Bachelor Projects 2025

Data Intensive Systems Group

The below are possible project proposals that can include more than one project.

1.EnSud: Entity Summarization Goes South

Advisor: *Davide Mottin* (*davide@cs.au.dk*), *Akhil Arora* (*akhil.arora@cs.au.dk*) and *Ira Assent* (ira@cs.au.dk)

Entity summarization [1] is the problem of finding a small and descriptive version of an entity. For instance, if we want to provide a short description of Denmark we could say that Denmark is a country, located in Europe, part of Scandinavia, with the capital Copenhagen. Entity summarization studies the problem in graphs where each node is an entity (Copenhaghen, Denmark) and each connection a relationship (capital of).

A number of methods have been proposed, but none of them have a clear summarization objective. This project delves into this issue from two separate angles.

Potential projects include:

1. [Practical] Experimentally evaluate and reimplement a number of methods on the existing benchmark [2] and our recent dataset based on Wikipedia.
2. [Theory and practice] Implement and improve a new method based on a novel probabilistic objective.

Tasks:

- Reading and understanding the main concepts of Entity summarization (start from [1])
- Familiarize with the current methods and our implementation.
- Implement a new method OR reimplement existing ones (see potential projects)
- Evaluate the existing baselines on our Wikipedia-based dataset.
- Report the results

Related work:

[1] https://sites.google.com/view/entity-summarization-tutorials/www2020
[2] https://github.com/nju-websoft/ESBM

2. OANa: Old algorithms, new hardware

Advisor: *Cheng Huang (cheng@cs.au.dk), Davide Mottin (davide@cs.au.dk), Ira Assent (ira@cs.au.dk)*

In recent years, we have witnessed substantial breakthroughs in Machine Learning, but the most powerful models are computationally demanding. As a result, much of the computation makes use of parallel algorithms on graphics cards, GPUs, and multi-core CPUs.
While efficient parallel algorithms for problems such as matrix multiplication exist, graph analysis (for data from social networks, protein networks, etc) is challenging to parallelize. Thus, old fashioned algorithms, such as shortest paths, linear assignment, need a restyling to work fast on GPUs. An example project is the parallelization of algorithms, such as methods for community detection in social networks [2].

Requirements: Interest in programming at low level and learn a number of tricks that apply to GPUs.

Tasks:

- Study and run simple programs on GPUs
- Reimplement a simple algorithm in a naïve manner for GPUs
- Optimize the algorithm to beat the fastest GPU algorithm
- Make experiments and report

References:

[1]
https://www.cherryservers.com/blog/introduction-to-gpu-programming-with-cuda-and-python#:~:text=%23What%20is%20GPU%20Programming%3F,(GPGPU%20computing)%20as%20well.

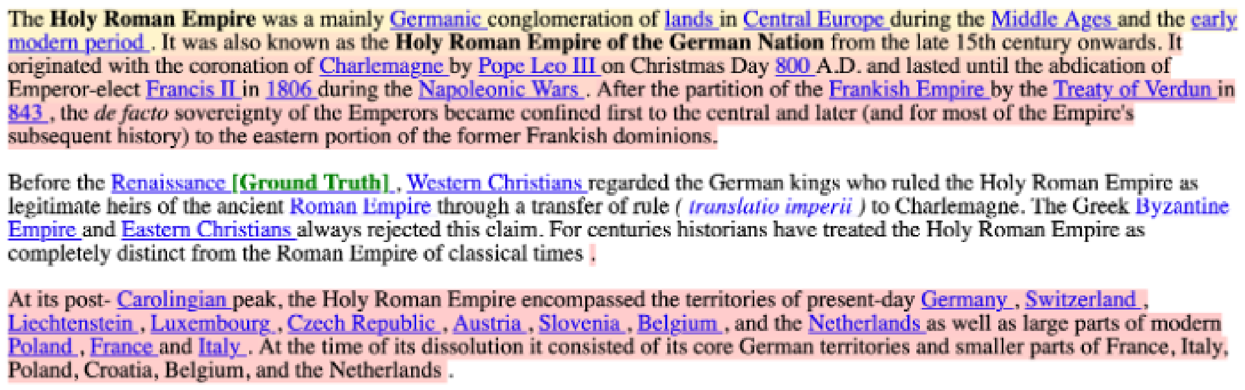
[2] https://link.springer.com/chapter/10.1007/978-3-319-32049-6_14

3. Entity-insertion recommender for Wikipedia articles

Advisor: *Akhil Arora* (akhil.arora@cs.au.dk)

Given a source and target Wikipedia page, we have devised ML models [1] by fine-tuning pre-trained language models (e.g. BERT, RoBERTa) to predict **where** in the source page a link to the target page should be inserted. The task of predicting where to insert a link to a new entity is essential to operationalize link recommendations at the Wikipedia scale.

