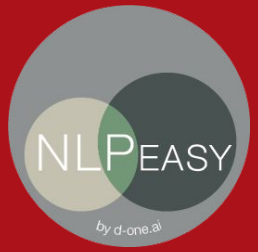
WE MAKE SENSE.

DATA DRIVEN VALUE CREATION

DATA SCIENCE & ANALYTICS | DATA MANAGEMENT | VISUALIZATION & DATA EXPERIENCE

D ONE, Sihlfeldstrasse 58, 8003 Zürich, d-one.ai



NLPeasy

a Workflow to Analyse, Enrich, and Explore Textual Data

Dr. Philipp Thomann
D ONE | NLP Expert Group 24. July 2020

NLP - Challenge for "normal" Data Scientists?

NLP is harder than "standard" data science

- higher dimensional
- specialised pre-processing needed
- NLP experts often assume the data is mainly text and maybe some "metadata"

What stands in the way of just trying exploration?

- Methods and models have reputation of being hard to use 

⇒ easy plugins for spacy, ...
- Data Scientist's toolbox is cumbersome for textual data:
  - ggplot, seaborn: How do you visualise text there?
  - Power BI, Tableau: How do you explore textual data in dashboarding tools?
  - {My-/Postgre} SQL {-ite, Server}: Does my DB system have good functions for textual data?

⇒ search interface in Elasticsearch/Kibana

Quick Demo

Connect to Elastic and Kibana or start in Docker (optionally)

Read / clean data in pandas, here title and abstract of NIPS papers
⇒ message, title, author, year, ...

Start Pipeline

Regex to extract LaTeX-Math
⇒ Tag-col: math

Calculate Sentiment of message
⇒ Num-col: sentiment

NLP-methods based on SpaCy
⇒ Tag-cols: message_entity, message_subj, title_subj, ...

```
[1]: import pandas as pd
     import nlpeasy as ne

[3]: elk = ne.connect_elastic(dockerPrefix='nlp', dockerElkVersion='7.1.1', dockerMountPoint=None)

     'No elasticsearch on localhost:9200 found, trying connect to docker container with prefix nlp'
     'No docker container with prefix nlp; starting one'
     ElasticSearch on http://localhost:32774
     Kibana on http://localhost:32775

     Get some data we already prepared

[4]: nips = pd.read_pickle('./nips.pickle')
     nips.shape

[4]: (8250, 20)

     Setup the pipeline

[5]: pipeline = ne.Pipeline(index='nips', elk=elk,
                            textCols=['message','title'], dateCol='year')
     pipeline += ne.RegexTag(r'\$([^$]+)\$', ['message'], 'math')
     pipeline += ne.VaderSentiment('message', 'sentiment')
     pipeline += ne.SpacyEnrichment(cols=['message','title'])

     Let's to the magic

[6]: nips_enriched = pipeline.process(nips, writeElastic=True)

     8250/8250 [==============================] - 4:53 36ms/step
```

Write the results to Elastic

Setup of ElasticSearch
Type of column is mapped

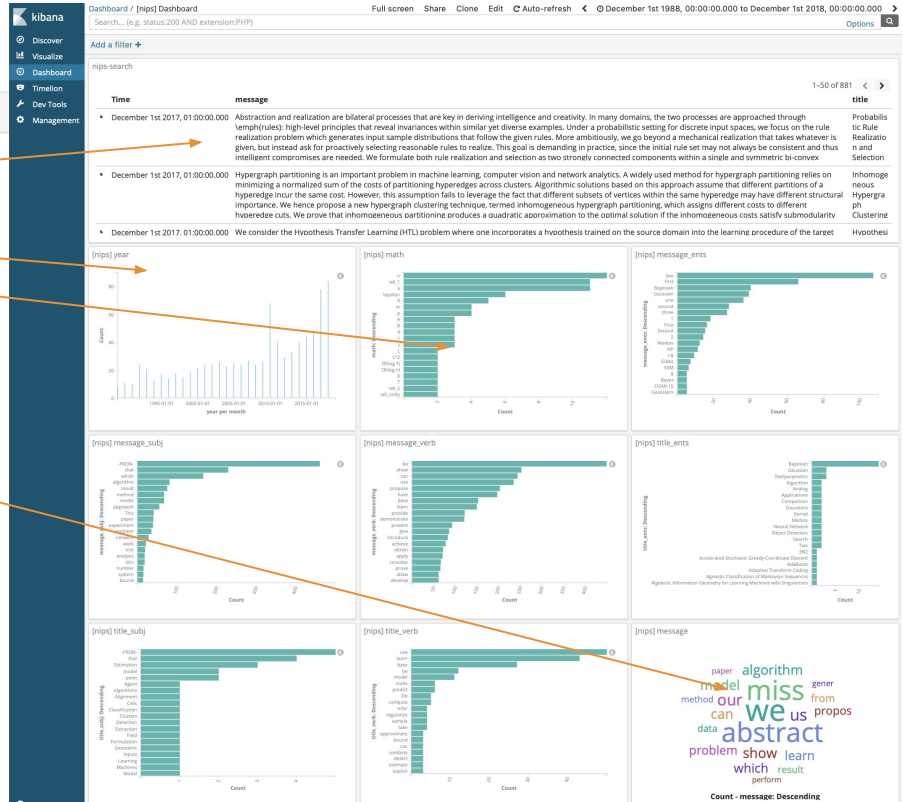
Run the pipeline in batches of 100 records

