Swiss Federal Institute of Technology

Master Semester Project

Graph-Based Representation of EPFL Course Catalog

Author:
Maxime Coriou

Professor:
Patrick Jermann

Spring 2017

Contents

1 Introduction

The goal of this project was to find a way to represent EPFL course catalog. The representation should allow people to search for keywords and highlight the courses containing those keywords but also to see the required and recommended prerequisites of each course. This tool should allow people to look for their interests and choose the course that will help them to go further in those domains. Thanks to EPFL online services regarding courses, we have access to the full course catalog with a complete description of each course.

There was three main parts in this project, first the data extraction, then the construction of the database and finally the visualisation. We will develop those three parts in the following report.

2 Data Extraction

In the first part of this project we had to retrieve all the information regarding courses at EPFL. To do so we first needed to retrieve a list of all the courses and then get all the necessary information for each one of them.

2.1 Course Code Retrieval

The first step was to retrieve a list of all the courses given at EPFL in 2016/2017. Fortunately, IS-Academia implements an API that returns JSON objects containing, among other information, the courses' codes of all the course of a given semester. In our case, as we want all the courses of 2016/2017, we had to use the two following links :

- https://isa.epfl.ch/services/courses/2016-2017/HIVER
- https://isa.epfl.ch/services/courses/2016-2017/ETE

Those links respectively give the courses' codes of all the courses given in winter 2016 and all the courses given in summer 2017 for a total of 1552 courses.

2.2 Course Information Retrieval

Once we had all the courses' codes we were able to retrieve further information with another API from IS-Academia : https://isa.epfl.ch/services/books/2016-2017/course/+ courseCode (eg. https://isa.epfl.ch/services/books/2016-2017/course/COM-402).
For each course, this link returns the following information in a JSON object :

- Code : Give the code of the course
- Title : Give the name of the course

- Language : Give the teaching language of the course (French or English)
- Summary : Give a summary of the course's content
- Content : Give the detail content of the course
- Keyword : Give the keywords related to the course
- Required prerequisites : Give the mandatory prerequisites for this course
- Recommended prerequisites : Give the recommended prerequisites for this course

At the beginning we kept all those information in a CSV file while we were looking for an efficient way to store them in a database.

3 Construction of the Database

After some research, we decided that the best way to store the courses' information was to use a graph database. The main advantage of a graph database is that instead of representing data as rows and tables, it uses a graph representation (nodes and edges) of the data. In our case, we can easily represent a course and its prerequisites using a graph, each node is a course and we create a directed edge between a course and each one of its prerequisites. We have two kinds of edges in the database, the one for required prerequisites (REQUIRE_OBL) and the one for recommended prerequisites (REQUIRE_IND).

3.1 Parse the Prerequisites

We already had the prerequisites for each course (see 2.2) but unfortunately the prerequisites were not in a convenient format (it usually was a text containing course's name, codes or keywords). In order to create the database, the best way was to create a list of courses' codes from this text, where each code of the list represents a prerequisite. We used a simple way to do that, we look if the text contains courses' names or courses' codes and from this we construct the list of courses' code. Once this was done we created a CSV file with two columns (Source and Target) that we fill by creating an entry for each prerequisite, putting the course as the Source and the prerequisite as the Target.
We created two CSV files, one for the recommended prerequisites and one for the required ones.

3.2 Neo4j Database

The graph database we chose for this project is Neo4j[1]. Fortunately, Neo4j implements a function to create nodes and edges from a CSV file. The first step was to create a node for each course using the global CSV file containing all the information of each course.

In Neo4j, each node can have attributes that you can set at the creation of the nodes. In our case, each course has the following attributes :

- Code : Give the code of the course
- Title : Give the name of the course
- Summary : Give a summary of the course's content
- Content : Give the detail content of the course
- Keyword : Give the keywords related to the course

Once the nodes were created, we created the edges between the nodes using the CSV files containing the source and target of each edge. You can see a visual representation of the database in figure 1.

