WE MAKE SENSE.

DATA DRIVEN VALUE CREATION

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

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

Agenda

Time	Content	Who
9.00	Introduction	Philipp
9.30	Hands-On 1: How to kick-start NLP	Jacqueline
10.15	Break / Letting People Catch up	All ;-)
10.45	About ELK	Philipp
11.00	Hands-On 2: Deep Dive	Jacqueline
12.15	Wrap up / Discussion	All

Today’s hosts

Philipp Thomann

Here today & tomorrow morning

Jacqueline Stählin

Here today & Monday till Wednesday

Setup

Course Git

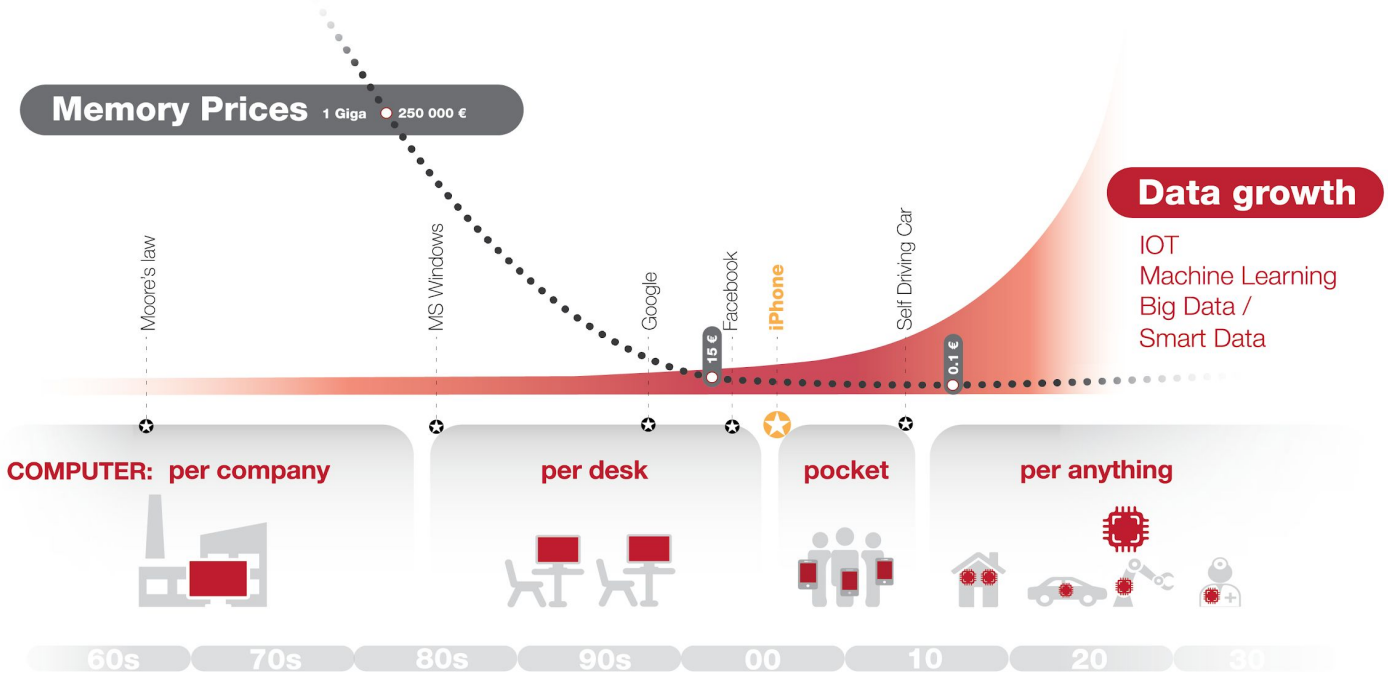
- https://github.com/d-one/NLPeasy-workshop

For infos during Workshop

- https://tlk.io/amld2020-nlpeasy

«Making sense of data
is the winning competence
in all industries.»

