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 | EuroPython 2020, 24. July 2020

About me

- Vita
  - PhD in Probability Theory
  - PostDoc in ML
  - Managing Consultant with D ONE Solutions
- Projects in Data Science, ML, AI, Infrastructure, Visualisation, Coaching
- (Co-)Creator of
  - liquidSVM - A fast and versatile SVM implementation
  - Nabu - vocabulary drilling tool
  - NLPeasy - Easy Peasy Language Squeezy
  - PlotVR - walk through your data

About

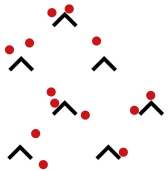
- Zurich-based, since 2005 focus: data-driven value creation, >1’000 data projects
- 50+ data professionals with excellent business and technical understanding & network
- International & Swiss clients - more than half of SMI-listed companies among our clients
- Selectively invested in start-ups - winji TRUE POWER

Digitale Transformation

Mit datenbasierter Intelligenz zur Energiewende

Business Consulting
- Guide on data journey
- Use cases, roadmap
- Transformation and change
- Building capabilities

Machine Learning / AI
- Ideation, implementation, operationalization
- Algorithms, modelling, NLP
- Image recognition, visual analytics
- Text analytics



Passionate & Down to earth

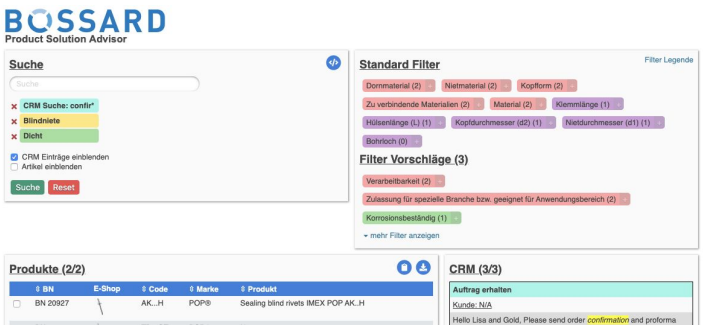
Data Architecture
- Data strategy
- Enterprise requirements
- Information factory
- BI and analytics
- Data science laboratory

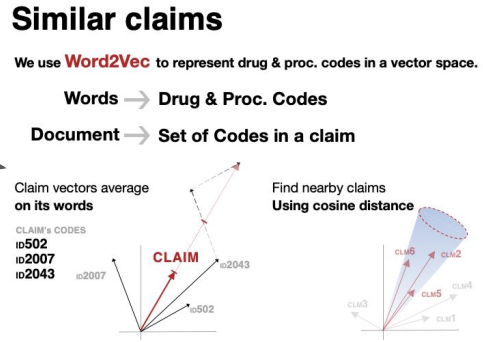
Data Management
- Data supply chain
- Data integration and modelling
- Structured & unstructured data
- Automated development framework

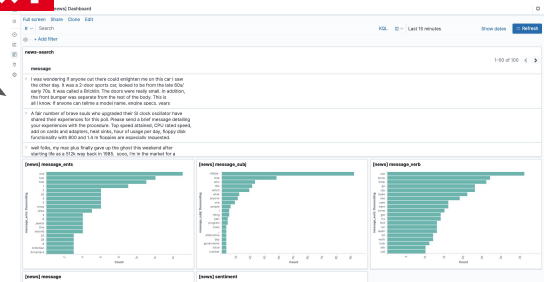
Data Experience
- From data to Insights to action
- Business report design
- Compelling data stories
- Communication & visualization

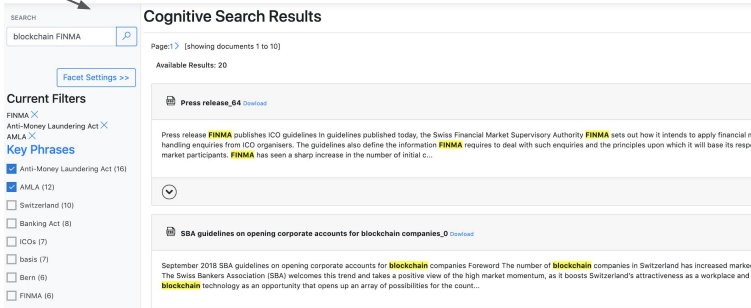
NLP Projects @ D ONE

- Product Solution Advisor: Elastic+Neo4j
- Health Insurance Claim Processing: Word2Vec on non-textual data
- Hawk Eye: Azure Cogn. Serv.
- Customer Feedback Analysis: spaCy Syntax
- KYC process support: Bing → NLP → Elastic
- NLPeasy: Open Source Python Package for NLP processing









