DIKU NLP Course 2024: Group Project

Last updated September 2, 2024
This project description may be slightly adapted throughout the course. We will notify everyone on Absalon in case of such changes.

The aim of the project is to create a **multilingual question answering** system using a publicly available dataset consisting of question-document-answer items in diverse languages.

The project is incremental and follows the course syllabus. Each of the sections below is intended for one course week. You have the freedom to choose the methods, from the course material or otherwise, to complete each task.

Organization. The project will be graded as a whole and all parts of this project are mandatory. Complete the project in groups of up to **3 students**.[1] Your group report must indicate:

- who was responsible for what part of the project (contributions);
- motivated explanations for the decisions and considerations you made;
- a human-readable description of what you have implemented;
- your results and observations; and
- whether and how you used AI assistance (see §8).

Format. Submit a single PDF document using the ACL template[2] (non-anonymized)[3] with at most 4 (four) pages, including figures and tables. References, appendices, contribution statement and AI assistant statements do not count towards the page limit, but content after the 4th page will not be graded. The report language must be English. Submit your code in a ZIP file.

Mid-term feedback. You **may** submit your intermediate report to receive feedback from the course instructors without grade by 3 October 17:00 via Absalon. In our experience, this greatly improves the final submitted projects.

Final submission. You **must** submit the final report and code before **1 November 17:00** Copenhagen time on Digital Exam (no need to submit on Absalon). Submit one report per group. Note that the system will automatically close for submissions at the exact deadline, and that you can edit your submission as many times as you would like before the deadline.

[1] You can complete the project on your own, but we recommend working in groups.
[2] https://github.com/acl-org/acl-style-files
[3] In LaTeX, remove the `[review]` option from `\usepackage[review]{acl}`.

Infrastructure. For running your code, we recommend using Google Colab,[4] which provides free access to computing resources including GPUs. Note that usage limits are time-based for free accounts. If you see the error message "No CUDA GPUs are available", waiting a few hours will resolve the problem. You should not have to pay for an account for this course, but make sure to complete and run your code regularly and in good time before submission.

1 Week 36 (2–8 September)

TyDi QA (Clark et al., 2020) is a multilingual question answering dataset. In one of its subtasks, a question and context document are given, and the task is to select the document span containing the answer. In this project, we will use the Reading Comprehension task from the Cross-lingual Open-Retrieval Question Answering dataset (XOR RC; Asai et al., 2021), as well as the XOR-AttriQA dataset (Muller et al., 2023). They are based on TyDi QA, with two main differences: some of the questions in them are unanswerable (impossible to answer given just the provided context); the context documents and answers are in English, but the questions are in other languages (Arabic, Bengali, Finnish, Japanese, Korean, Russian and Telugu).

(a) Explore the dataset from `https://huggingface.co/datasets/coastalcph/tydi_xor_rc`. Familiarize yourself with the dataset card, download the dataset and explore its columns. Summarize basic data statistics for training and validation data in each of the languages Finnish (`fi`), Japanese (`ja`) and Russian (`ru`).

(b) For each of the languages Finnish, Japanese and Russian, report the 5 most common words in the questions from the training set. What kind of words are they?[5]

(c) Implement a rule-based classifier that predicts whether a question is answerable or impossible, only using the document (context) and question. You may use machine translation as a component. Use the `answerable` field to evaluate it on the validation set. What is the performance of your classifier for each of the languages Finnish, Japanese and Russian?

2 Week 37 (9–15 September)

Let k be the number of members in your group ($k \in \{1, 2, 3\}$). Implement k different[6] language models for the questions in the three languages Finnish, Japanese and Russian, as well as for the document contexts in English (total $k \times 4$ language models), using the training data. Evaluate each of them on the validation data, report their performance and discuss the results. Reminder: a language model is a function that takes text as input and returns its probability.

[4] `https://colab.research.google.com/`
[5] Use machine translation models, e.g., opus-mt-fi-en, opus-mt-ja-en, and opus-mt-ru-en.
[6] Different approach (n-gram/neural) or different n, different smoothing etc.

3 Week 38 (16–22 September)

Let k be the number of members in your group. For each of the three languages Finnish, Japanese and Russian separately, using the training data, train k different[7] classifiers that receive the document (context) and question as input and predict whether the question is answerable or impossible given the context. Evaluate the classifiers on the respective validation sets, report and analyse the performance for each language and compare the scores across languages.

The classifiers can use machine translation, linguistic/lexical features (e.g., bag-of-words, n-gram counts, word overlap) word embeddings, or word/sentence representations from (multilingual) neural language models.[8] You can also train or fine-tune your own neural language models on the dataset. Different from 1(c), however, they must be *learned* rather than rule-based. Motivate your choice of features and classifier.

4 Week 39 (23–29 September)

We now move from binary classification to span-based QA, i.e. identifying the *span* in the document that answers the question.

Let k be the number of members in your group. Using the training data in Finnish, Japanese and Russian separately, train k different sequence labellers, which predict the tokens in a document context that constitute the answer to the corresponding question.[9] You can decide whether to train one model per language or a single model for all three languages. Evaluate using a sequence labelling metric on the validation set, report and analyse the performance for each language and compare the scores across languages. Note that if the question is unanswerable, a correct output must be empty (contain no tokens).

5 Week 40 (30 September–6 October)

We now introduce open QA, i.e. *generating* an answer to a question even when it is not extracted as a span from a document.

