Bachelor projects 2024

Data Intensive Systems group

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

1. Relating density- and center-based clustering methods

Advisor: *Andrew Draganov* (andrew.draganov@cs.au.dk) and *Ira Assent* (ira@cs.au.dk)

What does it mean for two things to be alike? We can say that ten cars are similar because they look like another car we've seen. But we could also say that these ten cars are similar because each one is a slight variation on the one before it. This is the essential difference between center-based clustering (grouping things that are similar to a prototype) and density-based clustering (grouping things that are similar to one another pairwise).

It was recently shown in [1] that under a specific setting, these center- and density-based methods are actually giving the same output. Specifically, the facility location center-based clustering method and the DBSCAN density-based method match when you reinterpret what 'distances' mean. This raises the question of how far this relationship goes. Are most of the density-based methods just reinterpretations of the center-based methods? In this project we will study exactly that question.

The project requires interest in discrete math/graph theory, optimization, and programming.

Related work:

[1] – https://dl.acm.org/doi/pdf/10.1145/3580305.3599283

2. GEnS: Entity Summarization meets GNNs

Advisor: *Atefeh Moradan (atefeh.moradan@cs.au.dk), Davide Mottin* (davide@cs.au.dk),

Anders Sandholm (anders@sandholm.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 to solve the problem [1], but none of them consider the graph a first-class-citizen, nor do they have appropriate benchmarks. We are currently building a new method that uses Graph Neural Networks [3], special Neural networks for graphs to overcome the limitations of previous methods [link available once the paper is ready].

Potential projects include:

1. Extending the current method including more expressive GNNs (e.g., GAT).
2. Improve the scalability of the method on large graphs

Tasks:

- Reading and understanding the main concepts of Entity summarization (start from [1])
- Familiarize with GNNs and our current method
- Change the method to improve the scalability or using more expressive GNNs.
- Evaluate the method against existing baselines (already implemented), and tune the hyperparameters.
- Report the results

Related work:

[1] https://sites.google.com/view/entity-summarization-tutorials/www2020
[2] https://github.com/nju-websoft/ESBM
[3] https://distill.pub/2021/gnn-intro/

3. OANa: Old algorithms, new hardware

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

Recent years, due to Machine Learning, many companies have been producing chips that go beyond the traditional GPUs and CPUs. One such a chip is the Intelligence Processing Units (IPUs) produced by Graphcore.ai [1]. An IPU can be thought of as a set of CPUs, called tiles, all having a tiny memory. As such, any tile can be almost completely independent of all the others. Yet, the old fashioned algorithms, such as shortest paths, linear assignment, need a restyling to work fast on IPUs.

We have been lucky enough to have access to some of these machines, thanks to a collaboration with Graphcore, and now we are looking for eager students willing to take very simple algorithms and revamp them to IPUs so that our algorithm becomes faster than that of GPUs.

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

Tasks:

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

References:

[1] https://www.graphcore.ai/bow-processors

4. Automatically labeling data for classifying rhetorical appeals

Advisor: *Amalie Brogaard Pauli (ampa@cs.au.dk), Ira Assent (ira@cs.au.dk)*

This project is within natural language processing and the aim is to detect rhetorical appeals in texts. The long-term aim of detecting rhetorical appeals is to understand and identify misinformation.
We want to construct a text dataset containing labels to train machine learning models for detecting rhetorical appeals on. We approach this by writing label functions through the Snorkel framework and by using state-of-the art Transformer models for zero-shot classification. The idea is to write multiple functions to (roughly) identify and predict the labels e.g. using different models. The advantage of this is the fact that with multiple predictions we can adjust the noise from each labelling function.
We also need to create some ground-truth labels by humans for testing and evaluation.
The data can be in Danish or English.

Tasks:

- Preprocess the data source
- Discuss and define guidelines for annotations
- Annotate a part of the data to use in the evaluation
- Get familiar with the Snorkel package (python) and run experiments
- Devise and write different label functions to build a dataset
- Evaluate, discuss and reflect on the results

References:

https://cs.brown.edu/people/sbach/files/ratner-vldb17.pdf
https://www.snorkel.org/use-cases/01-spam-tutorial
https://en.wikipedia.org/wiki/Modes_of_persuasion

5. Elastic Graph Indexing

Advisor: *Konstantinos Skitsas (skitsas@cs.au.dk)*, *Davide Mottin* (*davide@cs.au.dk*), Panos Karras (piekarras@gmail.com)

Subgraph matching seeks all isomorphic occurrences of a query subgraph within a large graph and it finds applications in chemistry, bioinformatics, social network, computer vision, pattern recognition, and many others. As it reduces to subgraph isomorphism, it is NP-complete. Therefore, the current solutions overcome the theoretical impediment with a practical approach that reduces the computation by filtering unpromising candidates.

Many algorithms focus on building an offline, static [1] index based on the data graph, so that when the query arrives they will quickly prune unpromising candidates. The remaining candidates will be passed to a backtracking algorithm to return the matchings. Other algorithms focus on improving the isomorphic check. One way to enhance the backtracking algorithm is to change the traversal order of visit of query vertices [4]. Another way to improve the backtracking algorithm is to add an additional filtering step before exploring a branch of a selected vertex, to remove a vertex before exploring its branch [5].