Introduction

About Natural Language Processing (NLP)

- Big progress in last years
  - Word2Vec
  - Deep Learning: (Bi-)LSTM, CNN, RNN, …

    ⇒ many good pre-trained Models
- Abundant Data
  - CRM entries, mails, documents, customer reviews, ...
  - Many Use-cases: classification, sentiment, named entity recognition (NER), …

    ⇒ Next big thing?

Challenges for Data Scientists?

NLP is harder than "standard" machine learning

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

Why not try out exploratively for a use case?

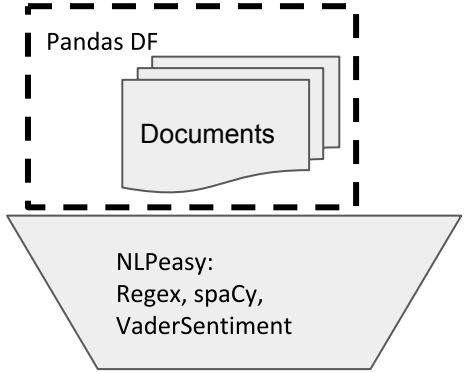
- Methods and models have reputation of being hard to use
- Standard tools cumbersome for textual data:
  - ggplot, seaborn: How do you visualise text there?
  - Power BI, Tableau: How do you explore textual data in dashboarding tools?
  - (My-/Postgre) SQL (-ite, Server): Does my Database systems have good functions for text?

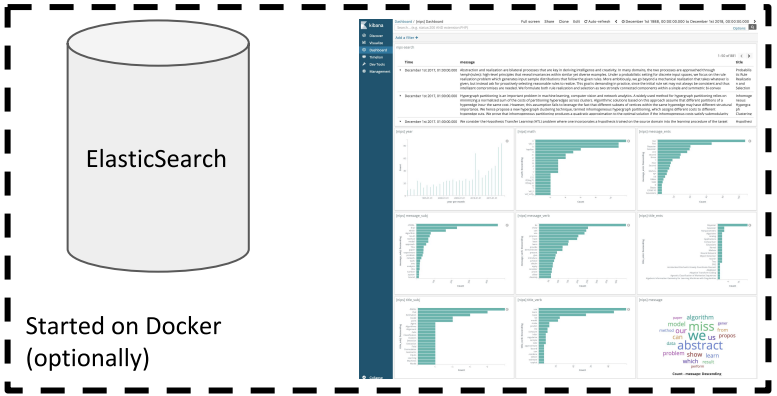
NLPeasy can help you overcome these obstacles

NLPeasy

Workflow that enables painful integration of many well-known NLP tools into a quick but powerful workflow:

- **Pandas** based pipeline enabling:
  - **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**
  - Add good default config (mappings)
  - Support of iterative workflow (todo)
- Gives a quick Bootstrap and then allows for an **agile** workflow to use the power of the tools to get more insights
- Simple start of Elastic/Kibana servers in **Docker** if needed.
- Apache License 2.0, https://github.com/d-one/NLPeasy,
- `pip install nlpeasy`
- https://github.com/d-one/NLPeasy/blob/master/demo.ipynb





