CS510 Advanced Information Retrieval Project Proposal

- **Title:** ReSearchy: Guarding Novelty, Guiding Ideas
- **Track**: Development Track
- **Member**:
  - Shilong Li (sli148@illinois.edu)
  - Yixuan Li（yixuan19@illinois.edu)
  - Ya-Ting Pai (yatingp2@illinois.edu; coordinator)
  - Xuanming Zhang (xz130@illinois.edu)

- [**Functions and Users**] We plan to develop a platform and a plugin tool for researchers, including professors and students, as an extension to existing writing platforms. The tool offers two key features:
  - **Idea Overlap Detection** – When users input their research idea or abstract, the tool automatically searches recent papers in the relevant field and displays potentially overlapping titles and abstracts.
  - **Content Comparison** – Users can select any retrieved paper to compare with their own input, with overlapping parts highlighted to help identify similarities. This assists researchers in avoiding redundant work early in the ideation process and supports refinement or innovation based on existing literature.

- [**Significance**] One of the biggest pain points in academic research is that after investing a lot of time, you find that similar results have already been achieved. ReSearchy can help avoid ineffective work by providing early warning of overlapping research content. Existing literature review methods are not only inefficient, but also often miss important literature due to differences in term expression. Our technology can accurately overcome this limitation. Our tool automatically finds problems by semantically matching ideas to existing literature.

  With ReSearchy, the research process becomes more efficient because academic resources will focus on real knowledge gaps rather than duplicating work. This meets society's need to advance the frontiers of knowledge rather than reinvent existing work.

  For senior researchers and students, the tool helps identify where real contributions lie, making academic progress more efficient and education more effective.

- [**Approach**] ReSearchy will be developed as a standalone web tool, with plans to support plugin integration in the future, such as a Chrome extension. Users can input their research ideas on the platform to receive semantically matched related papers and content comparison results.

  Python (FastAPI) will be used for the backend and **React.js** for the frontend, combined with pre-trained language models from **Hugging Face** (like BERT or Specter) for semantic matching.

Literature search will be based on open-access databases (arXiv), using **FAISS** or **Elasticsearch** for efficient semantic retrieval.

Open-source models and datasets will be used to quickly build a prototype and avoid developing NLP algorithms from scratch.

The main risks include semantic matching accuracy and limited data sources, which can be mitigated by fine-tuning models and focusing on open-access datasets first.

Data:

- arXiv: https://info.arxiv.org/help/api/basics.html
- Sentence-transformer: https://www.sbert.net/docs/pretrained_models.html
- BERT: https://huggingface.co/bert-base-uncased
- Specter: https://huggingface.co/allenai/specter

- [**Evaluation**] We evaluate ReSearchy using the CHEAT dataset, which contains human-written and ChatGPT-generated academic abstracts.
  - **Idea Overlap Detection**: We test if ReSearchy can retrieve the corresponding synthetic abstracts when given a human-written abstract. We report Recall@K and Precision@K.
  - **Content Comparison**: We compare the input abstract and retrieved abstract to check if ReSearchy correctly highlights overlapping content. We use ROUGE, BERTScore, and human evaluation.
- [**Timeline**]
  - **March 29** – Team formation and project proposal submission
  - **April 3** – Finalize project requirements and architecture design
  - **April 10** – Implement core functionality: abstract input & paper retrieval
  - **April 17** – Develop content comparison and highlighting feature
  - **April 24** – Integrate UI with backend, basic testing
  - **May 1** – Conduct user testing and gather feedback
  - **May 5** – Final polish, bug fixes, and documentation
  - **May 10** – Submit final project report
  - **May 11** – Submit project presentation recording
  - **May 13** – Final project presentation via Zoom
- [**Task division**]
  - Ya-Ting Pai is responsible for system architecture design and implementing the paper retrieval module.
  - Shilong Li focuses on developing the semantic comparison and highlighting features using NLP techniques.
  - Yixuan Li handles project coordination, front-end development, and user testing design.
  - Xuanming Zhang is in charge of system integration, testing, and supporting final deliverables.