Tasks:
The goal of this project is two-fold:
(1) to develop an API for extracting entity insertion recommendations from the already developed fine-tuned language models, and
(2) to develop a Web interface to showcase the extracted recommendations to the end user by overlaying a heatmap on top of the Wikipedia article. Following is a mock-up of the interface with an overlaid **heat map** that we would like to build:



The **Holy Roman Empire** was a mainly Germanic conglomeration of lands in Central Europe during the Middle Ages and the early modern period . It was also known as the **Holy Roman Empire of the German Nation** from the late 15th century onwards. It originated with the coronation of Charlemagne by Pope Leo III on Christmas Day 800 A.D. and lasted until the abdication of Emperor-elect Francis II in 1806 during the Napoleonic Wars . After the partition of the Frankish Empire by the Treaty of Verdun in 843 , the *de facto* sovereignty of the Emperors became confined first to the central and later (and for most of the Empire's subsequent history) to the eastern portion of the former Frankish dominions.

Before the Renaissance **[Ground Truth]** , Western Christians regarded the German kings who ruled the Holy Roman Empire as legitimate heirs of the ancient Roman Empire through a transfer of rule (*translatio imperii*) to Charlemagne. The Greek Byzantine Empire and Eastern Christians always rejected this claim. For centuries historians have treated the Holy Roman Empire as completely distinct from the Roman Empire of classical times .

At its post- Carolingian peak, the Holy Roman Empire encompassed the territories of present-day Germany , Switzerland , Liechtenstein , Luxembourg , Czech Republic , Austria , Slovenia , Belgium , and the Netherlands as well as large parts of modern Poland , France and Italy . At the time of its dissolution it consisted of its core German territories and smaller parts of France, Italy, Poland, Croatia, Belgium, and the Netherlands .

Expected outcome:
- A working user interface (cf. [2] for an example developed by me when I was as student)
- A demo paper submission

Required skills:
- Strong programming skills
- Experience with Web development: knowledge of the full-stack: front and back-end frameworks, e.g., JavaScript, Node.js, React, CSS, HTML, SQL, etc.
- A creative mind with an inquisitive attitude.
- Experience with PyTorch and Vector databases is a plus

Related work:
[1] Arora et al. Entity Insertion in Multilingual Linked Corpora: The Case of Wikipedia, EMNLP' 24

[2] https://quotebank.dlab.tools/

4. CacheSaver: Efficient, Cost-Effective Experimentation with Large Language Models

Advisor: *Akhil Arora* (akhil.arora@cs.au.dk)

CacheSaver is a high-efficiency caching tool for experiments using large language models (LLMs), designed to reduce costs, ensure reproducibility, and streamline debugging. By caching multiple responses per query and tracking usage, CacheSaver enables unique sampling across experiments while preserving full statistical integrity. Built on top of Python's asyncio, it supports asynchronous execution, batching, prompt deduplication, and race-condition prevention.
Note that we intend to hire at least three groups for this project, and each group will work on implementing tasks for different AI reasoning frameworks.

Tasks:

- Familiarize yourself with CacheSaver's code [1], features, and mechanics.
- Familiarize yourself with one AI reasoning framework. Some recommended frameworks are [2] and [3]. Each group will work on one framework.
- Familiarize yourself with tasks (e.g., Game of 24 [2], Crosswords [2], WebShop [4]). We will work with ~10 tasks, others will be communicated close to the start of the project.
- Using a single AI framework, perform extensive experiments and ablation studies on the tasks to show the benefits of CacheSaver.
- Ideally, experiments will be repeated using multiple LLMs (LLaMa, GPT-4, Claude, etc.).

Expected outcome:

- Clean code and documentation of the implemented tasks and AI reasoning frameworks.
- Successful integrations of the tasks/frameworks into our CacheSaver [1] framework.

Required skills:

- Strong programming skills and knowledge of Python, multi-threading, async execution.
- A creative mind with an inquisitive attitude.
- Experience with LLMs and prompting is a plus.

Related work:

[1] CacheSaver Github repository
[2] Fleet of Agents: Coordinated Problem Solving with Large Language Models using Genetic Particle Filtering
[3] Tree of Thoughts: Deliberate Problem Solving with Large Language Models
[4] WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents

5. Reflective Fleet of Agents

Advisor: *Akhil Arora* (akhil.arora@cs.au.dk)

Fleet of Agents (FoA) [1] is a framework inspired by genetic particle filtering where large language models (LLMs) act as autonomous agents exploring a search space. During a *mutation phase*, each agent independently moves to a new state guided by a policy, while a heuristic value function assesses the potential of each state. After several steps, a *selection phase* refocuses agents on high-value areas, resampling states based on their evaluation scores.

