Calculating Similarity Between EPFL Courses and Using It to Build a Graph

Master's Semester Project

Robert Injac

École polytechnique fédérale de Lausanne, Center for Digital Education
Supervisors: **Francisco Pinto, Patrick Jermann**

January 15, 2019

Introduction

Problem:

- EPFL has more than 1000 courses
- Navigating thorough those can be difficult
- Hard to find courses relevant to a course

My project:

- Developing a method for establishing similarity/relatedness between two EPFL courses.
- Constructing a graph of related EPFL courses

Data

Following datasets are used:

- **Course descriptions:** contains course name, description, summary, and other data for all EPFL courses
- **Course dependencies:** contains some EPFL courses for which professors said are similar
- **Course keywords:** contains keywords for each EPFL course
- **Course semesters:** contains in which semester is each course mostly taken (for example, Master 2. semester)

Problems: missing data, language mix-up

Pre-processing

For each EPFL course there is the following data:

- **Course name:** name of the course.
- **Course description:** around 50-100 words, description about what is the course about.
- **Course summary:** summary of the curriculum of the course
- **Course keywords:** important concepts learned in the course.

Using all 4 of that, tokenized and then concatenated into a single word list.

Standard pre-processing on this word list: convert to lowercase, removing stop words, removing punctuation.

Also, removing top 100 most common words found in all courses. **Why?**

Removing top 100 common words

We want each course word list to have as much as possible words related to the course, and the least possible amount of words which are not related to the course.

We can do that by removing words related to all courses (such as "course", "project", "theory", etc)

Type	Related sim.	Unrelated sim.	Difference
Normal	0.97402	0.86890	**0.10511**
Top 100 removed	0.96340	0.80567	**0.15773**
Human	0.94336	0.72045	**0.22291**

Table: Effectiveness of removing top 100 common words

Word embedding method

Word embeddings:

- name for a set of NLP techniques in which words or phrases from the vocabulary are mapped to vectors of real numbers.
- using *fasttext word2vec* embeddings: 1 million word vectors trained on Wikipedia[1]..

Similarity method: both courses are projected into the vector space and the cosine similarity is found between them.

Embedding of the course in the vector space: taking the average of word embedding vectors for each word of course.

[1] `https://fasttext.cc/docs/en/english-vectors.html`

Evalutation

We know which courses are truly similar to each other (for some courses) - the course dependencies dataset.

One hyperparameter, threshold: if the similarity is above the threshold, the courses are related, and if it is below, they are not.

Evaluation dataset consisting of 172 similar and 250 not-similar courses.

Using cross-validation: 70% for the train set and 30% for the test set

Accuracy	Precision	Recall	F1-score
0.81889	0.77777	0.79245	0.78504

Table: Results of the evaluation

Graph creation

The graph has the following structure:

- **Nodes:** EPFL courses
- **Edges:** two courses will share an edge if they are related (according to our similarity method)

The graph is **directed**. We have the information which courses are taken when (course semesters dataset), and the courses are only connected from earlier to later.

Visualization

I used the work of an EPFL student Raphaël Steinmann.

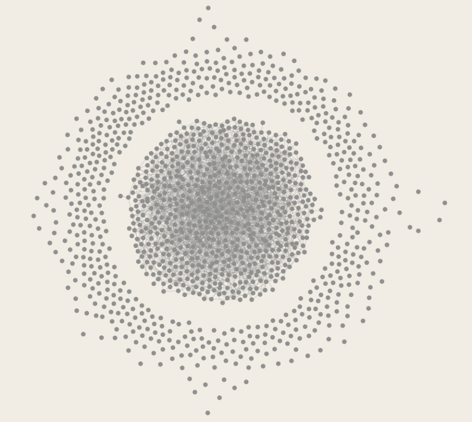
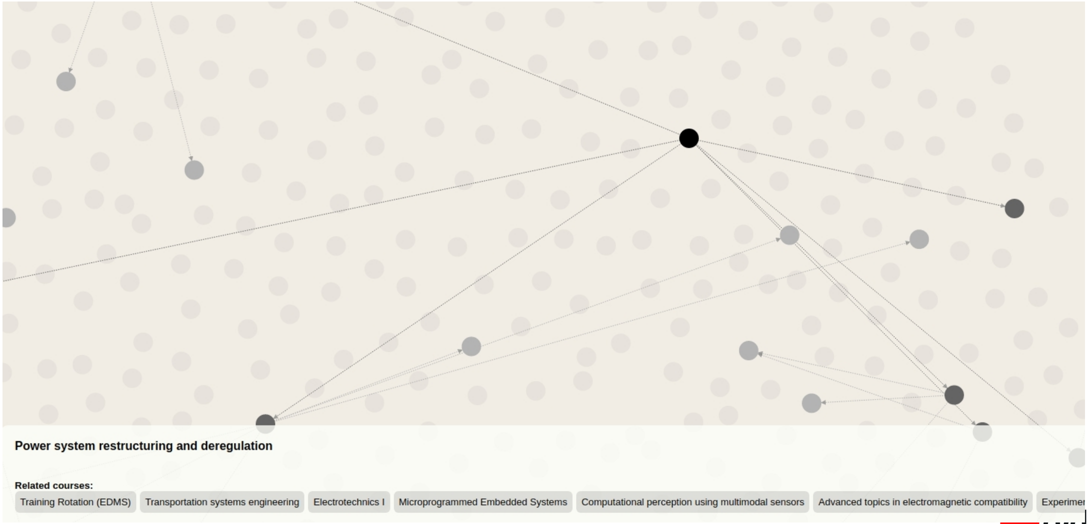


Figure: **Graph visualization**



Power system restructuring and deregulation

Related courses:

Training Rotation (EDMS) | Transportation systems engineering | Electrotechnics I | Microprogrammed Embedded Systems | Computational perception using multimodal sensors | Advanced topics in electromagnetic compatibility | Experimer

Figure: **Graph visualization - detail**

Conclusion

Thank you for your attention!

Visualization is available online[2].
The entire repository for the project is available on GitHub[3].

[2] https://robertinjac.github.io/EPFL-Courses-Similarity/

[3] https://github.com/RobertInjac/EPFL-Courses-Similarity