaning and context to the subject of each conversation.

Mistral

For testing our corpus, we chose **Mistral-7B** (accessible from *https://huggingface.co/*), with nice implementation documentation and fine-tuning capabilities. This model is claimed to be the most powerful language model for its size to date.

It was also shown, that Mistral-7B can be fine-tuned (with no proprietary data) for chat application, and it outperformed all models with 7 billion parameters on MT-Bench [4]. It's comparable to models with 13 billion parameters. It boasts with being performant in English and code.

We also acknowledge the existence of *Mixtral 8x7B* which is a better model and is also implemented with open licence. Comparatively to Mistral7B, it is also fluent in French, Italian, German and Spanish. With this assignment, we wished to also fine-tune this model and compare it to its smaller brother. Both models have a 32k context window.

We tested understanding of Mixtral 8x7B with Slovenian text, which it was perfectly capable of answering and translating it into English. Therefore we confirm this model to be perfect for our application.

Fine-tuning Mistral model

Once we collected the scraped data into a well organized corpus, we wanted to fine-tune the Mistral model, to make it better in conversing in Slovene. Despite our efforts, we were not able to produce meaningful fine-tuned model (either on Arnes HPC, Google-colab or locally). We used different Nvidia GPUs for this application, with little to no success, because of lack of computational resources to complete such large task. f Unfortunately, we were not successful in fine-tuning the model, due to lack of time and resources. But we did prepare and upload a full fine-tuning Jupyter notebook for this application, for future development or for anyone with good enough computational resources to complete the task.

Results

Evaluation methods

Evaluating our corpus is crucial. We used human evaluation for measuring the quiality of our corpus and planed the same for our fine-tuned model.

Data statistics

The final dataset exhibited the following characteristics:

- **Total posts**: 166565 conversations
- **Vocabulary size**: 36159817 tokens

Random sampling

In random sampling, we select a random sample of posts from the dataset and manually review them for relevance, coherence, and completeness. Posts should be meaningful, contextually appropriate, and free of grammatical errors. The criteria were:

- **Clarity**: how easily understandable and unambiguous the text is
- **Relevance**: text should be consistent in its message and relevant to the topic or conversation

Every member of the group chose ten random samples from the dataset. 77 % of text was rated as understandable and 65 % as relevant.

Duplicate detection

We detected and removed duplicates from posts. Duplicates were especially prevalent when authors cited other user's messages. We removed those messages.

Topics

We analyzed the distribution of topics. The majority of posts were from technology, followed by finance posts and lifestyle posts. This was expected since we crawled the biggest IT and finance forums in Slovenia.

Discussion

We created quite large dataset for creating Slovenian conversational LLM. Though unsuccessful in the endeavour of creating our own, we are very confident (almost certain) in the possibility of fine-tuning one of LLMs with an open licence. Our work is a step in the right direction for creating LLM for Slovenian language, which is underrepresented in this field of study, as it has comparatively very little native speakers, compared to other languages. The results of the project demonstrate the feasibility of fine-tuning large language models for specific languages like Slovene. Fine-tuned model would exhibit improved capabilities in understanding and generating Slovene text, making it a valuable tool for various applications. Throughout the project, several challenges were encountered, the primary challenge was ensuring high dataset quality. Handling the nuances of the Slovene language also required careful attention. Fine-tuning LLM also demanded significant computational resources (mainly GPUs) and unexpected errors which we could not resolve.
Future work will focus on expanding the dataset to include more diverse sources, further refining the preprocessing techniques, and exploring additional evaluation metrics.

References

[1] BigScience Workshop: Le Scao et al. Bloom: A 176b-parameter open-access multilingual language model, 2023.

[2] Touvron et al. Llama 2: Open foundation and fine-tuned chat models, 2023.

[3] Ouyang et al. Training language models to follow instructions with human feedback, 2022.

[4] Zheng et al. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.