Python programming — projects

Finn Årup Nielsen
DTU Compute
Technical University of Denmark

September 1, 2014

1 Project for the Python programming

1. To pass the course you need to make
   (a) a Python program project and
   (b) make a report based on it as well as
   (c) a poster presentation.
2. Project groups should preferably be two persons, maximum 3.
3. The programming project should utilize some of the Python libraries concerning data mining (including text and image mining).
4. Choose one project on the following page or (perhaps better) suggest one yourself.
5. The project should **not** overlap with a project you are making in another course.

2 Project: Expert mining

Create a tool that will allow a user to identify researchers that are experts within an area and within a specific country. The user might specify a set of tags (keywords for a search, e.g., “music processing”, “signal processing”, “audio”, etc.) and a country (e.g., China) and expect to get lists of persons, groups and universities within the country ordered according to expertise.

Google Scholar may be a tool that can provide help. It does not have an API but it is possible to search and get HTML back with citation information that can be extracted with BeautifulSoup or similar. Another tool is *Microsoft Academic Search* that records researchers and organizations and also has citation information available. Other tools that might be of help are Wikipedia, Wikidata and DBpedia that however only records notable researchers and not often research groups. But it do indeed often have good overview of universities in countries. It may be possible to make a semantic queries for universities in a specified country in DBpedia and get back names.

Entities can, e.g., be characterized by number of citations or h-index (Hirsch, 2005) that tell how many papers a researchers have written that receive more citations than the number of papers. A high h-index may signify an expert, — but the h-index does not tell which area the expert work. Perhaps topic mining of papers, e.g., just their titles, may help to determine which area the person is expert in.

The goal is to have a system that will allow a researcher to get to know which other researchers and groups are relevant to contact in connection with research collaborations.

Please note that excessive downloads from Microsoft and Google may ban you. Commercial services such as ProxyMesh may get around this problem.

3 Project: Online news comment mining

This project will data mine Danish news comments. A particular interesting site is Dagbladet Information that has recommendations for both articles and comments and the readers are able to see the user name for the recommendations, thus a network of recommendations and comments can be established.

Research questions that may be posed:

1. Is it possible to see communities among the users? Do they clustered depending on political attitude, gender, ...?
2. Is it possible to get a value of how "entrenched"/"clustered" a discussion is? What topics generate such clustered discussions?
3. Why do some articles have many comments and other fews? Do sentiment has anything to say in this? Can we identify words/topics that generate comments?
4. Is it possible to predict whether a person will comment on or recommend a piece of information?
5. Is it possible to predict the gender of the commentor, e.g., a machine learning based approach.
6. Overall network characteristics: Do we have a power-law like law, e.g., for the number of comments?

The project may include social network analysis (of the network between people commenting), sentiment analysis and/or topic mining.

When you crawl the website be very sure not to overload the servers behind information.dk by pausing some seconds (e.g., 10 seconds) between downloads and better yet perform the download during nighttime. Also set the User-agent of your crawler, so information.dk IT-guys know it is you.

Issues:

- An article may be original Information material or a "telegram". telegrams are under the URL prefix: http://www.information.dk/telegram/
- An article with many comments has multiple pages.
- A user may be deleted, see http://www.information.dk/telegram/304318

4 Project: Facebook wall mining

Itziar Castello, a Spanish researcher from the Copenhagen Business Scholar, has together with two other researcher manually labeled posting and comments on a wall to Plastic Pollution Coalition with their own sweat of the brow. The comments and labels are available in a spreadsheet format. There are several manual annotations, e.g., for sentiment and conflict. Hundred posts are labeled by the 3 researchers so interrater analysis is possible.

Your task in this project is to analyze this data, e.g.:

- Train a machine learning classifier (e.g., with the NLTK naïve Bayes classifier) that can predict the (labeled) sentiment from the words in the postings.
- Train a machine learning classifier that can predict the other labels from the words in the postings.
- Compare word list labeled sentiment scoring (using, e.g., AFINN list) with the machine learning based sentiment scoring.
- Perform topic mining on the posts and comments.
- Interrater analysis: How much do the 3 researchers agree on the labeling?

5 Project: Online mechanical turk

Several companies, SamaSource, CrowdFlower and Amazon.com with MTurk, provide online services where internet users can work on small microproblems, e.g., score text for sentiment.