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

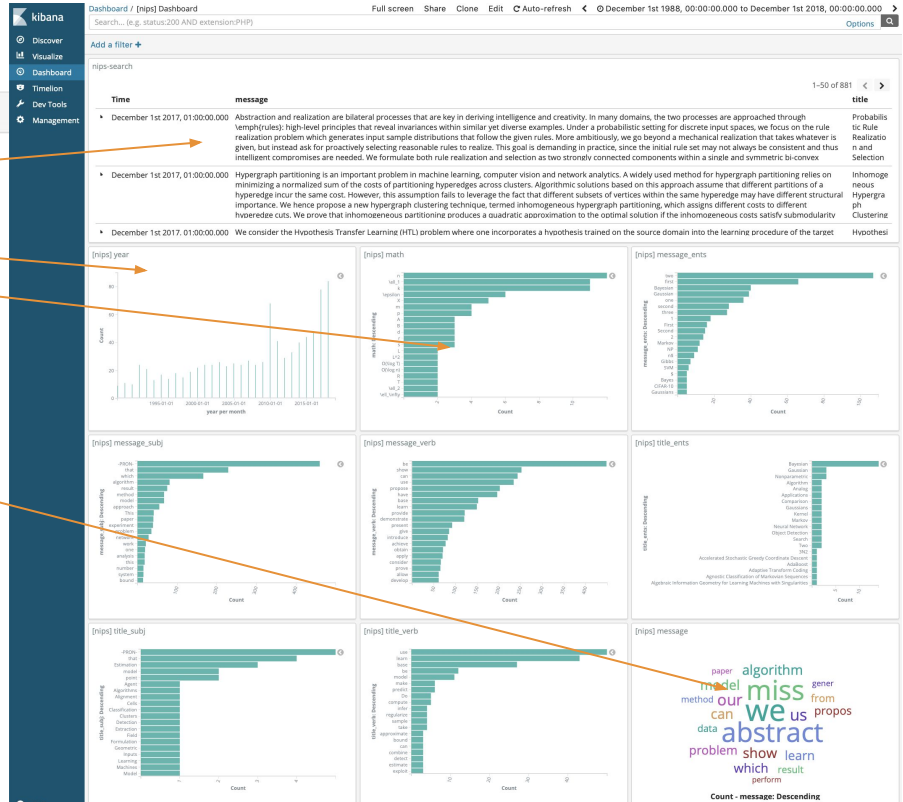
Run the pipeline in batches of 100 records

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 automatic Visualisations can be changed in the Kibana UI.

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

```
pipeline = ne.Pipeline(index='nips', elk=elk,
                       textCols=['message','title'], dateCol='year')
```

text: **message, title** | date: **year**

```
pipeline += ne.RegexTag(r'\$([^$]+)\$', ['message'], 'math')
```

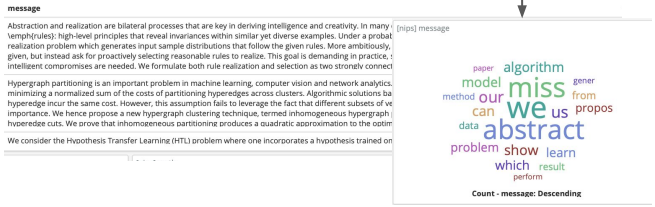
text: message, title | tag: **math** | date: year

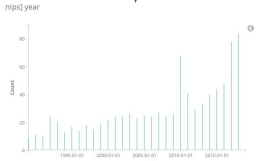
```
pipeline += ne.VaderSentiment('message', 'sentiment')
```

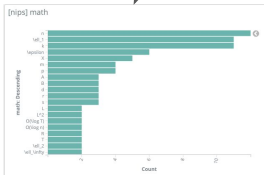
text: message, title | numeric: **sentiment** | tag: math | date: year

```
pipeline += ne.SpacyEnrichment(cols=['message','title'])
```

text: message, title | numeric: sentiment, **title_num_verb, ...** | tag: math, **title_ents, message_verbs, ...** | date: year