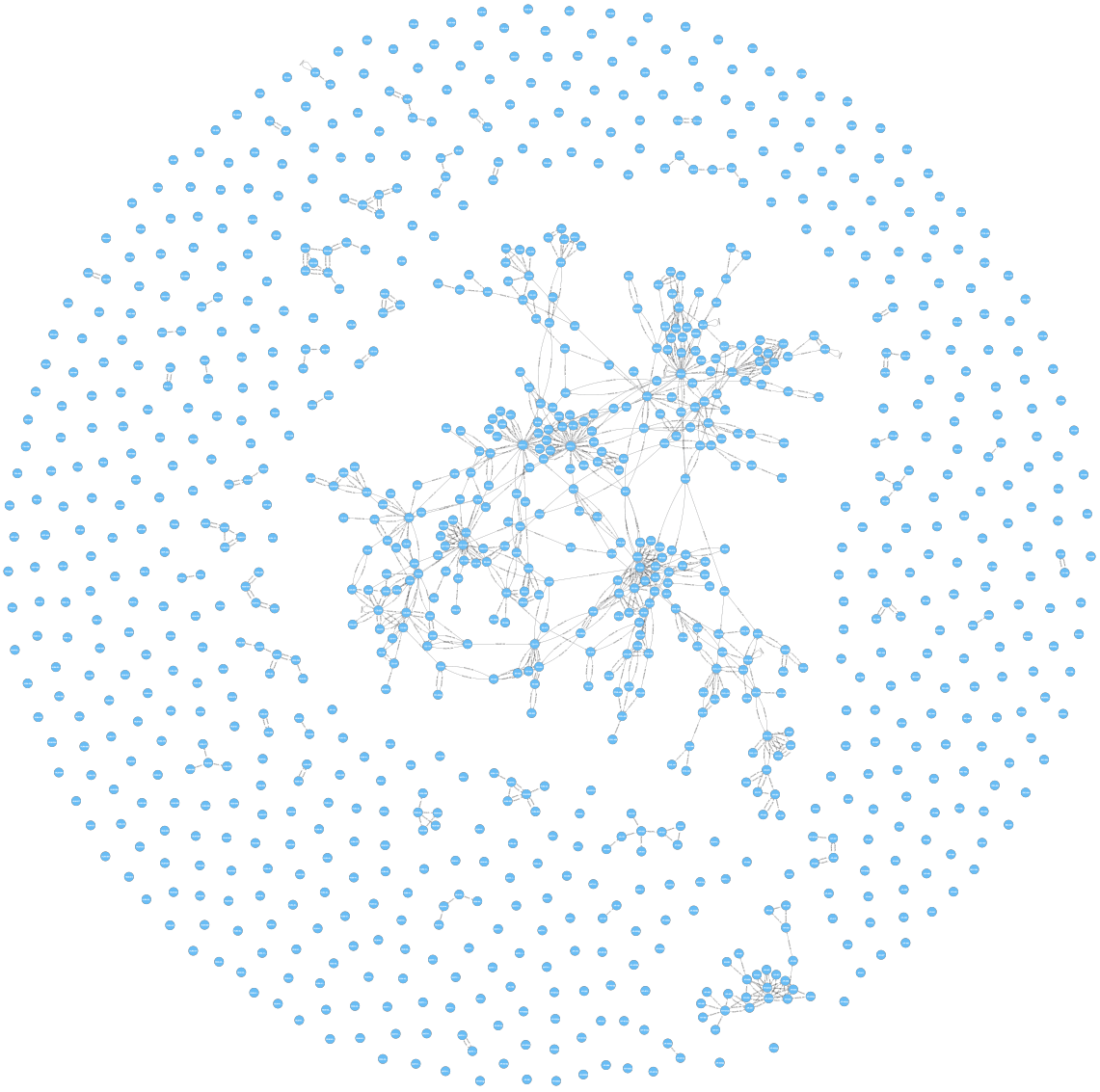

Figure 1: Neo4j database

4 Visualisation and Search Tool

4.1 Visualisation

In order to visualise and interact with the database we created a javascript representation of the database. To do so we used Sigma.js[2], a javascript library that includes a plugin to instantiate a graph from a Cypher request (Cypher is the language used to send request to a Neo4j database). We loaded the whole graph in a Sigma.js instance to work only with javascript and avoid doing many queries to the database.

4.2 Search Tool

Once we had the Sigma.js graph, we were able to implement a search tool that can look for keywords in the attributes of each node. For that we created a search box in which you can type some content, the algorithm is then looking, into each attribute of each node, for what you typed in (case insensitive).
You can see the global visualisation and a example of a search respectively in figure 2 and 3.

