DIKU NLP Course 2023: Group Project

Last updated September 1, 2023
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 the publicly available TyDi QA[1] dataset, a set of question-document-answer items covering 11 typologically diverse languages. You will work with the following languages: **Arabic**, **Bengali** (Bangla) and **Indonesian**.

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**.[2] Your group report must indicate:

- who was responsible for what part of the project;
- 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[3] (non-anonymized)[4] with at most 4 (four) pages. References and appendices 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 2 October 15:00 via Absalon. In our experience, this greatly improves the final submitted projects.

Final submission. You **must** submit the final report and code before **3 November 15:00** Copenhagen time on **Digital Exam**. 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] Paper: https://arxiv.org/abs/2003.05002, talk: https://slideslive.com/38929512
[2] You can complete the project on your own, but we recommend working in groups.
[3] https://github.com/acl-org/acl-style-files
[4] In LaTeX, remove the `[review]` option from `\usepackage[review]{acl}`.

Infrastructure. For running your code, we recommend using Google Colab,[5] 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 (5–11 September)

While the **MinSpan** subtask of the original TyDi QA dataset only contains answerable questions, we will use its extention **Answerable TyDi QA**, which additionally contains questions that are **not answerable**: https://huggingface.co/datasets/copenlu/answerable_tydiqa

(a) Familiarise 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 Arabic, Bengali and Indonesian.

(b) For each of the languages Arabic, Bengali and Indonesian, report the 5 most common words in the documents from the training set. Then report the 5 most common words in the *questions* from the training set. What do you observe?[6]

(c) Implement an "oracle" function that indicates whether a question is answerable or not given the document and answer. That is, the function will output 1 if the answer to the question appears in the document and 0 otherwise. Then implement a rule-based classifier that predicts whether a question is answerable only using the document and question. Use the oracle function to evaluate it. What is the performance of your classifier on the validation set for each of the languages?

2 Week 37 (12–18 September)

Let k be the number of members in your group ($k \in \{1, 2, 3\}$). Implement k different[7] language models for each of the three languages, separately for the questions and the documents (total $k \times 3 \times 2$ language models), using the training data. Evaluate each of them on the validation data, report their performance and discuss the results.

3 Week 38 (19–25 September)

Let k be the number of members in your group. Implement and train k different[8] supervised classifiers for each of the three languages separately, using the training data for that language. The classifiers must only use the document and question as input. Evaluate the classifiers on the respective validation sets,

[5] https://colab.research.google.com/
[6] Hint: use a machine translation tool for languages you do not speak.
[7] Different approach (n-gram/neural) or different n, different smoothing etc.
[8] Different architecture, different features, or both.

report and analyse the performance for each language and compare the scores across languages.

The classifiers can use linguistic/lexical features, e.g., bag-of-words, n-gram counts, overlaps of words between question and document, etc.; word embeddings, or word/sentence representations from neural language models. You can, for example, find pretrained Transformer language models for different languages, trained with different language objectives, and fine-tuned for different downstream tasks, from Hugging Face.[9] You can also train or fine-tune your own neural language models on the dataset. Motivate your choice of features and classifier.

4 Week 39 (26 September–2 October)

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

Let k be the number of members in your group. Using the training data, implement k different sequence labellers for each of the three languages, which predict which tokens in a document are part of the answer to the corresponding question. Evaluate the sequence labellers on the respective validation sets, report and analyse the performance for each language and compare the scores across languages.

5 Week 40 (3 October–9 October)

For both the binary classification and the sequence labeling tasks, select one of the models you implemented which uses a Transformer architecture. If you have not implemented one yet, do so now.

Select one validation instance for each of the three languages, and visualize the attention in one of the model layers for both the binary classification and the sequence labeling models. Can you explain the model predictions based on the visualization?

6 Week 41+ (from 10 October)

Perform zero-shot cross-lingual evaluation by training a classifier *and* a sequence labeller on each language's training set and testing each of them on each of the other two language's validation set.[10] Discuss the results and compare them to the monolingual results (obtained by evaluating on the same language as in training). Which language is the most beneficial as training language?

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

[9] https://huggingface.co/models

[10] Hint: make sure to use an encoder or features that support multiple languages.

the contributions of each group member.[11] 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.

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:[12]

- 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.

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

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

- 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[13] must be stated in the submission—e.g, in a dedicated section. Here is such a statement for example:

```
ChatGPT August 3 Version was used as a writing assistance tool
 and as a serch tool to identify related literature.  GitHub
 Copilot July 14 Version was used while developing the code.
```

[13] If multiple version have been used throughout the project, list all of them. In the ChatGPT web interface, the version is specified at the bottom of the screen, under the text input field.