The project here is to build a similar system that collect internet users response to some task, e.g., classification of posts from Twitter or Identi.ca. Issues to consider:

- Configurable setup or fixed task?
- Should the Internet user provide a single answer/label for each item (e.g., its text) or multiple?
- Which kind of scoring.
- Handling of users such login.
- Statistical summary of collected data, e.g., if multiple users classify the same item how much do they agree.
- Collection and extraction of the raw data. Databasing. Export format.

There is a Python platform for crowdsourcing called PyBossa. It is used on http://crowdcrafting.org/. This might be relevant to dig into.

6 Project: Topic mining of research papers about Wikipedia

In connection with a systematic review a Semantic MediaWiki site has been setup that records scientific articles reporting research on Wikipedia: http://wikilit.referata.com. This wiki records much structured information about each article: abstract, authors, year of publication and and manual labeling, such as topic and domain.

Your task is to download, topic mine the abstracts and compare it with the manual labeling.

The topic mining can be with non-negative matrix factorization (NMF), or with methods from the `gensim` package or other way Topics from the automated topic mining can be compared to the annotation from manual labeled topics, and the topics can be visualized with respect to, e.g., domain or year of publication.

Extensions can include data mining of the co-author network.

7 Project: Topic mining in Twitter

In blogs, microblogs such as Twitter and other online forum people are usually taking about "something" and have an opinion about "something".

This project will involve crawling, downloading and extraction of texts from Twitter storing them (e.g.) in a database. You should the use text mining and language processing tools in Python for contextual analysis to detect the topic by multivariate analysis such as non-negative matrix factorization.

A system could be provides answers to "What topics are the people that follow RaceWithInsulin on twitter talking about", or "What other topics on twitter are mentioned in tweets about Libya".

Relevant books (Segaran, 2007; Bird et al., 2009; Makice, 2009).

8 Project: Link mining in Twitter

Status messages in Twitter often contain links to outside stories, such as news postings and blog entries. The links often these are shortened, e.g., with bit.ly.

This project will download tweets and find links, expand the shortened link and count in various ways to determine the impact of story on Twitter.

It could also, e.g., model the "impulse reponse" of a story.

Relevant books (Segaran, 2007; Makice, 2009).

Somewhat related scientific articles: "Modeling Information Diffusion in Implicit Networks"

9 Project: Sentiment analysis of company information on Wikipedia

Download pages from Wikipedia describing companies (either from the live version or the XML dump), e.g., Fortune 500. Analyze them for sentiment.

We have a word list, AFINN, that can be used for sentiment analysis (Nielsen, 2011). We have also a small company list with Wikipedia article match. Furthermore, DBpedia or Wikidata may be used to download relevant companies.

10 Project: Wikipedia first link analysis

Construct the first link network for several language versions of Wikipedia and compare them, e.g., with respect to PageRank across of articles.

You need to extract the first link and language links, e.g., with regular expressions and add the networks to NetworkX and/or NumPy. A PageRank method is available in NetworkX.

See also toolip of XKCD 903.

11 Project: YouTube sentiment analysis

Users can write comments on each YouTube video. Often these comments are outrageous and quite opinionated. With the YouTube API (gdata Python module) 1000 comments can be downloaded for each video.

The text mining technique "sentiment analysis" determines how positive or negative a text is. In the simplest form a labeled word list is used where each word is manually labeled for valence (positive/negative).

This project should construct a Python program that downloads comments from YouTube videos and perform sentiment analysis to rate each video, enabling queries like "What do people think about the specific viral video?"

Relevant books (Segaran, 2007). See also my blog post on Getting comments from YouTube via Python's gdata.youtube

12 Project: Twitter follower mining of politicians

Politicians are represented on Twitter and may gain followers.

This project should examine the network of politician followers and compared it to election results.

(Research of Juyong Park)

13 Project: Facebook mining

It is possible to search the public status messages on Facebook, e.g., with http://graph.facebook.com/search?q=IKEA

The data you will get are in a structured JSON format and slightly more complicated than the data that Twitter returns.

Instead of using Twitter the Facebook statuses can be used for topic mining. Note that lack of control of language and varying length of status may make it more difficult than Twitter mining.

14 Project: Twitter sentiment diffusion

Are tweets that are highly positive or highly negative distributed ("retweeted") more than neutral tweets? Which other variables determine what how much a tweet is retweeted, e.g., researchers have found that hashtag and followers correlated with the retweet rate (Suh et al., 2010), see also (Hansen et al., 2011).