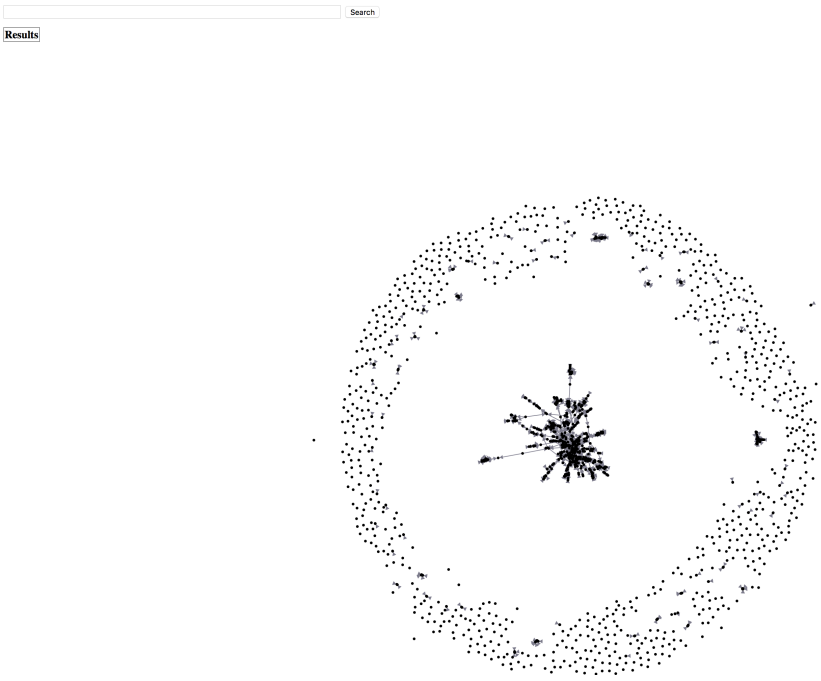

Figure 2: Full graph using Sigma.js

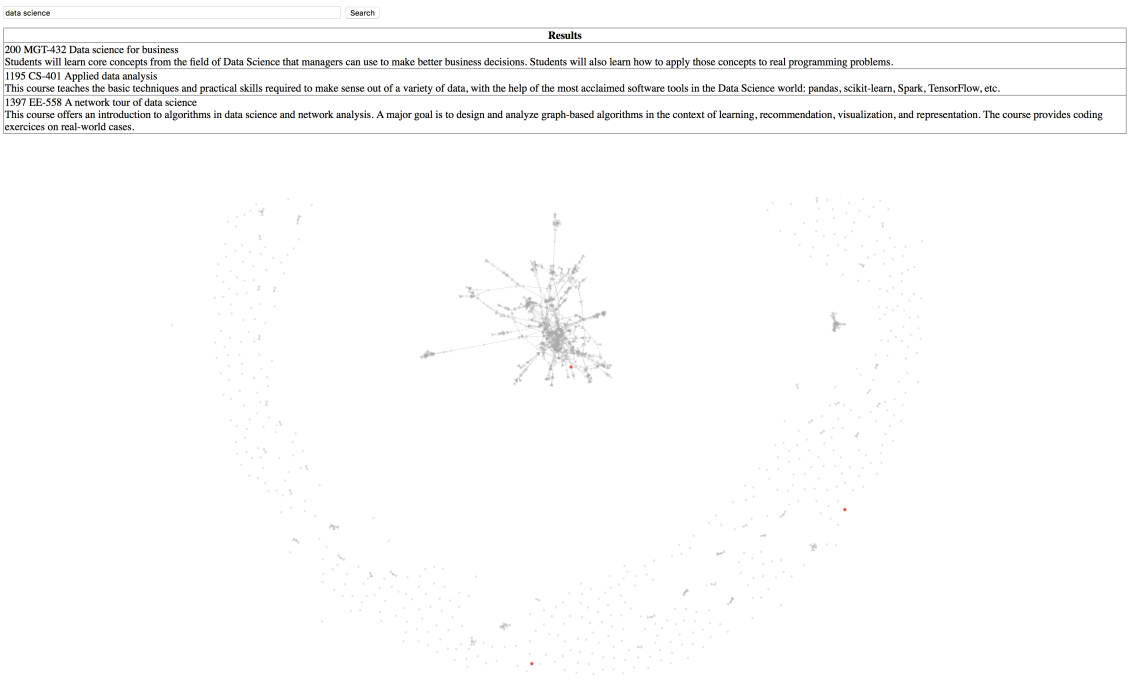

Figure 3: Results of the search "data science"

5 Conclusion

It was a great experience to work on a project in which we had to go through the whole process of data analysis. We had to extract the data, analyse and treat them, create a database, to finally realise a visualisation tool with a great potential to help futur students in the choice of their courses. It was also really interesting to work on a concrete project at EPFL on which it will be a pleasure to work again.

References

[1] Neo4j : https://neo4j.com/

[2] Sigma.js : http://sigmajs.org/