Automatic Dashboard Generation

Based on the column types different visuals are created, all integrated into a dashboard:

```
[7]: pipeline.create_kibana_dashboard()

nips: adding index-pattern
nips: setting default index-pattern
nips: adding search
nips: adding visualisation for year
nips: adding visualisation for math
nips: adding visualisation for message_ents
nips: adding visualisation for message_subj
nips: adding visualisation for message_verb
nips: adding visualisation for title_ents
nips: adding visualisation for title_subj
nips: adding visualisation for title_verb
nips: adding visualisation for message
nips: adding visualisation for title
nips: adding visualisation for sentiment
nips: adding dashboard
nips: setting time defaults
```

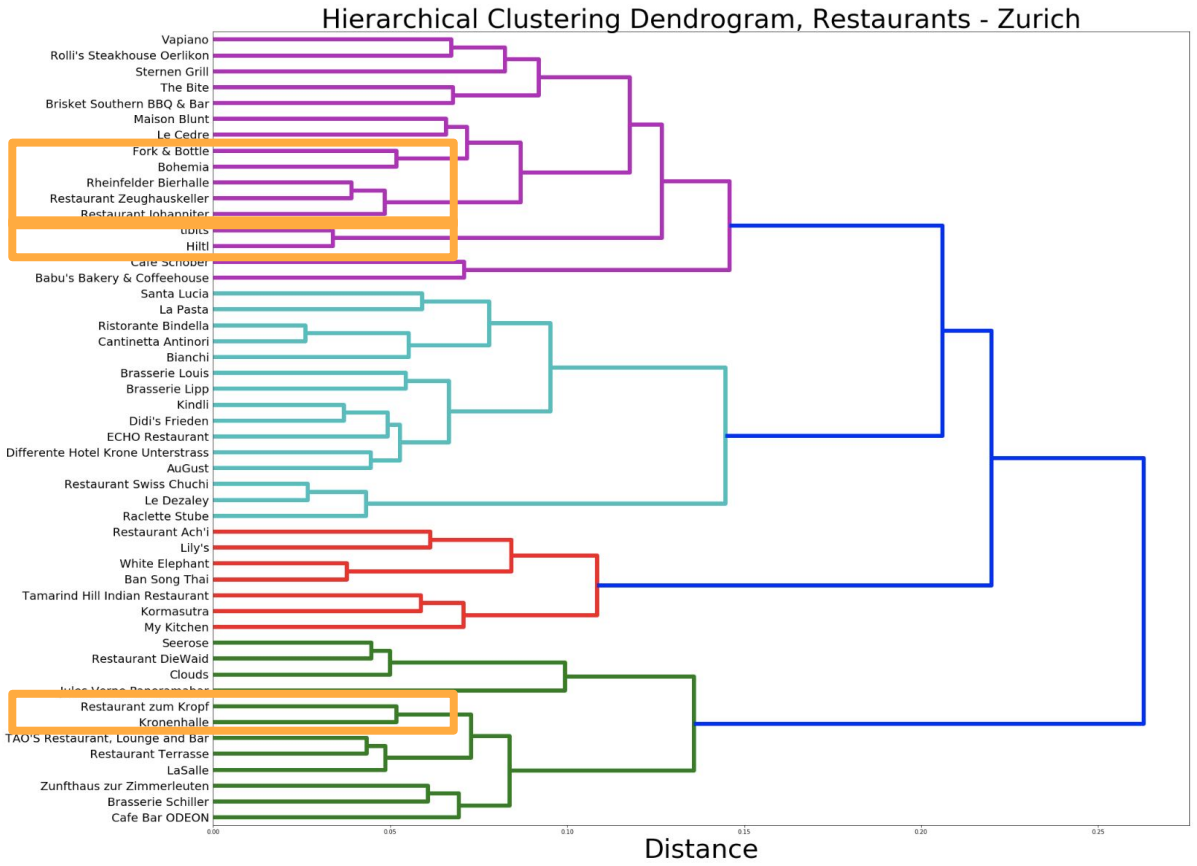


The generated visualisations can easily be changed in the Kibana UI.