This project aims to enhance the FoA framework by adding a reflective agent that reattempts high-rated states that previously failed. Unlike regular agents, the reflective agent's goal is to analyze past failures and use this information to retry and refine those states. The student group will explore Reflexion techniques and test how they can improve the search process within the FoA framework.

Tasks:

- Familiarize yourself with the FoA framework [1].
- Familiarize yourself with the Reflexion framework [2].
- Introduce a reflective agent to the FoA framework.
- Tune the hyperparameters of the new reflective FoA framework.
- Conduct a thorough and extensive empirical evaluation by repeating the original experiments to demonstrate the improvements brought by the addition of the new reflective agent.

Expected outcome:

- Clean code and documentation of the implemented functionality.
- Successful integrations of the feature into our FoA [1] framework.

Required skills:

- Strong programming skills and knowledge of Python.
- Proficient at reading and understanding large codebases.
- A creative mind with an inquisitive attitude.
- Experience with LLMs and prompting is a plus.

References:

[1] Fleet of Agents: Coordinated Problem Solving with Large Language Models using Genetic Particle Filtering
[2] Reflexion: Language Agents with Verbal Reinforcement Learning
[3] CacheSaver Github repository

6. Explainable AI

The primary objective of this project is to develop and evaluate algorithms that enhance the explainability of machine learning or data mining models. As these models become increasingly complex, it is crucial to make their predictions interpretable to non-technical users, domain experts, and regulators to build trust and ensure ethical usage. This project aims to assess algorithms that identify influential features, visualize decision boundaries, and quantify the reliability of model predictions.

Tasks:

- Literature Review of current algorithms and methods in explainable AI, with a focus on evaluation metrics
- Develop/implement algorithms for explainability, such as feature attribution methods (e.g., SHAP [2], LIME[1]) or methods for generating counterfactuals. Possibly design new variations or optimizations
- Evaluate on standard datasets (e.g., MNIST, CIFAR-10) using quantitative metrics, such as fidelity or stability, possibly including new metrics or evaluation approaches
- Document algorithms, metrics, and evaluation process, including insights into how to improve interpretability and a discussion on metric effectiveness.

Expected outcome:

- An implementation of one or more explainability algorithms models
- A quantitative assessment based on metrics like fidelity, or stability
- A comparative analysis of any developed methods against existing ones
- A discussion of notions of explainability and its metrics, with suggestions for future improvements

References:

1. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
2. Lundberg, S. M., & Lee, S.-I. (2017). "A Unified Approach to Interpreting Model Predictions." Advances in Neural Information Processing Systems (NeurIPS).
3. Vilone, Giulia, and Luca Longo. "Notions of explainability and evaluation approaches for explainable artificial intelligence." Information Fusion 76 (2021): 89-106.
4. Nauta, M., et al. (2023). From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. ACM Computing Surveys, 55(13s), 1-42.

7. Graph correspondences

Advisor: *Konstantinos Skitsas (skitsas@cs.au.dk)*, *Davide Mottin* (davide@cs.au.dk)

Graphs are structures where a set of points are connected through some relationships. For instance, a social network is a graph of people connected by friendship or collaboration relationships. Many problems require finding, in two graphs, which nodes correspond to one another. This problem has several names, depending on the type of correspondences and the size of the graph. You might have heard of subgraph matching, graph matching, or graph alignment. All of them have very similar formulations but vary with the final goal.

This project will explore some of the problems we have been working on in the past few years.

Examples projects:

5. [If you like implementation] Construct a benchmark and library to evaluate current graph alignment algorithms including several datasets. Our preliminary evaluation was [1].
6. [If you like improving algorithms] Can we push our FUGAL [2] algorithm further by modifying some of its characteristics?
7. [If you like the design of algorithms] Can we add “Diversity” [3] when we search for a subgraph in another graph?

Tasks include:

- Study the related work and our previous solutions
- Implement previous algorithms or improve the existing ones
- Run experiments on available data and queries

References:

[1] Experimental evaluation of graph alignment algorithms
[2] FUGAL
[3] Diversified Top-K subgraph Querying in a Large Graph

8.An Empirical Study and Analysis of Generalized Zero-Shot Learning for Graph Machine Learning

Advisor: *Zhiqiang Zhong* (zzhong@cs.au.dk), Akhil Arora (akhil.arora@cs.au.dk), *Davide Mottin* (davide@cs.au.dk)

Generalized Zero-Shot Learning (GZSL) involves scenarios where the inference step of a pre-trained machine learning model must handle both seen and unseen class labels. This requires the model to effectively generalize to new data while retaining essential knowledge from previously seen classes. GZSL presents a greater challenge compared to Zero-Shot Learning (ZSL), where the inference is restricted to only unseen class labels. Although GZSL has been extensively studied in computer vision, its application in Graph Machine Learning (GML) remains largely unexplored.

