AI Final Assignment: Scientific Abstract Classification

Description

Build and train a classifier to categorize scientific abstracts into one of three chosen scientific fields out of four options: Astronomy, Political Science, Psychology, and Sociology.

1. **Research and Data Collection**: First, conduct preliminary research in the domain of text classification. This involves reading relevant journal and conference papers, online tutorials/articles, and understanding the best practices in this field of NLP. I recommend starting the project by reading a survey paper on text/document classification.
   Choose three fields out of the four provided for the classification task. Collect at least 300 abstracts for each selected field (you must find relevant journals/conference proceedings), ensuring a balanced dataset for a total of at least 900 abstracts. You may collect the abstracts manually or automatically (with a tool, web scraping, etc.), but the dataset must be unique. Students are encouraged to discuss methodologies, but direct collaboration on datasets is not permitted (i.e., you cannot share the datasets for the project, the collection must be done individually).
   Note: if you do web scraping/crawling, make sure you check first if the website prohibits scraping. And even then, use a VPN (Proton VPN has a free version), to make sure your own IP is not banned in case you scraped too fast, etc.
2. **Dataset Creation and Preprocessing**:
   - Label the dataset according to the chosen fields. For example, you can create a single file in a CSV format where the texts are annotated according to their class, or just use plain text files in a folder structure, etc.
   - Based on the research conducted, apply appropriate preprocessing steps if needed (this depends on the machine learning model used) such as tokenization, stop-word removal, normalization, vectorization, etc.
3. **Model Selection, Training, and Validation**:
   - Choose an appropriate machine learning model for text classification. For bonus points, fine-tune a pre-trained deep learning model (but keep in mind that the model, architecture, etc. must be explained in the report).
   - Train the model with the dataset, implementing cross-validation to ensure robustness.
   - Use metrics: accuracy, precision, recall, and F1-score for evaluation.
   - Present a confusion matrix to showcase the model's performance across different categories.
   - Save the trained model (so it can be loaded later for inference).
4. **Report**: Compile a comprehensive report about the project.

Submission

- Students must submit their well-documented source code and the dataset.
  - Students must use Python 3 (both .py and Jupyter Notebook files are ok).
  - Students are not required to make the algorithms from scratch and may use any libraries and reuse any code found on the internet, textbooks, etc. *Using AI tools to create the code itself is allowed.*

- Students must explain all functions/methods in their own words using comments (using functions from libraries) or docstrings (user-defined functions). *Using AI tools to create the comments/explanations is not allowed.*
- The main file(s) must be named: studentname_aifinal.extension, for example, KovacsMate_aifinal.py or KovacsMate_aifinal.ipynb, etc.
- There is no filename restriction for the dataset (and if you have it in a folder structure, just zip it).
- Do not submit the saved model (but the code must include saving the model).
- Note 1: if the code(s) are not documented well (in the comments/docstrings), the submission will not be accepted.
- Note 2: if you used web scraping to obtain the abstracts, you do not need to submit the code for the scraping part (although you need to mention it in the report when you explain about data acquisition).

• Students must submit a report explaining the project and summarizing the results.

- *Using AI tools to generate the report is strictly prohibited.*
- The report must be written in English and have a logical structure (with sections). For example, Introduction, Methodology, Experiments, Results, and Conclusions. Using tables, diagrams, and other figures for illustration is highly encouraged, but all must be original. You cannot use figures/tables from the Internet/papers (even if you would cite the source).
- Explain about the domain, data acquisition, preprocessing, model development (with e.g., the detailed network architecture), evaluation/validation, and interpret the results.
- Having a "References" (bibliography) section is not a requirement.
- The minimum length required is 3000 words, maximum is 3500 words. Only the main text counts (e.g., student name, references, etc., do not count). From Merriam-Webster: *minimum (noun) - the least quantity assignable, admissible, or possible.* So for example, a submission with 2999 words is not accepted.
- The document must include the student name, student ID, course, and assignment name.
- The only accepted format is PDF, and the report must be named: studentname_ai_finalreport.pdf, for example, KovacsMate_ai_finalreport.pdf
- **Note: all reports and the code comments will be checked 1. for plagiarism, 2. in-class and back to 2 years inter-student similarity, 3. with a state-of-the-art system for detecting AI-generated text. Academic misconduct/cheating in any form will be reported and investigated by the university's committee.**