Soon: Also auto-visualisations for Networks and GeoLocation (as in examples)

Restaurant similarity

- Based only on similarity of reviews
- Clusters detect review similarity for
  - vegetarian places
  - beer halls
  - ethnic food
  - decor



Average sentiment score of restaurant reviews

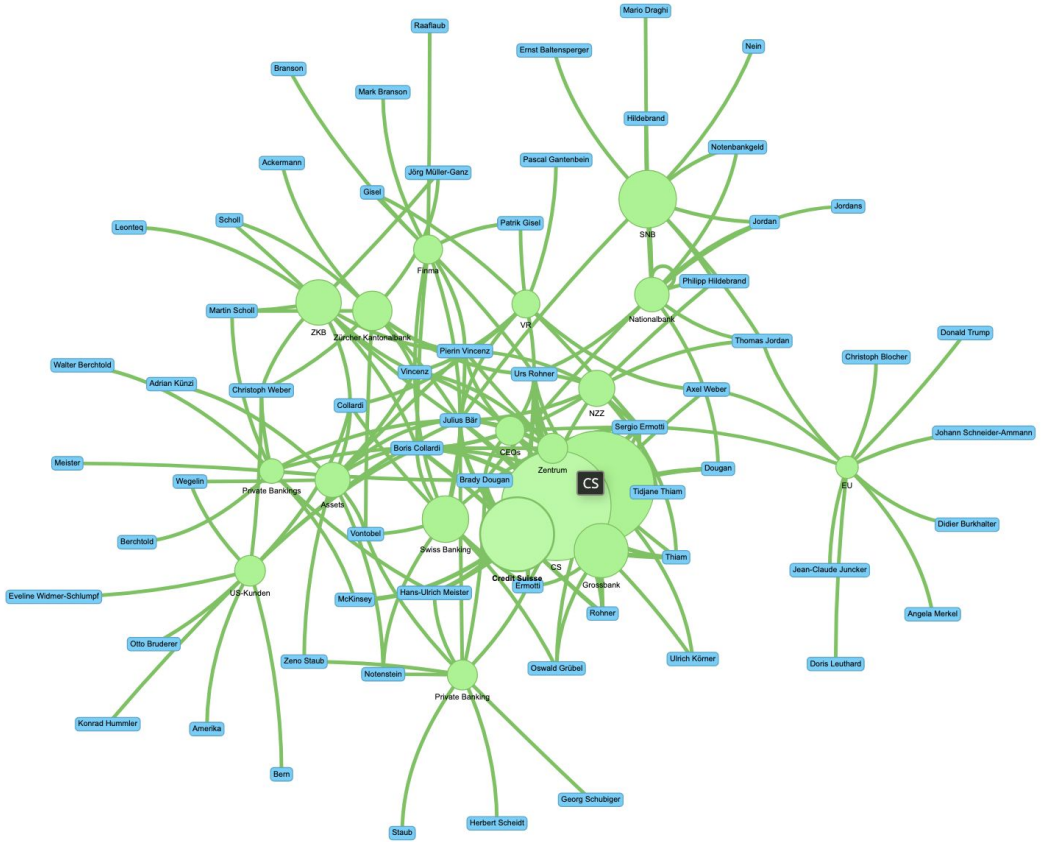
Unfortunately for me I was not aware of this until a bit later when I started being very ill. Fortunately for me my allergy does not result in anaphylactic shock so I did not end up in hospital but it has resulted in my spending the entire night vomitting and having very painful stomach cramps.

I would never, ever go near this restaurant even if you paid me. In my opinion a restaurant that so clearly risks customers' health by ignoring their stated allergies, and produces such mediocre food and rude service, does not warrant any customers at all.

Date of visit: September 2013

Value	Service
Atmosphere	Food

Insideparadeplatz.ch - what is it and what does it stand for?