Data drives Transformation



About

- Zurich-based, established in 2005 with focus: data-driven value creation
- 50 data professionals with excellent business and technical understanding & network
- International & Swiss clients
- In selected cases: investor for data-driven start-ups


Mit datenbasierter Intelligenz zur Energiewende

winji TRUE POWER

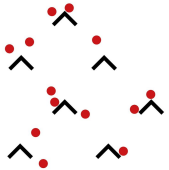
SKUNK SPORTS


Und dann verteilt er den Fussballern plötzlich Damen-BHs

Capabilities

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



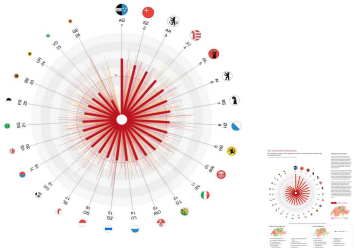
Passionate & Down to earth

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

Visualizations - the “last mile” of data



Award winning visualizations in national and international media and competitions

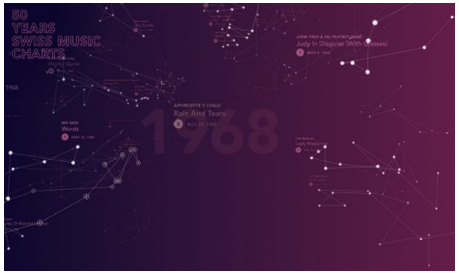
Swiss Music Charts: Playful Digitization
https://50-jahre-hitparade.ch/

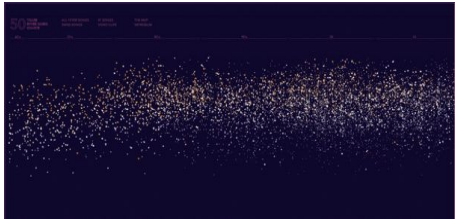
From the Data Story to the Saturday Night TV Show

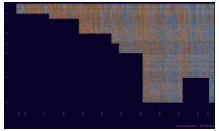
Highly creative handling of data
→ enriched with spotify data

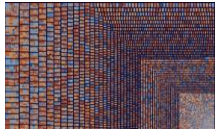
Data Stories development for SRF
https://50-jahre-hitparade.ch/analysis

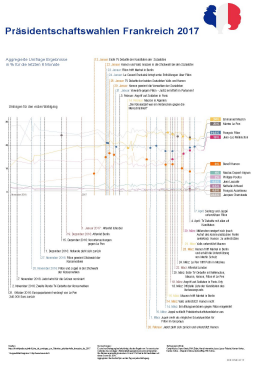
High-end implementation - broad range of tools (Web-GL, D3, Tableau)

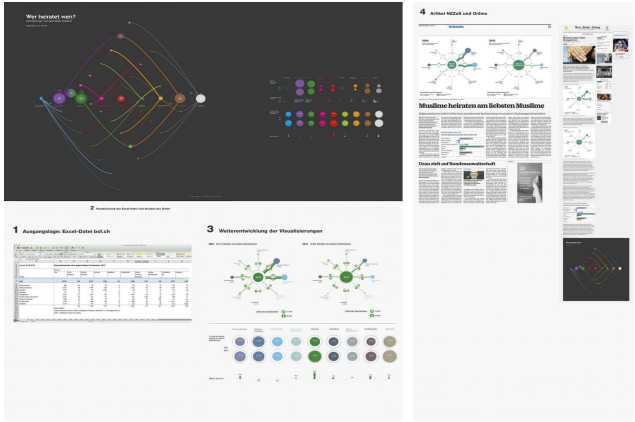












VIZ published:

NZZ am Sonntag
watson
BZ Berner Zeitung
Tages-Anzeiger
20 Minuten

ADC
best of web gold 2018
ED European Design Awards
AWWWARDS
THE WEBBY AWARDS
Museum für Gestaltung Zürich
FWA Site of the Day

Introduction

About Natural Language Processing (NLP)

- Big progress in last years
  - Word2Vec
  - Deep Learning
  - Pre-trained Models
- Abundant Use Cases
  - CRM entries, mails, documents, for classification, sentiment, named entity recognition (NER), translation, summarization, ...
  - Next big thing?
- Challenges for Data Scientists:
  - Models / tools have bad reputation?
  - Standard tools (e.g. plotting, SQL-DBs, Dashboars) cumbersome for exploration