A Python program should download tweets from Twitter, score them with regard to sentiment and find out which tweets are retweets.

15 Project: A web-service for brain activation

At DTU Compute we are running several so-called ’neuroinformatics’ Web services where neuroscientists can search for information, see
http://neuro.imm.dtu.dk/services/brededatabase/

This project will construct a Web service to handle 3D brain activations as well as their associated data, visualize the coordinates, and perform a statistical analysis of their distribution (Nielsen and Hansen, 2002).

This project will use the numerical and web parts of Python libraries.

16 Project: Characterization of wiki spamming

Wikis are usually open so anybody on the Internet can write on it. That means that anonymous users can vandalize pages on the wiki. Wiki administrators have the possibility to block anonymous edits from specific users. On MediaWiki web sites it is possible to see the blocked IP-number and the blocked user name.

This project will make a program for storing MediaWiki sites, that then can be queried to identify which IP-numbers are blocked and for what reason. The data could then be statistical analyzed to find possible patterns or be compared to services such as Project Honey Pot or Spamhaus or for geographical profiling.

See for one pointer to some MediaWikis: http://s23.org/wikistats/

17 Project: Comparing matlab and Python

In this project you need to know Matlab well, — as well as Python.

The idea is to compare the performance of Matlab and Python on a range of different tasks with respect to speed, memory performance and perhaps ease of implementation.

Choose some relevant numerical processing operations as well as string operations.

18 Project: DTU course mining

Download the courses of DTU from the course database.

Build an online system that show the relationship between courses, e.g., the network from prerequisites and perform text mining to show a list of similar courses.

Extensions of the projects can include handling of courses on other universities that DTU and/or course planning.

19 Project: Authorship or style classifier

Download texts from different authors or sources, e.g., Runeberg, Wikisource, NLTK corpora, Twitter, news . . . . Extract features and the train a classifier to distinguish between one or more classes: type (e.g., is it a news story), author gender (is it written by a male or female), author (e.g., H.C.Andersen), language, year.

20 Project: Named entity extraction

Make a system that can extract named entities from text. Named entities are, e.g., personal names, names of organizations, etc. The NLTK book (Bird et al., 2009) has an explanation of this problem, in section 7.5.

Named entity extraction usually requires a gazeteer: a dictionary with the interesting words and phrases to find. Such a system could, e.g., be based on Wikipedia information. Some named entity recognition systems will attempt to link the extracted entities to other databases, e.g., DBpedia.

NLTK’s named entity chunker may do a reasonable job:

```
import nltk
import re
from pprint import pprint

text = u"""The Technical University of Denmark (Danish: Danmarks
Tekniske Universitet), often simply referred to as DTU, is a
university in Kongens Lyngby, just north of Copenhagen, Denmark. It
was founded in 1829 at the initiative of Hans Christian Ørsted as
Denmark’s first polytechnic, and is today ranked among Europe’s
leading engineering institutions, and the best engineering university
in the Nordic countries."""
text = re.sub("\n", " ", text)

process = lambda sent: nltk.ne_chunk(nltk.pos_tag(nltk.word_tokenize(sent)))
def is_named_entity(tree, nodes=["ORGANIZATION", "GPE", "PERSON"]):
    return hasattr(tree, ’node’) and tree.node in nodes

sentences = nltk.sent_tokenize(text)
```

Result:

```
>>> pprint(map(lambda tree: filter(is_named_entity, list(tree.subtrees())),
               map(process, sentences)))
[[Tree(’ORGANIZATION’, [(u’Technical’, ’NNP’), (u’University’, ’NNP’)]),
  Tree(’GPE’, [(u’Denmark’, ’NNP’)]),
  Tree(’PERSON’, [(u’Danmarks’, ’NNP’), (u’Tekniske’, ’NNP’), (u’Universitet’, ’NNP’)
  Tree(’ORGANIZATION’, [(u’DTU’, ’NNP’)]),
  Tree(’GPE’, [(u’Kongens’, ’NNP’), (u’Lyngby’, ’NNP’)]),
  Tree(’GPE’, [(u’Copenhagen’, ’NNP’)]),
  Tree(’PERSON’, [(u’Denmark’, ’NNP’)])],
 [Tree(’PERSON’, [(u’Hans’, ’NNP’), (u’Christian’, ’NNP’)]),
  Tree(’PERSON’, [(u’Denmark’, ’NNP’)]),
  Tree(’GPE’, [(u’Europe’, ’NNP’)]),
  Tree(’GPE’, [(u’Nordic’, ’NNP’)])]]
```

Not quite right: Where is “of Denmark”? Is it possible to do better?

