IS 584: Deep Learning for Text Analytics

Term Project

Aspect-Enhanced Peer Reviews in Journals

Middle East Technical University
Graduate School of Informatics

Date: April 8, 2025

Introduction

This document outlines the final project guidelines for IS 584: Deep Learning for Text Analytics. In this project, you will formulate research questions based on the ASAP-Review dataset, perform a literature review, and investigate your questions using transformer-based deep learning models.

Specifically, you have to employ one of the following techniques:

- Fine-tuning an open-weight large language model
- Retrieval augmented generation
- Agent based framework

The project is divided into clear phases with specific deliverables, grading criteria, and submission guidelines. Please ensure that all work is original and that academic integrity is maintained throughout the project.

Dataset Description

The project is based on the ASAP-Review dataset which includes:

- **Conference Submissions:** Text versions of submitted papers.
- **Reviews:** Reviews for each submission in text format, self-reported confidence score in review and score given to submission.
- **Aspect Classes:** Information on the type of sentence (comparison, summary etc.) a given character or word belongs to for each review.
- **Final Decision and Metadata:** Whether the submission was accepted, link to the pdf with images and tables intact, and other metadata like title and authors.

It is essential to thoroughly inspect the dataset (and its README file) to understand its structure. It can be downloaded from this link. **Please note that we can aid you in scraping reviews from other journals/years.**

Project Phases and Timeline

There will be three phases for the term project. It is recommended that you start your project work as soon as possible. All of the project phases will be graded according to the guidelines in the grading section. Deliverables and due dates of each phase can be found below:

Phase	Deliverable	Due Date
Phase 1	Literature and Project Proposal	**27.04.2025**
Phase 2	Preliminary Results and Benchmarking	**18.05.2025**
Phase 3	Final Report and Git Repository	**24.06.2025**

Submission

At the end of each phase, upload your expanded report with links to your Git repository and Weights and Biases project. (Both of which should be public.) Only in the final phase you should additionally submit a copy of your repository as a zip file. Please use `.gitignore` to avoid comitting your dataset or other big files.

Deliverable 1: Project Proposal (20 pts)

Your proposal (maximum 2 pages, IEEE format) must have:

- **Literature Review (6 pts):** Survey at least three related research papers from reputable journals or conferences.[1] You should demonstrate an understanding of the field and identify potential gaps to investigate. Please follow the instructions in the appendix for reputability and citation formats.
- **Proposal (4 pts):** Develop at least two focused, creative, and feasible research questions.[2] At this stage, you should define which model architecture and framework will be used, at least two relevant performance metrics, expected outcomes and your approach's potential benefits.
- **EDA and Quality Checks (5 pts):** Conduct descriptive analyses and quality checks on the dataset. Report key statistics and data profiling results.

In addition, you must initialize a public Git repository and push **at least two commits** on different days with detailed comments. **(5 pts)** Make sure to follow reproducible and clean coding practices.[3]

Deliverable 2: Intermediate Report (20 pts)

Your final report (maximum 4 pages, IEEE format) must include:

- **Baselines (5 pts):** For the identified performance metrics, come up with simple baseline methods and evaluate them.

- **Preliminary Results (10 pts):** Your preliminary results should surpass the pre-defined baseline of the task, and accompanied by a code framework that is reproducible and documentation to explain your intention.
- **Benchmarking (5 pts):** Compare your initial results with the baseline. Then propose two different hyper-parameters to tweak to improve upon initial results. It is advised to start hyper-parameter tuning earlier on. Include statistical tests where applicable.

At this stage you should have **at least one more commit** to your Git repository, and initialized a **Weights and Biases** (WANDB) experiment page to log preliminary results.