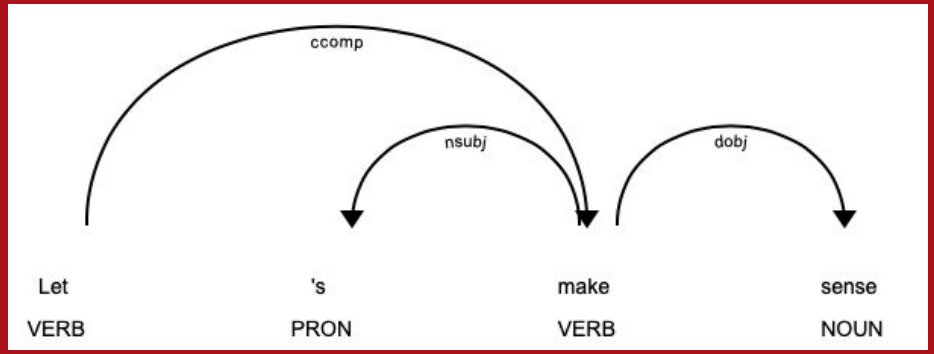


Thanks

- NLPeasy is OpenSource
  - https://github.com/d-one/NLPeasy
  - PRs welcome ;-)
  - https://mybinder.org/v2/gh/d-one/NLPeasy/master?urlpath=lab 

  - Tutorial: https://github.com/d-one/NLPeasy-workshop 

- Package still in development! Likely upcoming features
  - Adding more Stage-Plugins (BERT, Cleaning, …)
  - Support for incremental working (e.g. train on vecs and upload them to ElasticSearch)
  - Support for other DBs (Solr, SQLite, Postgres), dashboard generation into dash?
  - MyBinder examples: N(eur)IPS, Enron, OKcupid, Git-Commits, tripadvisor(?), insideparadeplatz(?), ...
- If you're interested in NLP or other projects, contact me
  - at philipp.thomann@d-one.ai



LET'S MAKE SENSE

Philipp Thomann
philipp.thomann@d-one.ai

+41 44 435 10 24

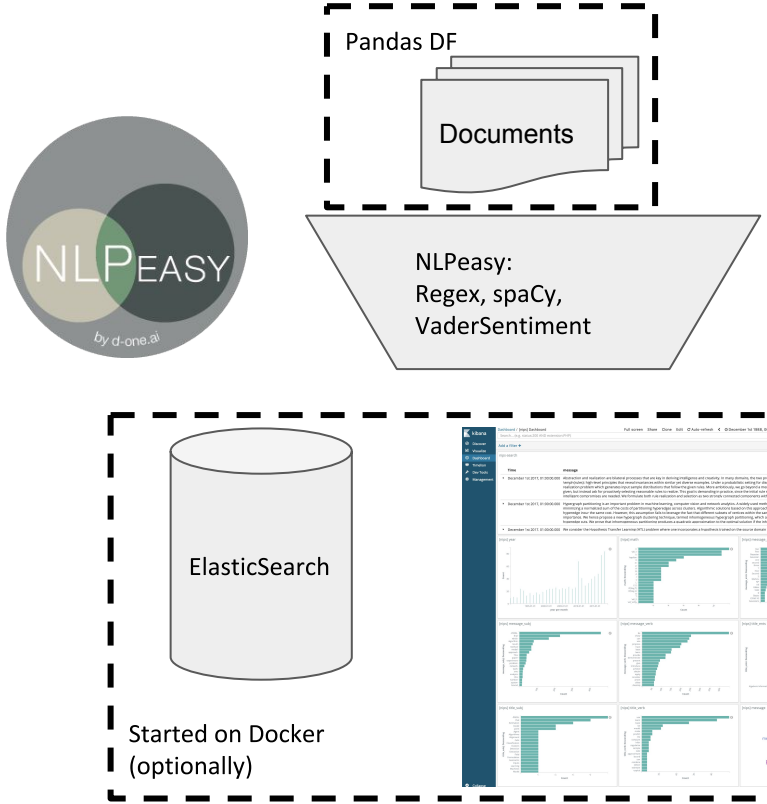
D ONE Solutions AG
Sihlfeldstrasse 58
CH-8003 Zürich

NLPeasy

Workflow that enables
- painless integration of many well-known NLP tools
- into a powerful workflow
- for quick exploration and more

Features:
- **Pandas** based pipeline that enables easy use of
  - **Regex**-based Tagging
  - **SpaCy**-based NLP-methods:
    Named Entity Recognition, Syntax Analysis
  - **Vader** SentimentAnalysis (en)
  - Support for Scraping using **BeautifulSoup**
  - … all you want to add
- Write results to **ElasticSearch**
  - Set good default config ("mappings")
  - Simple start of Elastic/Kibana servers in **Docker** if needed
  - Automatic Kibana Dashboard generation
- Apache License 2.0, https://github.com/d-one/NLPeasy

```
pip install nlpeasy
```



Why another pipelines package?

NLPeasy has a (simple) ontology of data-types:

- **tag**: list of str
- **text**: strings to be fed into NLP
- **numeric**, e.g. sentiment, counts
- **date** useful for Kibana
- **geo location**

NLPeasy stages add columns with the correct type and the automatic dashboard generation has decent defaults per data-type.