Text Mining for Social Media Analytics - Course projects

January 27, 2016

1 Project descriptions

1.1 Topic classification

- Underlying dataset: 20 newsgroups (http://qwone.com/ jason/20Newsgroups/; standard dataset used for classification tasks)
- Task: correctly predict the **topic** of a new input text (assumption: new input text belongs to one of the 20 topics already in the dataset)

Hints

- Simplifications for prototyping:
  - Build own mini-corpus (from VectorSource) with 2 classes, 3-5 examples for each class
  - Pick two subclasses from 20newsgroups (instead of all 20 classes)
  - Common algorithms: SVM, Naive Bayes, logistic regression
- Online tutorials to check: http://www.svm-tutorial.com/2014/11/svm-classify-text-r/, https://github.com/chenmiao/Big_Data_Analytics_Web_Text/wiki/Machine-Learning--Text-Mining-with-R
- Packages to check: RTextTools, e1071

1.2 Sentiment analysis

- Underlying dataset: ca. 9000 product reviews from Amazon (positive/negative/neutral)
- Task: correctly predict the **sentiment** (positive/negative/neutral) of a new input text

Hints

- Simplifications for prototyping:
  - Build own mini-corpus (from VectorSource) with 2 classes, 3-5 examples for each class
  - Drop the neutral class to get cleaner positive/negative distinction
- Common algorithms: SVM, Naive Bayes, logistic regression
- Consider using n-gram features
- Online tutorials: http://chengjun.github.io/en/2014/04/sentiment-analysis-with-machine-learning-in-R/

1.3 Concept extraction

- Dataset: ca. 2000 job descriptions extracted from CVs
- Task: extract concepts from classes that are relevant for recruiting:
  - Positions (e. g. Java Developer)
  - Companies (e.g. Siemens)
  - Skills (e. g. programming)

Hints

- Research on Named Entity Recognition / Entity Extraction
- Build up small lists/dictionaries for the concept classes.
- Frequent and relevant terms can be found with tf-idf.
- Packages to check: openNLP

1.4 Web scraping

- Dataset: small seedset of websites
- Task: extract texts and their relevant metadata (title, date, author...) and store them (preferably in JSON format)

Hints

- Use BeautifulSoup package (http://www.crummy.com/software/BeautifulSoup/bs4/doc/)
- Main relevant features in html code: links (<a>), paragraphs (<p>), divs ( ), headings (<h1>, <h2> etc.), classes (<h1 class='title'>)
- Tutorial: http://www.analyticsvidhya.com/blog/2015/10/beginner-guide-web-scraping-beautiful-soup-python/
- Starter code (example on BusinessInsider website):

```
from bs4 import BeautifulSoup
import requests

url = "http://uk.businessinsider.com/?r=US&IR=T"
start_page = requests.get(url)
soup = BeautifulSoup(start_page.text)

for title in soup.findAll("a", {"class": "title"}):
    print(title.text)
```

2 Planning and submission

- Project work during class: Jan 27th (10-12.30h) and 28th (8.20-10h)
- Intermediate status presentation: Jan 28th 10.30-12.30
- Autonomous finalization of project until February 20th (please contact me via email for any issues)
- Submission:

- Code (with comments on different parts, steps etc.)
- 2-3 page description:
  - application functionality
  - the challenges met during the development process
  - desiderata for future work