NLP Primer

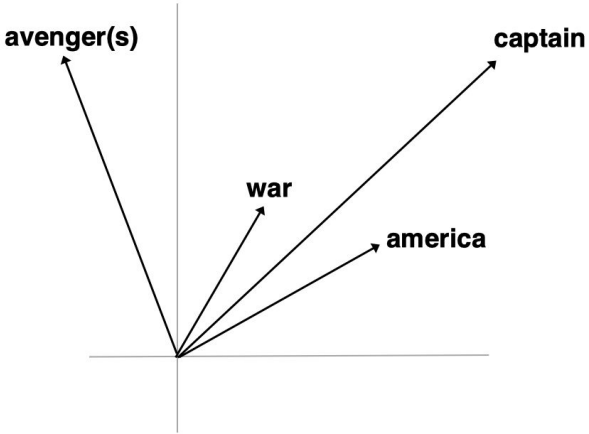
- Handling Texts
  - Tokenization, Stopwords
  - Scenarios: Classification, Topic Modeling, Sentiment Analysis
  - Make it amenable for ML
- Libraries (python)
  - classical: nltk, gensim
  - word2vec, FastText, Sent2Vec
  - "industrial-strength NLP": spaCy
  - Deep Learning: TensorFlow, BERT, …
- Libraries (R):
  - tidytext, reticulate, ...

NLP 101 - From Text to Vector

Traditional NLP:
Frequency of each word in Text

infinity	endgame	civil	first	war	captain	america	the	avenger(s)
							1	1
		1		1	1	1		
			1		1	1	1	1
1				1				1
	1							1

NLP 102 - Word2Vec:
Estimate distribution based on context



Vectors then can be used for:

- Topic Modelling: Consider top eigenvalues of matrix
  ⇒ Each eigenvector is a "topic"
- TF-IDF (Term-Frequency Inverse-Document-Frequency):
  For a word, divide its frequency in a given document by its frequency in general

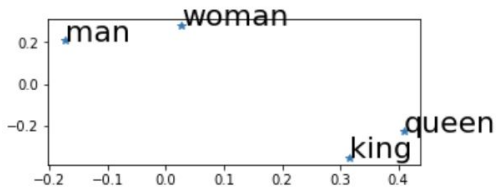
spaCy

```
[1]: import spacy
     import numpy as np
     import matplotlib.pyplot as plt
```

```
[2]: nlp = spacy.load('en_core_web_md')
```

```
[3]: words = 'man woman king queen'.split()
     v = { i: nlp.vocab[i].vector for i in words }
     matrix = np.stack(list(v.values()))
```

```
[4]: plt.figure(figsize = (5,2))
     ax = plt.plot(matrix[:,0], matrix[:,1], '*')
     for i, word in enumerate(words): plt.text(matrix[i,0], matrix[i,1], word, size=22)
```




```
[5]: k,r,s = nlp.vocab.vectors.most_similar( (v['king'] - v['man'] + v['woman']).reshape(1,-1), n=10)
     for K,R,S in zip(k[0],r[0],s[0]):
         print(nlp.vocab[K].text,R,S,)
```

```
King 2182 0.8024
KIng 3149 0.8024
Queen 5309 0.7881
QUEEN 6025 0.7881
COMMONER 11899 0.6401
Prince 7473 0.6401
Kings 6602 0.6209
SULTANS 9575 0.6209
Princess 8297 0.6126
PRINCESSES 9117 0.6126
```

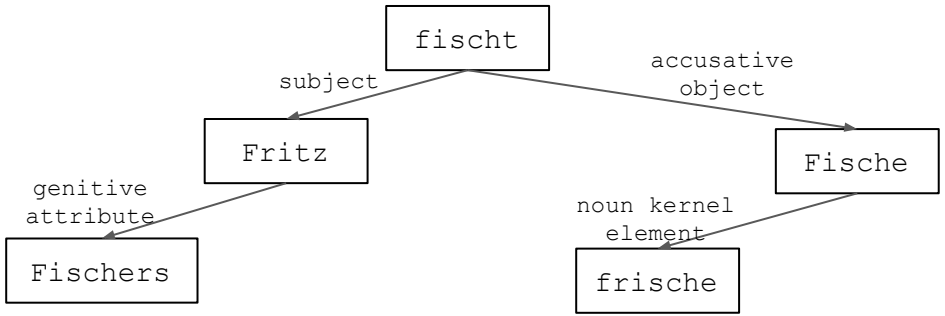
Installation:

```
pip install spacy
python -m spacy download de
python -m spacy download en
```

Features

- share a CNN based on embedding
- predict *super tag* for POS, morphology and dependency label
- trade a little accuracy for lot of speed
- implemented in cython