This project plans to

1. Understand the concepts of ZSL and GZSL through existing literature in computer vision [1].
2. Extend the GZSL concept to GML and design a benchmark for evaluation.
3. Implement the designed benchmark and evaluate representative Graph Neural Network (GNN) models [2].

Expected outcomes:

1. An open-source evaluation benchmark for GZSL in GML.
2. A publication.

Related work:

[1] An Empirical Study and Analysis of Generalized Zero-Shot Learning for Object Recognition in the Wild

[2] A Comprehensive Survey on Graph Neural Networks

9. Graph Mining Techniques for Large-Scale Datasets

Advisor: *Anders Sandholm* (anders@sandholm.dk), Akhil Arora (akhil.arora@cs.au.dk), and *Ira Assent* (ira@cs.au.dk)

This project aims to develop and scale graph mining techniques to large-scale datasets, for tasks such as entity reconciliation to optimize data pre-processing for large model training, or Trust and Safety applications, such as detecting anomalous behavior or malicious content. The project will explore advanced graph-based clustering methods to group related entities, facilitating more efficient training or filtering of data.

Tasks:

- Conduct an in-depth review of graph mining techniques for a particular task (found in discussion with us), focusing on methods suited for large-scale data.
- Select or create synthetic large-scale datasets relevant to the task
- Develop or adapt graph mining algorithms for the task
- Test the algorithms on selected datasets, evaluating the clustering quality and performance on the task
- Describe and discuss the task definition, algorithms, dataset curation and evaluation process and results

References:

- Kirielle, N., Christen, P., & Ranbaduge, T. (2023). Unsupervised graph-based entity resolution for complex entities. ACM Transactions on Knowledge Discovery from Data, 17(1), 1-30.
- Christophides, V., Efthymiou, V., Palpanas, T., Papadakis, G., & Stefanidis, K. (2020). An overview of end-to-end entity resolution for big data. ACM Computing Surveys (CSUR), 53(6), 1-42.
- Ma, X., et al. (2021). A comprehensive survey on graph anomaly detection with deep learning. IEEE Transactions on Knowledge and Data Engineering, 35(12), 12012-12038.

10. Can Bandits estimate Cardinality?

Advisor: *Kasper Overgaard Mortensen (km@cs.au.dk), Davide Mottin (davide@cs.au.dk)*

Traditional cardinality estimation relies on database statistics to find the best query execution plan to minimize the overall execution cost. In recent years we have seen a growing interest in learned cardinality estimation [1], where a deep learning model is trained to produce reliable cardinality estimation. Yet, such models struggle to handle non-stationary environments where query workloads change over time, and State-of-the-art [2] that tackles drifts still maintains a costly training overhead and does not adapt dynamically.

This project seeks to reformulate learned cardinality estimation as a Multi-armed bandit problem [3] with expert advice, where experts consist of traditional DBMS statistics-based estimation, and a selection of multiple smaller learned models. ([1]+[2]+Bandit). Doing so could enable reliable learned estimation, without system downtime, that automatically switches to a new model, or to traditional techniques, when a workload shift is detected.

Tasks include:

- Study the related work and get familiar with Multi-armed bandits
- Run previous methods for cardinality estimation
- Generate an adversarial workload
- Implement bandits for cardinality estimation
- Run experiments on available data and queries

Related work:

[1] Estimating Cardinalities with Deep Sketches
[2] Robust Query Driven Cardinality Estimation under Changing Workloads
[3] Introduction to Multi-armed bandits

11. Generative Quotation Extraction from Online News using Large Language Models

Advisor: *Akhil Arora* (akhil.arora@cs.au.dk)

Quotations are an important source of primary information that can be extracted from news articles, and access to a repository of such speaker-attributed quotations stands to benefit a multitude of linguistic, social, journalistic, and psychological investigations. In previous work, we have prepared Quotebank [1], a dataset of 235 million unique, speaker-attributed quotes that were extracted from 196 million English news articles published between September 2008 and April 2020.

In this project, the student group will explore the ability of large language models (LLMs) to extract quotation-speaker pairs from online news articles using zero- and few-shot prompting.

Tasks:

- Familiarize yourself with QuoteBank's code, schema, and formatting [1].
- Familiarize yourself with zero- and few-shot prompting with LLMs.
- Extensive evaluation of the quotation extraction ability of LLMs
- Ideally, experiments will be repeated using multiple LLMs (LLaMa, GPT-4, Claude, etc.).

Expected outcome:

- Clean code and documentation of the implemented Quotation extract pipeline..

Required skills:

- Strong programming skills and knowledge of Python, multi-threading, async execution.
- A creative mind with an inquisitive attitude.
- Experience with LLMs and prompting is a plus.

Related work:

[1] Vaucher et al., Quotebank: A Corpus of Quotations from a Decade of News, WSDM'21