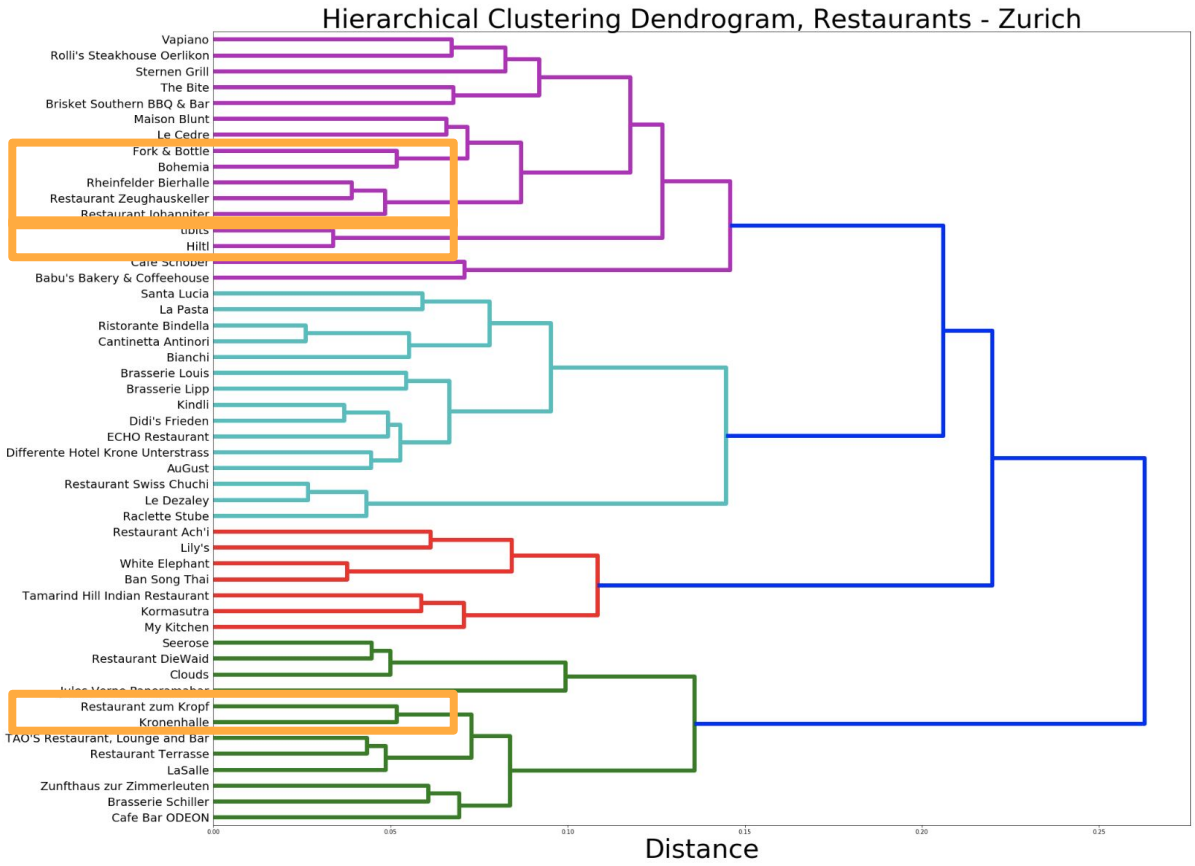




Demo

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

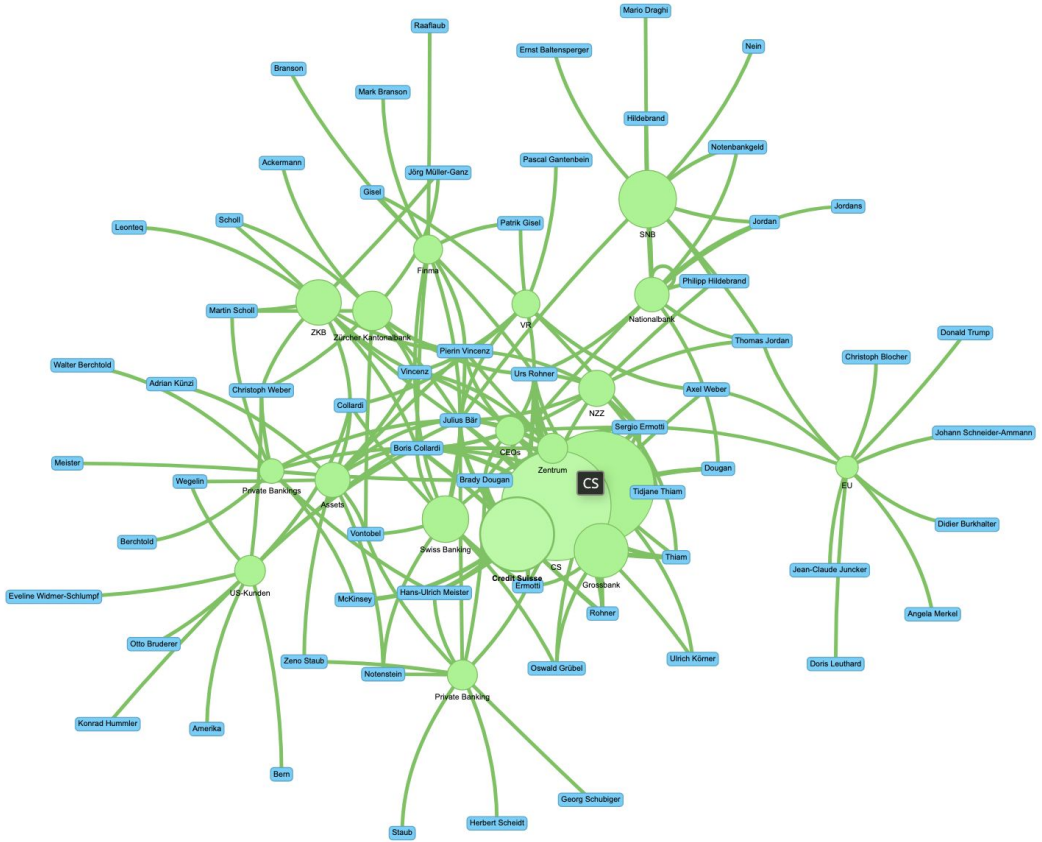
Unfortunately for me I was not aware of this until a bit later when I started being very ill. Fortunately for me my allergy does not result in anaphylactic shock so I did not end up in hospital but it has resulted in my spending the entire night vomitting and having very painful stomach cramps.

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?



We have different setups to try it out or work with it:

https://mybinder.org/v2/gh/d-one/NLPeasy/master?urlpath=lab

Mybinder-VM

Docker container: Jupyter
/user/d-one.../lab

/user/d-one.../kibana
/user/d-one.../proxy/5601

Kibana
Port 5601

/user/d-one.../proxy/9200

elasticsearch
