Natural Language Processing

And why it's not that opaque

September 23, 2018

Roman Jurowetzki
roman@business.aau.dk

Department of Business and Management
Aalborg University
Denmark

AALBORG UNIVERSITY
DENMARK

Agenda

NLP: Why?

Where to use?

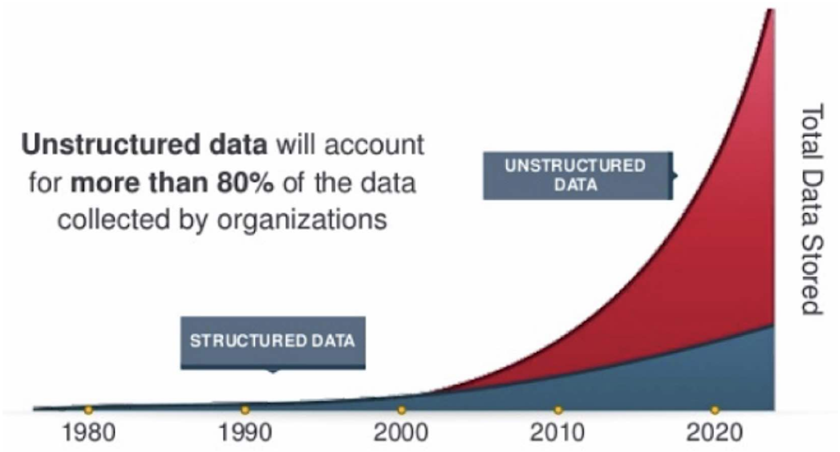
How to?

NLP: Why?

Dept. of Business and Management
Aalborg University
Denmark

Big Data

Is for the most unstructured



What would you do if you had 1000s of reliable and extremely fast assistants at hand?

Dept. of Business and Management
Aalborg University
Denmark

Where to use?

Dept. of Business and Management
Aalborg University
Denmark

3 main use-cases

An ad-hoc taxonomy

- Holistic exploration of the "discourse" – Document is the unit of analysis
- Identify and extract some elements and their relationships
- Use text for labeling of other some other observations (e.g. sentiments, classes etc.)

How to?

Dept. of Business and Management
Aalborg University
Denmark

- Text $\rightarrow$ tokens
- preprocessing: filters, stemmers, lemmatizers, bigram encoders
- Bag of words vs. Sequence
- Modelling: Depends on the task

So many great papers on neural networks $\rightarrow$ 'So', 'many', 'great','papers','on,'neural','networks'$\rightarrow$ 'many', 'great', 'paper', 'neural_network'

R

- R: *tidytext* `https://www.tidytextmining.com`
- topicmodels, quanteda

Python

- Python: *NLTK* `https://www.nltk.org/book/`
- TextBlob (simple API for NLTK)
- Fuzzywuzzy (string-matching)
- Polyglot (multilanguage - jobs)
- gensim (high-performance ML on text)
- spaCy (modern all in one high-level NLP)

NER

Identifying things in text

```python
import spacy
from spacy import displacy

text = """But Google is starting from behind. The company made a late push
into hardware, and Apple’s Siri, available on iPhones, and Amazon’s Alexa
software, which runs on its Echo and Dot devices, have clear leads in
consumer adoption."""

nlp = spacy.load('custom_ner_model')
doc = nlp(text)
displacy.serve(doc, style='ent')
```

But Google **ORG** is starting from behind. The company made a late push into hardware, and Apple **ORG** ’s Siri **PRODUCT** , available on iPhones **PRODUCT** , and Amazon **ORG** ’s Alexa **PRODUCT** software, which runs on its Echo **PRODUCT** and Dot **PRODUCT** devices, have clear leads in consumer adoption.

(O-H-E, TF-IDF) LSI, LDA & Co.