Linguistic Features

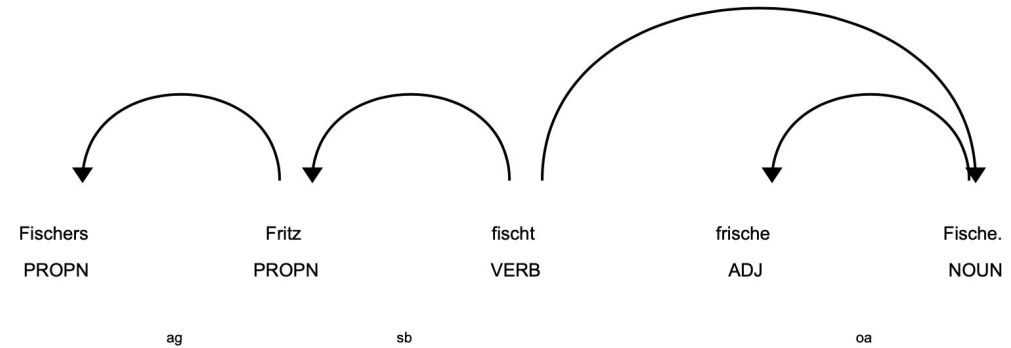


Word	Fischers	Fritz	fischt	frische	Fische
Lemma	*Fischer*	*Fritz*	*fischen*	*frisch*	*Fisch*
POS	NOUN	PROPN	VERB	ADJ	NOUN

spaCy

```python
import spacy
nlp = spacy.load('de')
doc = nlp(u'Fischers Fritz fischt frische Fische.')
for toc in doc:
    print( toc.text, toc.pos_, toc.dep_, spacy.explain(toc.dep_), toc.head )
spacy.displacy.render(doc, jupyter=True)
```

```
Fischers PROPN ag genitive attribute Fritz
Fritz PROPN sb subject fischt
fischt VERB ROOT None fischt
frische ADJ nk noun kernel element Fische
Fische NOUN oa accusative object fischt
. PUNCT punct punctuation fischt
```



Installation:

```
pip install spacy
python -m spacy download de
python -m spacy download en
```

Features

- share a CNN based on embedding
- predict *super tag* for POS, morphology and dependency label
- trade a little accuracy for lot of speed
- implemented in cython

Why NLPeasy

- Quick-start toolkit for general data scientists to start exploratory analysis of textual data
- Out-of-the-box features for the texts in your data:
  - Vader sentiment analysis
  - SpaCy
  - Word2Vec
- Explore your data in search engine (Elasticsearch) and a powerful dashboard framwork (Kibana)
- Quickly gain an overview with automatically built Kibana dashboard
- Take it from there and reveal insights!

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

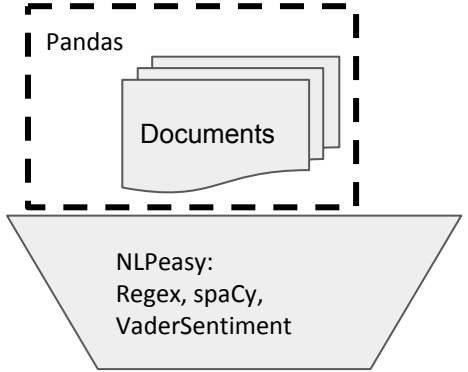
Setup of ElasticSearch
Type of column is mapped

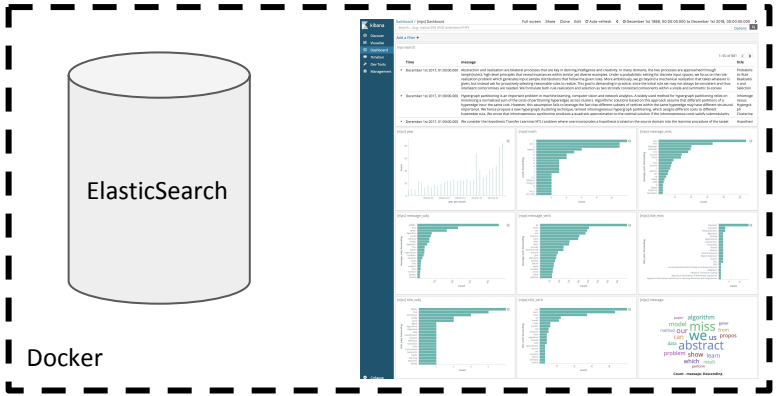
Run the pipeline in batches of 100 records