Deliverable 3: Final Report (50 pts)

Your final report (maximum 6 pages, IEEE format) must include:

- **Approach and Evaluation (20 pts):** Your approach will be graded according to the benchmarking setup, experiment variety, results, and depth of implementation. If you have successfully adopted a transformers-based approach for the task, and achieved better results than both the baseline and preliminary results, you will earn full grades from your approach section.
- **Experiment Tracking (5 pts):** Your experiments should be tracked and reported via WANDB. Your WANDB experiments page or reports page should be publicly available. If it is not you will not get any points from experiment tracking.
- **Interpretability (10 pts):** Investigate why your final model gives certain outputs with appropriate techniques based on the architecture. You can visualize attention matrices of certain layers, use LIME or SHAP methods, or refer to the Captum library for visualizing gradient information. This section is left intentionally vague as your research question will dictate which parts of the model architecture need to be probed.
- **Documentation (15):** Proper documentation should clearly explain all steps you have taken. Your final report should have the following parts: abstract, introduction, dataset, modeling, evaluation, results, discussion, and conclusion.

Creativity / Novelty: We appreciate the hard and creative work that you put in your project. If you demonstrate the work is novel/creative (based on literature etc.), you will be awarded with a **bonus** up to **10 pts**.

Deliverable 4: Git Repository (10 pts)

Submit a link to your public Git repository along with an offline backup.

- Ensure the repository has a clear folder structure (e.g., `data/`, `notebooks/`, `source/`, `reports/`, `figures/`), a comprehensive `README.md`, and a `requirements.txt` file.
- Record at least five detailed commits made on separate days with proper messages and issue tracking.

Appendices

Developing Research Questions

A good research question should be:

- **Clear and focused.** In other words, the question should clearly state what the writer needs to do.
- **Not too broad and not too narrow.** The question should have an appropriate scope. If the question is too broad, it will not be possible to answer it thoroughly within the word limit. If it is too narrow you will not have enough to write about and you will struggle to develop a strong argument (see the activity below for examples).
- **Not too easy to answer.** For example, the question should require more than a simple yes or no answer.
- **Not too difficult to answer.** You must be able to answer the question thoroughly within the given timeframe and word limit.
- **Researchable.** You must have access to a suitable amount of quality research materials, such as academic books and refereed journal articles.
- **Analytical rather than descriptive.** In other words, your research question should allow you to produce an analysis of an issue or problem rather than a simple description of it.

We urge you to check these links as they will guide your question asking and literature search:

- Developing a research question [7:23] University of Melbourne, Academic Skills Youtube Series
- How to Create Research Questions & A Literature Review [54:57] METU Academic Writing Center by Zeynep Ünlüer
- How to Write a Research Question (George Mason University)
- Identifying a Research Problem and Question, and Searching Relevant Literature (Daniel J. Boudah, 2011)

Literature Survey and Citation Guidelines

You should not choose random papers returned from your query in Google Scholar. These papers should have been published in a respectable journal or conference proceeding. Some tell-tale signs of a good publication are that it might have been:

- Indexed by Web of Science or Scopus
- Published by respectable publishers such as IEEE, ACM, Elsevier or AAAI

- Appeared in a conference/journal that is indexed by ScienceDirect or Scimago Journal & Country Rank

You should especially cover high impact journals that are indexed in Google Scholar Metrics (Artificial Intelligence).

For in-text citations you should use the IEEE format. (Latex is useful to handle citations easily.) You are also expected to reference them properly in the end-text in IEEE format. All the fields should be complete and proper. Do not rely on what returns from Google Scholar as they are often missing. Although some appear to be published in Arxiv, they might have been published in a respectable journal or conference afterwards. Hence you need to update the details accordingly.

If you cite your sources incorrectly or didn't use a paper from a good resource (such as one from a predatory journal), you will lose grades accordingly.

Project Paper Guideline

- The final report should be in IEEE format. You can find the template https://template-selector.ieee.org/ (choose publication type: Conferences, Original Research, Word or Latex format).
- Your report must include a minimum of one graph and table. Maximum number for both is four. Make sure the graphs you provide are exported at 300 dpi and not copied and pasted. Failing to do so, or not staying within limits will result in grade reductions.
- Your results must be shown in tables or other appropriate structures. Do not copy and paste code or outputs from a notebook for the paper. This will result in grade deductions.
- Make sure your graphs, tables, and other material are legible, i.e. readable. Make sure different parts of your graphs (like lines, boxes, points) can be identified by their shape as well as color. (Like with changing line types, markers etc.) Treat this as if this paper would be printed out in black and white.
- Your report will also be evaluated based on adherence to the required format, correctness and sufficient explanation of each section, creation of clear and accurate captions for figures and tables, and references being written completely, accurately, and in the correct format.