21 Project: Characterize links from and on the DTU web-site

Download web pages from DTU homepages and extract the links, so you will be able to answer how the different (departmental) servers are links, and which outside web servers are linked.

Use, e.g., `NetworkX` and a non-negative matrix factorization algorithm on the adjecency matrix.

See, e.g., for algorithms (Segaran, 2007).

22 Project: Online Python

Make a web service where users may upload code that is then executed and possible evaluated for result, speed and memory usage.

One application of such a system is for online programming course evaluation, where a teacher sets up an exercise and students submit (small) programs with a solution to the exercise.

One problem with execution of user code on a webserver is that the user might potential write "bad" code. The solution is to "sandbox" the code, e.g., so the code cannot open files on the web server. There are some efforts in PyPy for sandboxing of code.

See an example with online execution in PyPedia.

23 Project: Government data

The Danish government has recently released government data, e.g., information about Danish companies. Use this in an application.

- CVR download. Data is provided in comma-separated values files.
- Tinglysning. It is not clear how easy it is to download the information contained in the Tinglysning.
- Open Data Aarhus. Aarhus municipality has setup CKAN data publishing site with many different data set in various formats.
- Open Data København. Copenhagen Municipality data.
- . . . ?

Other country beyond Danmark also have sites with government data is published.

24 Project: CampusNet mining

See CampusNet public API

25 Project: Machine learning wikidata

Wikidata https://wikidata.org is a Wikipedia sister project with semi-structured data. The semi-structured data can be fetch in Python via, e.g., pywikidata.

Wikidata has items with claims, e.g., that Berlin is the capital of Germany.

An idea is to predict values of claims from other values of claims, e.g., predict whether the gender of a person based on other values.

26 Project: Kaggle

Kaggle is a machine learning prediction competition site. There are various tasks that one can compete in, see the competition list. For many of the competitions there are a monetary reward. One example (of an expired) MLSP 2013 Bird Classification Challenge.

27 Machine learning as a Service

Build a webservice where users can submit problems that are solved by the server either with pretraining machine learning classifiers or classifiers trained on user-provided data.

For inspiration see, e.g., uClassify and by topic-sentiment webservice.

28 Portable EEG

Emotiv manufactures a low cost portable EEG system. There is a Python module for capturing data from Emotiv. This project should build a system for EEG storage, analysis and visualization of the EEG signal from the Emotiv system. We have a limited set of Emotiv systems available for those which are interested in this project.

References

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly, Sebastopol, California. ISBN 9780596516499.

Hansen, L. K., Arvidsson, A., Nielsen, F. Å., Colleoni, E., and Etter, M. (2011). Good friends, bad news — affect and virality in Twitter. In Park, J. J., Yang, L. T., and Lee, C., editors, *Future Information Technology*, volume 185 of *Communications in Computer and Information Science*, pages 34–43, Berlin. Springer. Link.

Hirsch, J. E. (2005). An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences of the United States of America*, 102(46):16569–16572. DOI: 10.1073/pnas.0507655102.

Makice, K. (2009). *Twitter API: Up and running*. O'Reilly, Sebastopol, California.

Nielsen, F. Å. (2011). A new ANEW: evaluation of a word list for sentiment analysis in microblogs. In Rowe, M., Stankovic, M., Dadzie, A.-S., and Hardey, M., editors, *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages*, volume 718 of *CEUR Workshop Proceedings*, pages 93–98. Link.

Nielsen, F. Å. and Hansen, L. K. (2002). Modeling of activation data in the BrainMap$^{\mathrm{TM}}$ database: Detection of outliers. *Human Brain Mapping*, 15(3):146–156. DOI: 10.1002/hbm.10012. Link.

Segaran, T. (2007). *Programming Collective Intelligence*. O'Reilly, Sebastopol, California.

Suh, B., Hong, L., Pirolli, P., and Chi, E. H. (2010). Want to be retweeted? large scale analytics on factors impacting retweet in Twitter network. In *2010 IEEE International Conference on Social Computing (SocialCom10)*. IEEE.