Ostbayerische Technische Hochschule
Amberg-Weiden

Project Work Natural Language Processing Winter 2024

Prof. Dr. Patrick Levi

November 26, 2024

Start: 09.12.2024 (via Moodle)
Deadline: 09.01.2025 23:59:59 (via Moodle)

General

This is the final project for the Natural Language Processing Lecture (winter 2024). Your task will be to build up mainly the retrieval part of a Retrieval-Augmented-Generation (RAG) system for medical text data.
The challenges are to demonstrate a sound understanding of NLP techniques, the ability to acquire new techniques required to solve this task and to deal with text in a difficult domain (medicine).

General Requirements

- Your solution of this project work consists of a documentation (approx. 10-15 pages) and code files.
- Your documentation must be a PDF. The use of LaTeX is recommended, but not obligatory. However, the documentation shall be in a proper report format (see grading). Markdown files or Jupyter notebooks hardly fulfill good report criteria.
- Hand in a zip file containing your documentation, the filled-in and signed declaration at the end of this project description for each team member, and your code.
- Upload your complete solution until the specified deadline via Moodle.
- Any documentation part or code parts created with AI tools must be specified, what tool was used and to what purpose the tool was used.
- Plagiarism: If your solution copies contains parts copied from any resource including solutions of team mates (for individual parts) or of other teams, all solutions involved in the plagiarism (the copy and the master solution) will be graded with 5.0 (fail).

Team Requirements

- The following project is subdivided into two parts, a team part and an individual part. The individual part must be done individually by each team member!
- You are free to do the whole project work individually. You do not have to work in teams.

- You must select a team via Moodle before the start date above. Team changes are not possible after this date. However, you are free to decide to do the task individually at any time. Just hand in an individual solution.
- Therefore, please indicate the authorship in your final solution (see below, requirements on team authorship).
- In case of a team breakup during the project, each team member has to hand in an individual solution. New teams may not be formed.
- In case you work in a team:
  - The team size is limited to two students.
  - Every student has to hand in a complete solution (code and documentation for team part and individual part).
  - Please indicate the correct authorship in your documentation (e.g. "This chapter was authored by name team member 1 and name team member 2." for team parts). The same applies for code. In your code, indicate the authorship in the docstrings of your classes and methods.
  - In the project specification you will find some more specific team requirements.
  - **Grading will be individual.**

Grading Criteria

General

- Your solution solves the task and has sufficient quality.
- Your solution is well founded and well justified. Explain your solution in the documentation.
- Do not limit yourself just to techniques from the lecture but also research other possible approaches to find the best way to solve the project. Include current knowledge in the field and the current literature
- Your solution is efficient and effective (do the right things, do the things right).
- Your solution exceeds the quality obtained by AI tools when they are asked to solve the task.
- Your solution demonstrates a deep understanding of the problem.

Code

- Code must be written in Python (except otherwise specified)
- Your code is well structured (packages, classes, methods, ...), easy to read, understandable, and there are sufficient comments in the code. Uncommented code will be down-graded.
- In addition to comments in the code, every function must contain an appropriate docstring. You can follow the NumPy docstring guide: `https://numpydoc.readthedocs.io/en/latest/format.html`. Notice, a Sphinx documentation is not required.
- As a rule of thumb: The more complex the function, the more comments are required in the code.
- Your code is efficient, understandable, and written in a way that is not error-prone.
- Wherever possible, use available Python packages. Restrictions might be specified in the project description.
- Your code must run on the computers in the GPU lab (DC 1.07).

Documentation

- Your documentation presents your solution. Avoid unnecessary information in the documentation. It is not intended to be a protocol of your progress, but a result report.
- It must be written in a way that another AI master student, who is not an expert in the field of the project task, could follow what you did and why you did it.
- It must follow a scientific writing standard. Take scientific papers as a template.
- It must be well-structured and written in proper language.
- Tables and figures shall be on point, clear, and consise.
- Each step in your solution must be well justified in the documentation.
- List all your references, use a proper scientific citation standard.

Project – Medical RAG System

Suppose you work for a company building information systems with Large Language Models (LLMs). They want to prototype a new idea and build a RAG system for medical applications used e.g. by physicians, nurses, biomedical researchers, ... The prototype is not required to run at scale, however, it shall be functional and created in a way that it can be re-used and extended for a productive application. Therefore, code must be structured, understandable, and maintainable. The documentation must be understandable by people with an AI background who are not necessarily experts in RAG and LLM technologies.

Your task is to build the prototype for the retrieval part. The requirements for the retrieval part are the following:

- The system shall work with three provided datasets (named dataset 1, dataset 2, and dataset 3, provided below.)
- Dataset 1 contains medical questions, usually four options for answers, and an indication of the correct answer.
- Datasets 2 and 3 contain medical information from PubMed database and medical textbooks, respectively. Use them as source for medical context on the questions in dataset 1.
- For every question from dataset 1, the retrieval part shall retrieve information for every one of the four possible answers, which relates to the question.
- The retrieved information shall be suitable for a medical professional to decide which of the options for the answer actually is the answer to the question.
- Therefore the retrieved information must relate to both, answer option and the corresponding question.
- The retrieved information per answer option shall be limited to a maximum of 700 tokens.
- These max. 700 tokens shall contain the most useful information found in the datasets.
- Your system does not need to answer the question, but just provide context on the answer options.
- Any further datasets or other external data sources must not be used (no download of PubMed papers, no queries to databases, ...).
- The retrieval part must not connect to any external API, especially no AI tools like ChatGPT or others.
- Resources are restricted to the computers in DC1.07.

Datasets

- Dataset 1 - MedQA-USMLE-4-options [1],
  `https://huggingface.co/datasets/GBaker/MedQA-USMLE-4-options?row=9`

- Dataset 2 - Medical Meadow Wikidoc,
  `https://huggingface.co/datasets/medalpaca/medical_meadow_wikidoc`
- Dataset 3 - PubMedQA, [2],
  `https://github.com/pubmedqa/pubmedqa?tab=readme-ov-file`

Project parts

The project is subdivided into two tasks. Part 1 can be done as a team, part 2 must be done individually. The effort split should be approx. 33% for part 1, 67% for part 2.

Part 1

Part 1 can be done by one student or by a team of two students. In case you work in a team of two, please follow the requirements at the end of the task description.

Get and explore the datasets. Goal of your exploration is to get an overview which topics are contained in the question dataset (dataset 1) and in the information datasts 2 and 3. Analyze how much the topics match, diversity of the questions and answers, ... Try to get every information that is required or helpful to create a good retrieval component. Present your analysis in the documentation. Your analysis shall give you a sound overview what these documents are about, whether they might be useful for the task or also possible difficulties that could come up during development. Consider the reader of your report does not know the datasets.

Create an initial retriever component that fulfills the above requirements. Document the components used and why you used them (references!).
Retrieve documents for the questions and answer options in dataset 1 with your retriever. Evaluate the quality of your retriever. Therefore, also check the quality of your retrieved documents for some selected example questions. Try to find one example where it works well. However, find 10 examples where you are not satisfied with the documents. Find examples with different kinds of reasons why the retrieved information does not match your expectations and analyze them manually. You can use any internet source to understand the medical texts (wikipedia, medical resources, ...). Try to understand at least roughly, whether the retrieved text is suitable or not. Present your analysis in the documentation. Consider your reader does not know the datasets in detail. Listing of the examples shall be moved to the appendix.

Disucss weaknesses of your solution and develop ideas how your retrieval component could be improved. Document references for your ideas, provide evidence why these ideas might work (consider you need to convince a project manager to release money for realizing these ideas). Priorize your ideas and break them down into 2-3 different tasks which can be solved by one person in about 2/3 of the time give for the whole project.

Team requirements
In case you work as a team of two, develop 4-6 ideas during your improvement discussion and break them down into tasks. Ensure an approximately even splitting of these tasks among both of you. Every team member shall continue the next task with a different set of tasks.

Part 2

This part must be done individually.

Realize your assigned tasks, which you previously defined in the team. Check whether you actually improve the retrieval component with these ideas. Re-evaluate your examples from above. Document your methods, tools, experiments, results and discuss them critically.

References

[1] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. arXiv preprint arXiv:2009.13081, 2020.

[2] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2567–2577, 2019.

Ostbayerische Technische Hochschule
Amberg-Weiden

Academic Integrity Declaration for the Project Work

Winter 2024
Prof. Dr. Patrick Levi

Surname, First Name:

Student Number:

Academic integrity declaration for examinations

Herewith I declare that we created this project work by ourself. Individual parts are marked in code and documentation correctly. I declare that I created individual parts of this project work by myself. All used material and references are decleared in the project work.

Place, date, signature