In this project we can explore different directions, such as rendering current indexes (1) adaptive to the queries [3] to answer future queries fast or (2) more structural aware [2], (3) improving the backtracking order, or (3) caching the results to improve the speed.

Tasks include:

- Study the related work and our previous solution and framework
- Implement simple strategies to improve the time
- Run experiments on available data and queries

The project requires an interest in math and graphs.

[1] https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6228085
[2] https://dl.acm.org/doi/10.1145/1353343.1353369
[3] http://vldb.org/pvldb/vol5/p502_felixhalim_vldb2012.pdf
[4] https://dl.acm.org/doi/pdf/10.1145/3318464.3380581
[5] https://dl.acm.org/doi/abs/10.14778/3587136.3587144

6. Large Language Model-augmented Graph Neural Network Pre-training

Advisor: *Zhiqiang Zhong* (zzhong@cs.au.dk), *Davide Mottin* (davide@cs.au.dk), *Anders Sandholm* (anders@sandholm.dk)

Graph Neural Networks have emerged as a powerful tool for analysing and extracting insights from graph-structured data, with applications spanning from social networks and knowledge graphs to intelligent drug discovery. While GNNs can be trained from scratch, pre-training GNNs to learn transferable knowledge for downstream tasks has recently been demonstrated to improve the state of the art [1]. However, the pre-training task definition and knowledge transfer schema design remain open questions. The project aims to explore the synergies between Large Language Models (LLMs) and GNNs for enhanced pre-training techniques [2], enabling more effective and efficient GNN models.

Tasks:

1. Familiarise yourself with existing GNN [3] and LLM frameworks [4].
2. Develop a strategy for integrating LLM-extracted knowledge into GNN pre-training.
3. Evaluate the performance of these models on various benchmark datasets and real-world applications.
4. Finalise the report.

Related Work:

[1] Learning to Pre-train Graph Neural Networks.

[2] Knowledge-augmented Graph Machine Learning for Drug Discovery: A Survey from Precision to Interpretability.

[3] PyTorch Geometric Documentation.

[4] VLLM Documentation.

7. Towards Effective Graph Representation Learning with the Guide of Label Correlation

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

Current graph representation learning models often make the assumption that differences between various node classes are consistent. This has led to the widespread adoption of representing true labels as One-Hot vectors as a standard practice. However, the One-Hot vectors may not adequately reflect the relation between the nodes and labels, as labels are often not completely independent, and harnessing this dependency could prove highly advantageous for graph representation learning.

While a similar concept has been explored in text classification [1], it is important to recognise that the characteristics in the context of graphs may differ. For example, the distances between nodes belonging to distinct classes can serve as potent indicators to uncover label correlations, subsequently enabling the differentiation of node embeddings for different classes. Hence, this project seeks to delve into the exploration of label correlations within graph data and apply the acquired knowledge to enhance downstream tasks, such as node classification.

Tasks:

1. Familiarise yourself with existing graph representation learning models [2,3] and label correlation learning methods [1,4].
2. Develop a strategy for learning label correlations on graphs to integrate it within a graph representation learning model.
3. Evaluate the performance of proposed models on various benchmark datasets and real-world applications.
4. Finalise the report.

Related work:

[1] Label Confusion Learning to Enhance Text Classification Models.

[2] A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications.

[3] PyTorch Geometric Documentation.

[4] Multi-Label Classification with Label Graph Superimposing.

8. Automatic Summarization of Meetings using Generative AI

Advisor: *Anders Sandholm* (anders@sandholm.dk) and *Ira Assent* (ira@cs.au.dk)

As online meetings become more and more common, recording a meeting is just a click away and advanced speech to text tools allow for easy transcription of what was said by whom in a meeting.

The real need, however, is often a succinct summary of what was discussed and decided in a meeting. Recent advances in Large Language Models (LLMs) has made the job of constructing a summary from the transcript a whole lot easier.

Potential questions to investigate and answer are:

- How do the different commercial and open source solutions currently in the market compare (e.g., MSFT Teams, Zoom, Google Meet, Supernormal, etc.)?
- Is the quality of the output summary good enough for everyday professional use?
    - What relevant evaluation metrics would you use to answer this?
- What key challenges / error types are most often occurring?
    - What experiments could be run, to find solutions to these issues (Data cleaning/upgrade, Prompt engineering, Fine-tuning, etc.)?
- Going forward, what next-generation ideas are there in this area? (E.g., video summarization, automated team feedback, etc.)
    - What would need to be or become true for these ideas to be viable?

Related work:

[1] Abstractive Meeting Summarization: A Survey | Transactions of the Association for Computational Linguistics | MIT Press (TACL'23 paper)

- https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00578/116993/Abstractive-Meeting-Summarization-A-Survey

[2] Training language models to follow instructions with human feedback (NeurIPS'22 - The RLHF paper from Open AI)

- https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html