While for all answerable questions in the dataset, the English answer is available, for some of the questions in the dataset, the answer in the same language as the question is also available, in the `answer_inlang` field. Use this subset of the questions in Finnish, Japanese and Russian to train (or fine-tune) an encoder-decoder model that receives the *question and context* as input and generates the in-language answer.[10] You can decide whether to train one model per language or a single model for all three languages.

If your group contains at least two members, additionally train an encoder-decoder model that receives only the *question* as input and generates the in-language answer.

[7] Different architecture, different features, or both.

[8] You can use a pretrained encoder, e.g., multilingual DistilBERT.

[9] That is, a label needs to be predicted for each token—you must indicate which encoding scheme and why. Note that the dataset specifies character indices, so a conversion is necessary.

[10] You can use a pretrained encoder-decoder model e.g., mT5-small.

If your group contains at least three members, additionally train an encoder-decoder model that receives only the *English answer* as input and generates the in-language answer.[11]

Evaluate using a text generation evaluation metric on the validation set, compare the results across languages and models and discuss them.

6 Week 41+ (from 7 October)

While generating an answer is more flexible than extracting it as a span, it may be *right for the wrong reasons*, i.e. the answer may be correct even if the question is unanswerable given the context.

Use *all* questions in Finnish, Japanese and Russian to train (or fine-tune) an encoder-decoder model that receives the *question and context* as input and generates the *English* answer. You can decide whether to train one model per question language or a single model for all three languages.

Evaluate using a text generation metric on the validation set, and compare the overall results between *answerable* and *unanswerable* examples. Can the model answer correctly even when the answer is not provided in the context? Discuss the results.

7 Structure and Grading of the Report

The report should clearly state your group name and the names of all group members. It should describe your approaches for all assignments of this project. Use one section per part of the assignment, as well as a section where you state the contributions of each group member.[12] It can also be a good idea to have a conclusions section, where you highlight some of the core challenges, findings and lessons learned from this project. Your report should contain enough details so that it is possible for someone to reproduce your results just by reading your report. Properly describe your models, training scheme, and data processing pipeline.

While we will verify the submitted code, your project will be mainly graded based on the submitted description document. This also means that we will only assign scores for implementations of methods described in the project. Points will be awarded not only for what you have done, but also for the reasoning behind your decisions. When you describe your choice of a model, a tool, etc., you should provide a brief explanation of it and the reason behind your choice in order to demonstrate your knowledge on the various topics. When you describe the results, you should pick appropriate metrics and baselines for comparison. You should not only report raw numbers, but also attempt to explain result trends and differences between sets of results. If you experimented with different methods for the assignment, it is fine to include the key results in the main part of the report, and additional results for, e.g., ablation studies or unsuccessful early experiments in an appendix. Note that you will also receive overall points for demonstrated mastery according to the criteria listed above.

[11] Hint: you can fine-tune a machine translation model for each language.

[12] In case it is not clear which member contributed to which part, all group members will receive the same grade.

8 Academic Code of Conduct

You are welcome to discuss the project with other students, but sharing of code is not permitted. Copying code or text from the report directly from other students will be treated as plagiarism. Please refer to the University's plagiarism regulations if in doubt. For questions regarding the project, please ask on the Absalon discussion forum.

In short, plagiarism means copying text or ideas from others without acknowledging the underlying sources. Crucially, this does not mean that you are prohibited from building on others' ideas or use external sources, but rather that you have to properly acknowledge all sources used in your work. This holds for instance for building on code from lectures or lab sessions. If in doubt, we recommend erring on the side of over- rather than under-acknowledging sources.

You are also welcome to use AI assistance (e.g., ChatGPT, GitHub Copilot) for tutoring purposes, augmenting the TAs for help with questions and issues. However, keep in mind that their output is not guaranteed to be either comprehensive, true or aligned with the course scope and expectations. Always check with the TAs in case of doubt. Importantly, the use of AI assistance while writing the project report is allowed only for the following purposes:[13]

- As coding tools (e.g., GitHub Copilot): no restrictions.
- As writing tools to improve the writing of original content, i.e. when the prompt you write contain all the ideas to be formulated: no restrictions.
- As search tools to identify related literature: no restrictions. Usual citation requirements apply (see plagiarism note above): you must cite the original work you identify, even if you used an LLM to find it. Just like you do not cite Google Search for papers you find using it, you should not cite ChatGPT for this either. In particular, always make sure that the citations it provides actually exist—LLMs are known to often generate plausible but nonexistent references.
- As generation tools for *new* ideas: generated content must be clearly highlighted even if post-edited by yourself. All prompts/transcripts from the tools used must be included as an appendix at the end of the submission in PDF, after the references.

For all uses of AI assistance, the purpose, tool, and version must be stated in the submission—e.g, in a dedicated section. Here is such a statement for example:

```
ChatGPT 4o was used as a writing assistance tool and as a search
 tool to identify related literature.  GitHub Copilot was used
                 while developing the code.
```

References

Akari Asai, Jungo Kasai, Jonathan Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. 2021. XOR QA: Cross-lingual open-retrieval question an-

[13] Note that evaluating LLMs as models on task data is not considered "AI assistance" and is not restricted or affected by the rules here.

swering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 547–564, Online. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Benjamin Muller, John Wieting, Jonathan Clark, Tom Kwiatkowski, Sebastian Ruder, Livio Soares, Roee Aharoni, Jonathan Herzig, and Xinyi Wang. 2023. Evaluating and modeling attribution for cross-lingual question answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 144–157, Singapore. Association for Computational Linguistics.