Technical University of Denmark

Data mining of Information.dk using Python

Eleftherios Manousakis - s141714
Sigurd Knarhøi Johannsen - s042910

02819 Data Mining using Python
10/12/2014

Description

We describe a lightweight script that performs online article mining with sentiment analyzing, using standard components of Python.
It first downloads the articles found on the main page of information.dk, then stores them in a local database, calculates their sentiment and plots them in a specific manner in a few hundred milliseconds.

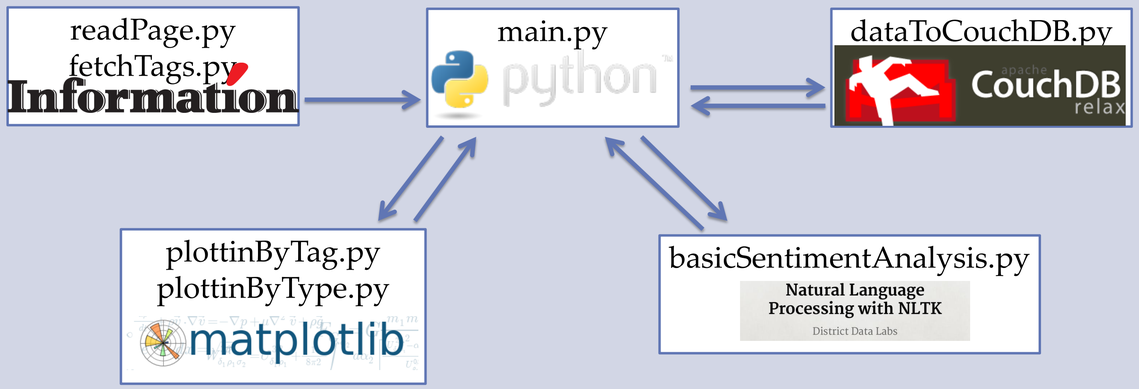
Data mining

www.information.dk has different kinds of articles of which we choose three to compare. Using sentiment analysis we plot articles against their comments, compare article types and examine article subjects, using tags provided by www.information.dk. We hope to find correlation in article subjects and differences in sentiment among article types.

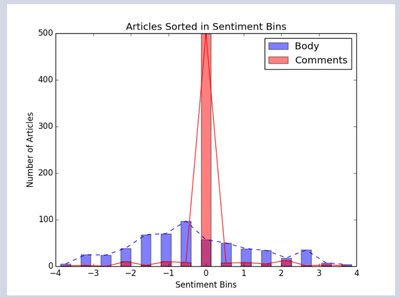
Technologies and Libraries used

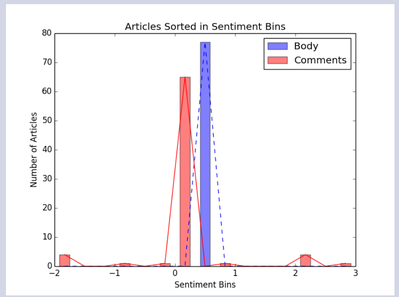
Python 2.7	
couchdb	
urllib	
bs4	Beautiful Soup
nltk	word_tokenize
matplotlib	pyplot
collections	Counter, defaultdict
flake8	
Sphinx	

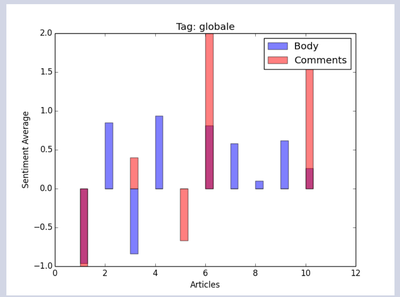
Architecture

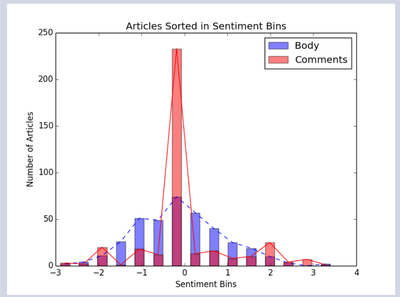


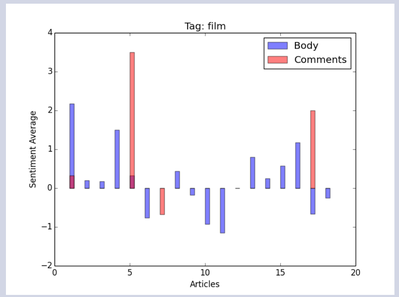
Data representation

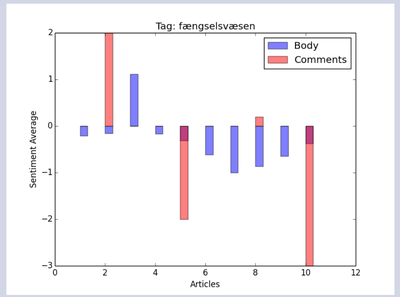












Data processing

Every day we parse the main page and have found that www.information.dk divides its news in the following categories: 'artikler', 'databloggen', 'føljeton', 'fotobloggen', 'kortfilmsbloggen', 'nyhedsblog', 'protokol' and 'telegram'.

All articles consist of a body, comments and tags. Usually between one and four tags accompany an article. Each tag consists of one or two words or a name. By sorting articles by tags, we are somewhat able to compare articles on the same subject, allowing us to see if sentiments follow the subject.
Telegrams have a body and comments but no tags.
News blogs consists of a headline, a link to another page and comments, but provides neither body nor tags.

How do we compare data?
By individually comparing the article body and its comments with a sentiment list, we get an average sentiment for each. We plot this correlation and compare the graphs for each of the three article types.
Afterwards we search our database by tags. The three most common tags are used - and articles with the same subject, according to tag, are compared.