General Guideline

- The projects should be prepared on an individual basis. You are not allowed to work together. You are expected to follow academic integrity rules and the generative AI conditions.
- Your whole analyses should be connected/related to each other. You should not do something for the sake of doing it. For instance, if you do apply a feature engineering technique and do not use the results in somewhere else or do not relate it to your research questions, your mark will be degraded accordingly.

- Storytelling: You are expected to conduct your analysis by referring to your research questions. You can pose sub research questions. It is important to tell your story in an easy to follow manner. Check for any spelling or grammar mistakes before the submission of your report. You can use Grammarly or Outwrite for this.
- The deadlines are strict. Hence, start working on your project and documentation as soon as possible. You can't finish it if you leave it to the last minute.

Academic Conduct and Generative AI

Students are expected to uphold the highest standards of academic integrity in all aspects of this project. The use of generative AI tools (e.g., ChatGPT, DALL·E, Copilot, etc.) is permitted under the following conditions:

- Ensure that code generation does not introduce security vulnerabilities and licensing conflicts.
- AI-generated code must be thoroughly reviewed, tested and understood by the author.
- Any code snippet, that is used as-is, should be referenced by either the original contributor or the AI-generator.
- AI-generated visualization should be reviewed and checked with the original data.
- Use of AI tools should be transparent. All instances of AI tools should be explicitly disclosed in the documentation (project report).

You **can** use AI generation tools for the following:

- **Debugging and optimization:** Errors are common when working with state-of-the-art and cutting-edge development. Fixes and solutions for different environments can be present in the open-source domain and AI tools are suitable for finding a solution (akin to searching through stack-overflow).
- **Spell and grammar checking:** AI tools can be used to check for spelling and grammar errors.
- Code suggestions and completions. AI tools such as ChatGPT and Copilot are suitable for code completion as long as the author knows what and how to implement them.
- **Visualization creation:** Creating visuals from existing data with AI tools can be productive and labor-saving.
- **Generative data augmentation:** AI tools can be employed for generating synthetic data which is
- **Textual drafts:** You can generate drafts with AI to adapt to a template.

You **should not** use AI generation tools for the following:

- **Creative content generation:** AI tools are infamous for plagiarism and copyright violations.
- **Finishing entire projects solely with AI tools:** Using AI to complete entire implementation tasks without understanding undermines the purpose of the projects and assignments. The projects and assignments are designed to be instructive, real world tasks are often much harder, complex and time-consuming.
- **Review literature completely with AI:** AI tools can be used for summarizing or finding relevant content, however, using only AI for literature review can introduce inherent biases to your work. More importantly, AI tools can hallucinate with the literature content, often leading to irrelevant or sometimes non-existent content.
- **Substitute yourself with AI tools:** Human creativity, consistency and character are irreplaceable with any AI tool, you should not diminish your effort and contribution by replacing yourself with any AI tool. However, it is beneficial to use AI tools as productivity tools to augment your work.

Reproducible Code

To ensure reproducability of you work, please follow the guidelines outlined in **Alan Turing Institute's Reproducible Project Template**. Where your project exceeds this structure, feel free to create new folders and document them in the README markdown file of the repository. Below are some key points:

- The README file should be informative enough to follow your folder structure with ease.
- There must be a requirements.txt file that documents your python environment with package names and versions.
- All figures, notebooks, scripts and preprocessed data should be in relevant folders.
- Do not upload raw data to a git remote, but preprocessed and compressed versions are fine.
- To learn more about code reproducibility, you can watch this presentation. Additionally, you can read the following documents: Organizing Your Projects and Reproducible Research: Goals, Guidelines and Git.
- If you are not familiar with Git and GitHub, you can take following courses on Datacamp: Introduction to Git and Introduction to GitHub Concepts. Please reach out to your TA for access to these courses.
- Even though all the details of your project will be available in your GitHub repository, you must submit the final report, a link to your git repository, and an offline copy of your repository through ODTUClass.

Late Submission

Each phase may be submitted up to 48 hours late at most. After this 2-day period, the submission system will close. You will get 10-point penalty for every day of late submission.