NLPeasy

Toolbox that enables quick and easy 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



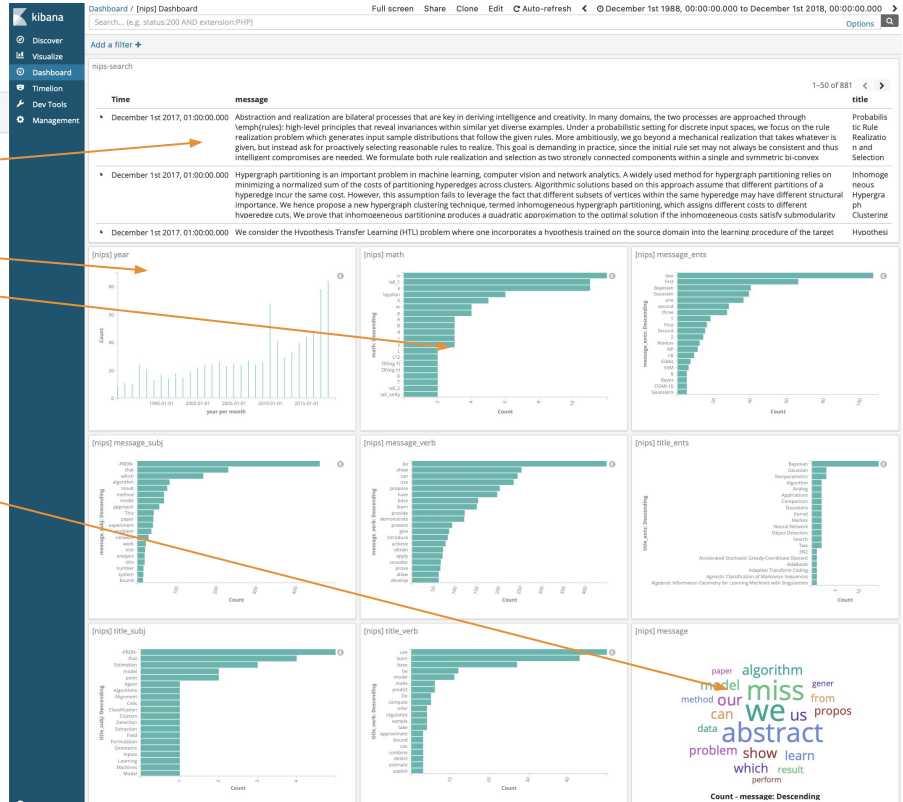


Automatic Dashboard Generation

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
```

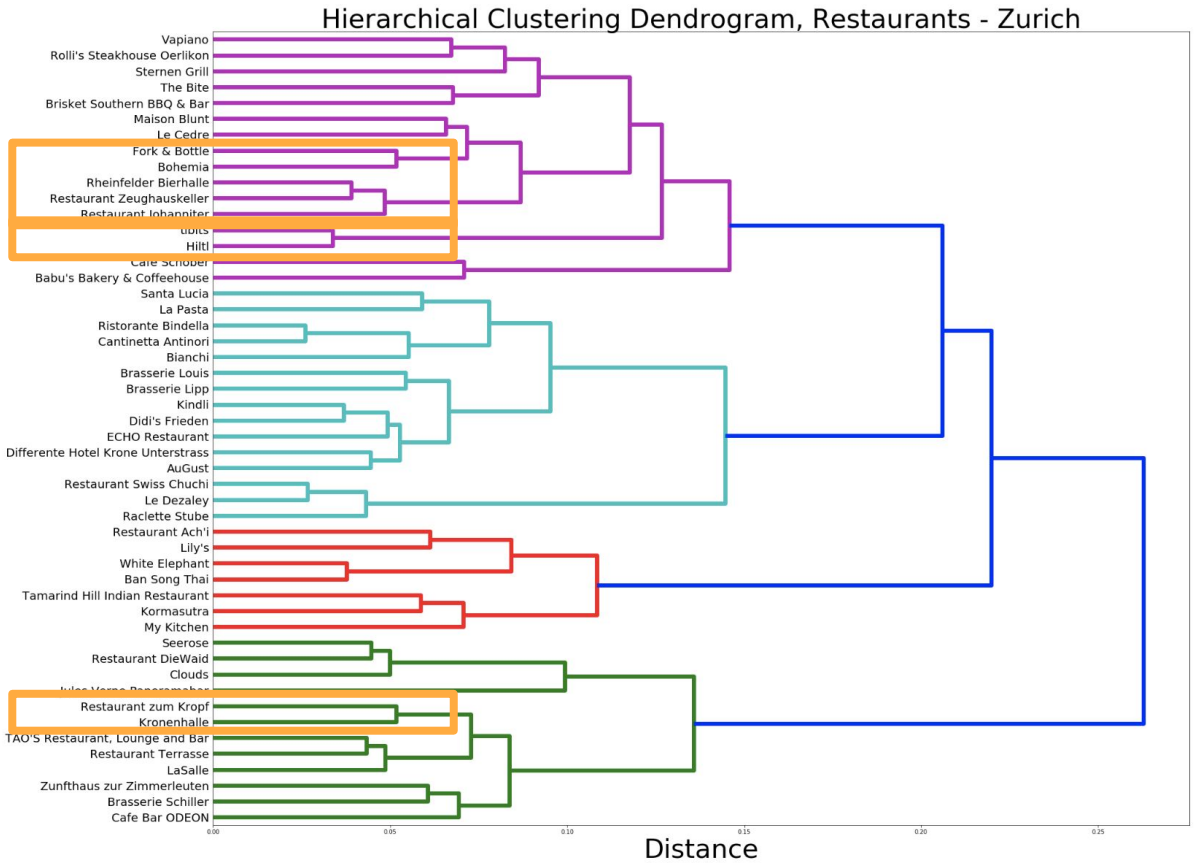


The automatic Visualisations can be changed in the Kibana UI.

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

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

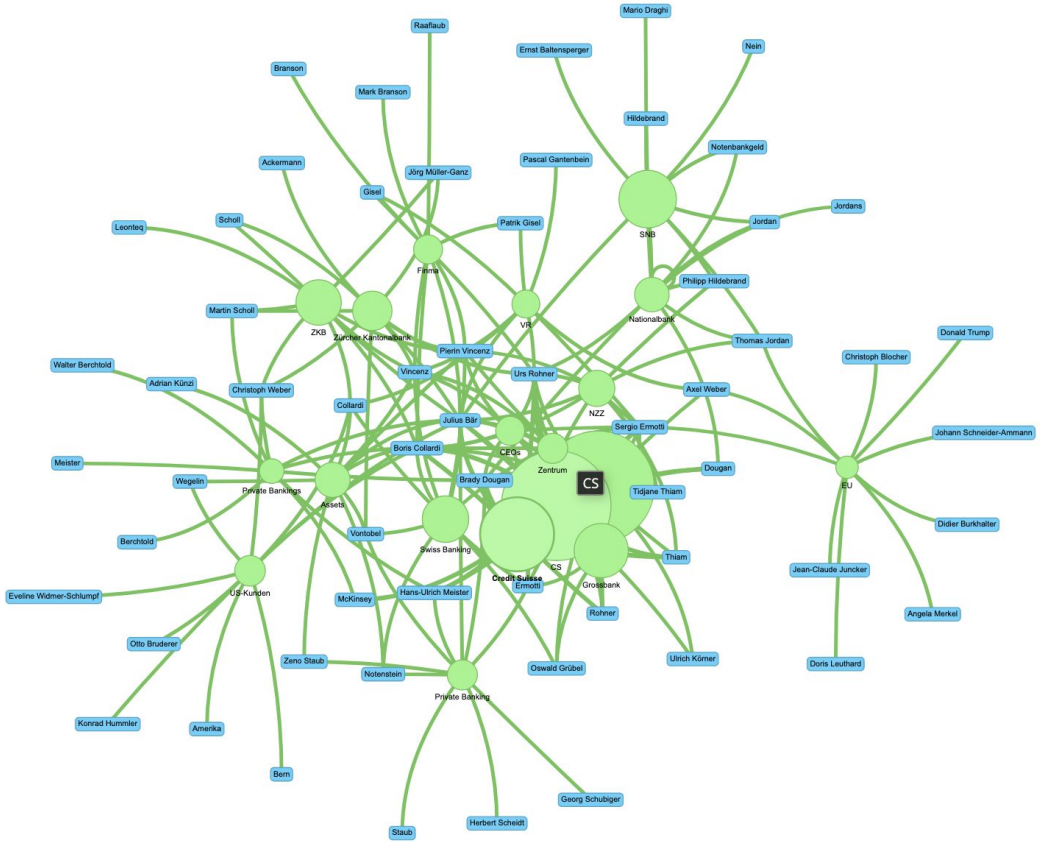
Unfortunately for me I was not aware of this until a bit later when I started being very ill. Fortunately for me my allergy does not result in anaphylactic shock so I did not end up in hospital but it has resulted in my spending the entire night vomitting and having very painful stomach cramps.

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?



Hands-On 1: How to kick-start NLP

Let’s get started

Today’s dataset: **OK Cupid** (volunteered sample from San Francisco)

- features: age, body type, diet, education, sex …
- textual data:
    - essay0- Myself summary
    - essay1- What I’m doing with my life
    - essay2- I’m really good at
    - essay3- The first thing people usually notice about me
    - essay4- Favorite books, movies, show, music, and food
    - essay5- The six things I could never do without
    - essay6- I spend a lot of time thinking about
    - essay7- On a typical Friday night I am
    - essay8- The most private thing I am willing to admit
    - essay9- You should message me if...

Break

Elasticsearch & Kibana

Elasticsearch and Kibana

- Elasticsearch is a distributed, open source, enterprise-grade search and analytics engine for all types of data, including textual, numerical, geospatial, structured, and unstructured.
- based on Apache Lucene (full text, fuzzy, auto-completion, faceting)
- Kibana is an open source analytics and visualization platform designed to work with Elasticsearch.

Common use cases:

- Large Search Systems (e.g. for Wikipedia)
- Logfile analysis

Elasticsearch Intro

- Document Database
- Organized in Indices (like DB or Schema in SQL-DB)
- Connection via REST-API - good Interface in Kibana Dev Tools
- "Schema-less", but you need to have one for our cases

Hands-On 2: Deep Dive

Deep Dive

After looking at Kibana Dashboards, you might have generated some hypotheses.

E.g.:

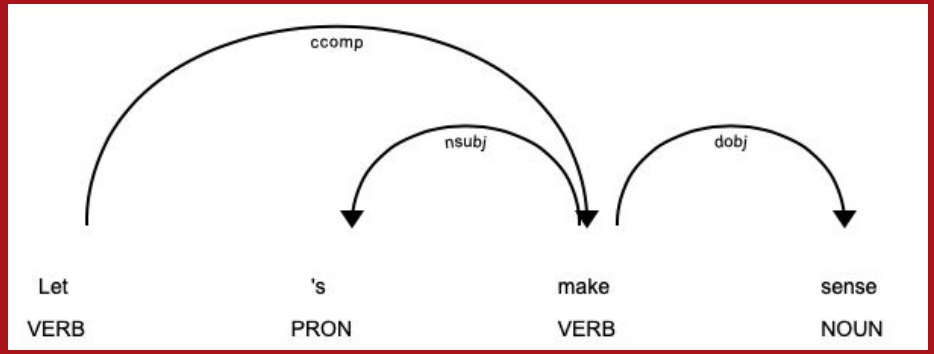
- University graduates can be identified from the text they have written and other features
- People who write about the same topics are also similar according to other features → clustering

Let’s answer some of these.

Wrap-Up

Roadmap for NLPeasy

Version 0.8	Features:	- Support for scraping / asyncio
		- Support for FastText-Vectors (both independent and in spaCy)
		- Lang support
		- Automated Map for geolocs
		- Stages: Split, clean HTML
	Package:	- More Documentation, readthedocs
		Package:
Version 0.9	Features:	- Security: User/Password; HTTPS; listening only on localhost?
		- Setup pipelines in yaml
		- CLI
		- Support for scheduled update of data
		- Support for BERT and ERNIE
Version 1.0	Features:	- Support for training of models: Spacy-NER, Word2Vec, etc.
		- Security considerations for Elastic: Password, HTTPS



LET'S MAKE SENSE

Philipp Thomann
philipp.thomann@d-one.ai

+ 41 79 747 45 66

Jacqueline Staehlin
jacqueline.staehlin@d-one.ai

+ 41 76 477 05 04

D ONE Solutions AG
Sihlfeldstrasse 58
CH-8003 Zürich