Port 9200

Local + Docker

Jupyter
Port 8888

Docker cont: Kibana
Port (random)
started by NLPeasy

Docker cont: elasticsearch
Port (random)
started by NLPeasy

Local + Install ELK

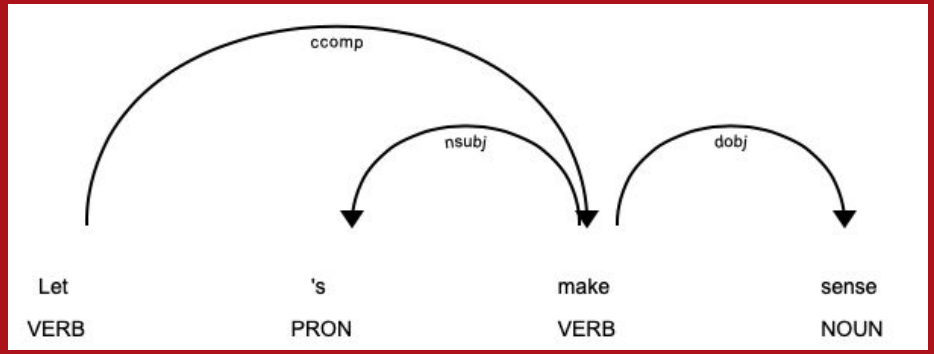
Jupyter
Port 8888

Kibana
Port 5601

elasticsearch
Port 9200

Thanks

- NLPeasy is OpenSource
  - https://github.com/d-one/NLPeasy
  - PRs welcome!
  - https://mybinder.org/v2/gh/d-one/NLPeasy/master?urlpath=lab
- 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)
  - More stable APIs, documentation
  - Support for integration of pipelines into ETL
- If you're interested in NLP or other projects, contact me
  - at philipp.thomann@d-one.ai
  - in the talk-nlpeasy discord channel now ;-)



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