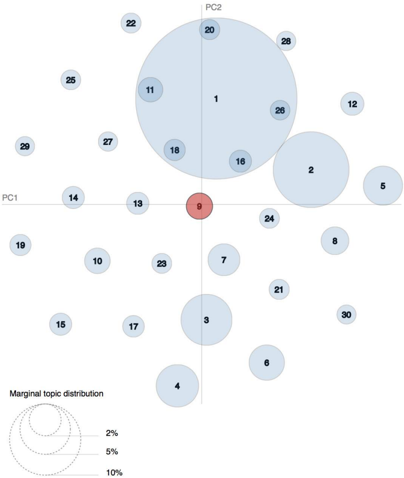
Check out Gensim: `https://radimrehurek.com/gensim/`

- Co-occurence of terms in docs
- Returns matrix of documents to topics
- Dot-product with transpose $\rightarrow$ Document-similarity adjacency matrix
- LDA mainly for topic discovery

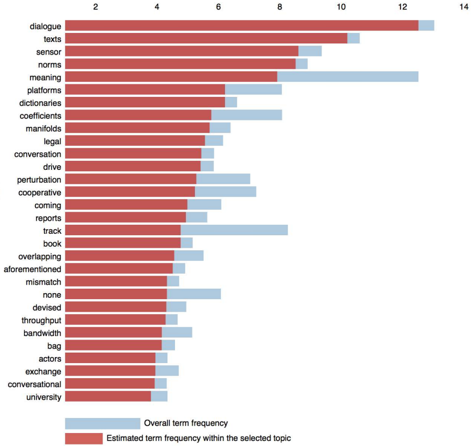
LDA

topic modelling, visualisation: `https://github.com/cpsievert/LDAvis`

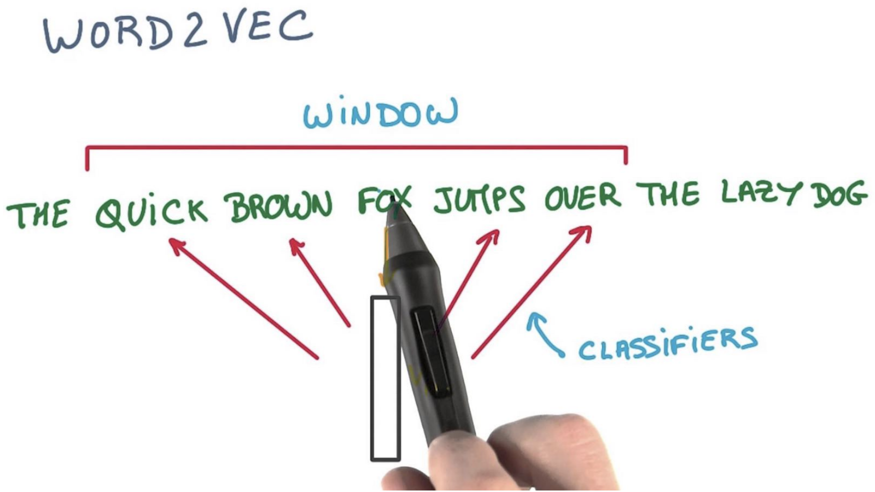
Intertopic Distance Map (via multidimensional scaling)



Top-30 Most Relevant Terms for Topic 9 (1.4% of tokens)



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)


WORD2VEC
WINDOW
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
CLASSIFIERS

Embeddings

Word2Vec, GloVe, Fasttext & Co.

```
]: w2v_model.wv.most_similar('rnn')

]: [('lstm', 0.9124458432197571),
    ('gru', 0.7881952524185181),
    ('crf', 0.7548298835754395),
    ('long_short', 0.7546945810317993),
    ('lstms', 0.7449157238006592),
    ('recurrent_network', 0.7438251972198486),
    ('attention_mechanism', 0.7428297996520996),
    ('autoencoder', 0.7388468384742737),
    ('cnn', 0.7268193960189819),
    ('encoder_decoder', 0.7267674803733826)]
```

```
In [24]: w2v_model.wv.most_similar(positive=(['neural']), negative=['cv'])

Out[24]: [('recurrent_neural', 0.4580277502536737),
          ('recurrent', 0.45371636748313904),
          ('rnns', 0.42905929684638977),
          ('rnn', 0.4164518117904663),
          ('hierarchical', 0.40013378858566284),
          ('sequence_to_sequence', 0.39685627818107605),
          ('compositional', 0.39215320348739624),
          ('term_memory', 0.38211631774902344),
          ('generative', 0.38072299957275390),
          ('lstm', 0.37855387258529663)]
```

Sentence Embedding

Doc2Vec, Avg. embeddings, TF-IDF weighted, Seq2Seq and other autoencoder based

- Average of all word vectors (easier filtering due to preceding w2v training.
- Weighted average – TF-IDF (great performance)
- Account for patterns and sequences: Autoencoder approaches (More complex) using the encoder part of a trained model to generate latent “thought vectors”.

Dept. of Business and Management
Aalborg University
Denmark

Sentence Embedding
Autoencoders for NLP