final_report

Industrial Immersion at Accounts Chamber: final report¶

Skoltech, 2019

Investigation of divergence between employment market and educational programs funded by federal budget¶

Student: Sergei Kozlukov

Company supervisor: Chistoborodov Alexander, Mikhail Petrov

Problem statement¶

In order to better prioritize and manage educational programs funded from federal budget, it is needed (among other things) to collect data on students graduating with various specializations in each region and data on open job positions at local companies. This data is implicitly available in open sources like aggregated govermental reports, social networks, employment-related web-services. Most of this data however is not linked nor is it machine readable.

Datasources¶

- Universities, their websites, offered programs, etc: **Obrnadzor**
- Graduated students in each programs, aggregated by region: **Minobrnauki** (VPO-1)
- Graduated students, per university: possessed by minobrnauki, currently unavailable because of lack of collaboration between governmental institutions
- Offered jobs with textural descriptions: **hh.ru**
- Offered jobs, linked to OKPDTR: **trudvsem.ru**
- OKPDTR linked to OKZ: **http://base.garant.ru/1548770/**

- OKZ linked to professional standards: **http://fgosvo.ru/docs/101/69/2**
- Professional standards linked to educational programs: **http://fgosvo.ru/fgosvo/142/141/16**

Parsing Obrnadzor data¶

Plain XML. See possible_sources.ipynb

Parsing Minobrnauki data¶

Minobrnauki reports data in standardized VPO-1 form, providing unstructured excel tables split into multiple sheets and apparently filled by hand. A coroutine-based non-deterministic finite state machine has been implemented to parse those reports. Resulting data contains graduates per program per region over the last year, although parser extracts much more data which could be used later.

Reports are parsed without any loss of data in generator_based_vpo1.ipynb and fed into MongoDB collection. Notebook graduates_regionwise.ipynb constructs a MongoDB aggregation pipeline which extracts from these parsed reports, filters and aggregates the data about graduates. This data is saved into graduates.csv

In [1]:

```
%run common.ipynb

pd.read_csv('graduates.csv').tail()
```

Out[1]:

	region	funded_by	time_involvement	program
21645	Удмуртская Республика	Частные	заочная	Менеджмент
21646	Удмуртская Республика	Частные	заочная	Государственное и муниципаль
21647	Удмуртская Республика	Частные	заочная	Торговое дело
21648	Удмуртская Республика	Частные	заочная	Строительство
21649	Удмуртская Республика	Частные	заочная	Техносферная безопасность

Educational programs (FGOS VO standards)¶

A bunch of loosely structured PDF-files listed at http://fgosvo.ru/fgosvo/142/141/16, some of them image-based. In okpdtr.ipynb I'm constructing a list of these documents and downloading them. The actual parsing is done in extract_fgosvo.py script (it's more comfortable to run such long tasks from terminal than in jupyter, plus I had problems with accessing ghostscript from firejailed conda environment). By the way, the script implements kind of reactive style (although it almost completely ignores error handling) which to my thinking is funny in the context of python. I'm using camelot-py to extract tabular data from PDFs in the form of pandas dataframe. It is not very efficient and also very unstable approach, which relies on access to ghostscript executable and a very high ulimit set, however it's Just Works (TM). Dataframes are filtered and links to profstandards are extracted:

In [2]:

```
pd.read_csv('program_to_profstandard.csv').tail()
```

Out[2]:

	program	ps
360	29.03.05	21.002
361	29.03.05	33.016
362	29.03.05	40.011
363	29.03.05	40.059
364	35.03.06	13.001

Professional standards to OKZ¶

Professional standards are defined by documents in http://fgosvo.ru/docs/101/69/2. This is again a huge pile of unstructured PDF files *most* of which are image-based. PDFs are downloaded in same notebook: okpdtr.ipynb. Then extract_okz.py, a script similar to previous one, filters out invalid and image-based PDFs and extracts links from professional standards to OKZ.

In [3]:

```
pd.read_csv('ps_to_okz.csv').tail()
```

Out[3]:

	ps	okz
330	5.005	2351
331	5.005	2359
332	5.005	3320
333	5.005	3330
334	5.005	3431

OKZ to OKPDTR¶

Finally, in the same notebook the link between OKZ and OKPDTR is extracted from some arbitrary webpage.

In [4]:

```
pd.read_csv('okz_to_okpdtr.csv').tail()
```

Out[4]:

	okpdtr	okz	name
7988	471103	3119	Техник службы пути
7989	471122	3119	Техник службы эксплуатации
7990	471226	3114	Техник-электрик-наладчик электронного оборудов...
7991	473378	2111	Физик (контролирующий) критического стенда
7992	478552	3113	Электромеханик устройств сигнализации, централ...

Merging data¶

In merging_tables.ipynb these CSVs are fed into a sqlite file and then simple join is used to link educational programs to OKPDTRs

In [5]:

```
import sqlite3


con = sqlite3.connect('programs.db')
```

In [8]:

```
pd.read_sql_query('select * from program_okpdtr limit 5;', con)
```

Out[8]:

	program	okpdtr
0	01.03.01	204395
1	01.03.01	204427
2	01.03.01	254784
3	01.03.01	254816
4	01.03.01	254841

In [9]:

```
pd.read_sql_query('select * from links limit 5;', con)
```

Out[9]:

	program	okpdtr	profstandard	okz
0	01.03.01	204395	01.001	2320
1	01.03.01	204427	01.001	2320
2	01.03.01	254784	01.001	2320
3	01.03.01	254816	01.001	2320
4	01.03.01	254841	01.001	2320

Trudvsem.ru¶

Trudvsem provides a huge XML file with open listings. These listings have textual descriptions and OKPDTRs. In trudvsem.ipynb I'm using established links to map OKPDTR lists in job listings to educational programs. I produce quantitative aggregated data about jobs similar to available data on educational programs and produce comparison histograms.

This data however does not allow to judge about divergence between programs and job market as for many educational programs the links to OKPDTRs are lacking (because of invalid or image based documents published by govermental institutions) and also trudvsem is unpopular service provided by government, thus it cannot represent the actual employment market's demand.

Future work¶

Data from trudvsem together with okz-profstandard-educational program hierarchy can be used to build and train a fuzzy model to map textual descriptions to educational programs. Specifically, available data forms a metric *graph* which we can try to isometrically-as-possible embed into a hyperbolic space, constructing embeddings for job listings from their textual descriptions via neural network. We could hope then that this model will be able to restore the larger graph, by using it to embed (in just forward mode) listings from, say, hh.ru into that same hyperbolic space.

In [ ]: