Chapter NLP:IX

IX. NLP Applications

- Frequency Extraction
- Keyword Extraction
- Cooccurrence Analysis
- Information Extraction
- Text Clustering
- Text Classification
- Machine Translation
- Text Generation
- Chat Bots
- Natural Language Understanding
- Machine Translation
- Text Generation
- Chat Bots
- Natural Language Understanding

Keyword Extraction

Overview

One task, many names:

- "Terminology mining, term extraction, term recognition, or glossary extraction, is a subtask of information extraction. The goal of terminology extraction is to automatically extract relevant terms from a given corpus." [Wikipedia]

Extended task: "Ontology learning"

- key terms and their (hierarchical) relationships (e.g. is-a, part-of, hypernym/hyperonym, synonym/antonym relations)
- See Information Extraction Slides

Evaluation:

- In Content Analysis Context: judgements on relevancy done by human experts
- Evaluation on datasets with given keyphrases

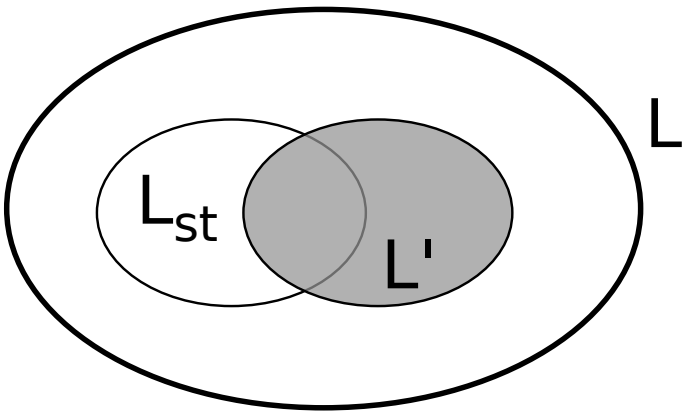
Relevancy measures based on:

- Frequency / Statistics
  - Frequency
  - $tf \cdot idf$, YAKE!, Rake, TopicRank, TextRank etc.
- Difference Corpus
  - Log likelihood / Significance
  - Characteristic elements diagnostics
- Generative
  - CopyRNN

Keyword Extraction

Overview

Specialized terms are words that occur **much more frequently** in **specialized texts of a domain** L' **(and only there)** than in other texts



We can use different paradigms

- fixed parameters for comparison
- TF/idf
- statistical tests

Keyword Extraction

Frequency

Assumption

- The more frequent, the more important
- Removing stop words helps to identify more relevant terms (see Zipf's frequency ranks)

Evaluation

- Language is Zipf distributed
- Raw frequency does not cover relevancy well

Example:

- Covid, Corona Corpus from Guardian 2020, Lemmatized, Stop Words removed, Lower cased

Approach to get n most relevant terms

- Create DTM from corpus
- Compute vector v of column sums
- order v in decreasing order
- output item 1 to n of v

Keyword Extraction

Example: Frequency

Rank	Word	Freq
1	people	132673
2	coronavirus	106041
3	government	80035
4	case	79659
5	time	74672
6	health	68820
7	work	66888
8	day	63634
9	year	63349
10	test	61854
11	week	60939
12	report	55795
13	home	52189
14	country	51617
15	update	51063
16	state	50672
17	trump	48949
18	good	48912
19	make	48763
20	pandemic	47939

The result does contain interesting vocabulary but also very unspecific words.

Keyword Extraction

tf · idf

We look for **terms that are particularly characteristic for certain documents**. These terms are **particularly frequent in a subset** of the document collection relative to the total set.

Keyword Extraction

tf · *idf*

Remember *tf* · *idf* term weighting

- Relevancy is correlated with term frequency and inversed document frequency

Evaluation

- The assumption of *idf* boosts terms which are used frequent but not everywhere

Example:

- Covid, Corona Corpus from Guardian 2020, Lemmatized, Stop Words removed, Lower cased

Approach to get n most relevant terms

- Create DTM from corpus
- Compute vector v of mean *tf* · *idf* values for each term
- order v in decreasing order
- output item 1 to n of v

Keyword Extraction

Example: $tf \cdot idf$

Rank	word	freq	tfidf
1	update	51063	1.69
2	trump	48949	1.20
3	case	79659	1.01
4	test	61854	1
5	death	45805	0.80
6	health	68820	0.80
7	report	55795	0.76
8	state	50672	0.73
9	government	80035	0.70
10	biden	16017	0.66
11	police	24582	0.64
12	australia	24652	0.64
13	virus	39680	0.63
14	school	30373	0.63
15	australian	22109	0.62
16	people	132673	0.62
17	president	26034	0.61
18	care	28376	0.60
19	hospital	25983	0.60
20	uk	45338	0.59

The result does nor correlate with the frequency. Furthermore, we see that very specific but frequent terms are ranked down (coronavirus) because the *df* is high. Interpretation: Words which have some specifity for the coronavirus topic, or related events.

Keyword Extraction

Corpus Comparison

Difference based Term Extraction methods follow a different approach:

- Comparing frequencies in a target corpus T with frequencies in a general comparison corpus C
- Significant deviation in T from expected term distribution measured in C is considered as relevancy criterion

Significance tests used in Content Analysis and Terminology Mining

- Log Likelihood [Dunning 1993] [Rayson, Garside 2000]
- Characteristic elements diagnostics [Lebart, Salem 1994]

Keyword Extraction

Log Likelihood significance test

	Corpus 1 (T)	Corpus 2 (C)	Total
Frequency of word	a	b	a+b
Frequency of other words	c-a	d-b	c+d-a-b
Total	c	d	c+d

Log Likelihood Test [Rayson, Garside 2000]
Goodness-of-Fit Test based on Frequency Estimator

- $E_i = \frac{N_i \Sigma_i O_i}{\Sigma_i N_i}$
- $-2\ ln\ \lambda(LL) = 2\ \Sigma_i\ O_i\ ln\left(\frac{O_i}{E_i}\right)$
- $E_1 = c \times (a+b)/(c+d)$
- $E_2 = d \times (a+b)/(c+d)$
- $LL = 2 \times ((a \times log(a/E_1))$
  + (b $\times log(b/E_2)))$

Alternative: Binominal Estimator, χ^2 Estimator

Example: Covid, Corona Corpus from Guardian 2020, Lemmatized, Stop Words removed, Lower cased, Reference Corpus 4 Mio engl. sentences

Approach to get n most relevant terms

- Create DTM from corpus
- Compute vector v of *tf* sums for each term
- Apply formula and order v in decreasing order
- output item 1 to n of v

Keyword Extraction

Example: Log Likelihood Significance Testing

Rank	word	freq	LL_sig
1	coronavirus	106041	207215.77
2	pandemic	47939	91842.32
3	lockdown	38436	73856.07
4	virus	39680	64090.08
5	update	51063	61803.26
6	trump	48949	54972.23
7	test	61854	52976.68
8	case	79659	49868.62
9	health	68820	48706.36
10	photograph	34677	47455.74
11	people	132673	38714.62
12	outbreak	23588	38517.27
13	government	80035	36597.14
14	infection	24983	35126.29
15	guardian	21497	33563.35
16	crisis	28603	31787.43
17	biden	16017	29891.74
18	mask	19477	29778.42
19	restriction	22614	29613.70
20	covid	15014	29339.01

The result does not correlate with the frequency. The result reflects 2020 in words quite well.

Keyword Extraction

Comparison of results

Rank	word$_{freq}$	freq	word$_{tfidf}$	freq$_{tfidf}$	tfidf	word$_{sig}$	freq$_{sig}$	LL_sig
1	people	132673	update	51063	1.69	coronavirus	106041	207215.77
2	coronavirus	106041	trump	48949	1.20	pandemic	47939	91842.32
3	government	80035	case	79659	1.01	lockdown	38436	73856.07
4	case	79659	test	61854	1	virus	39680	64090.08
5	time	74672	death	45805	0.80	update	51063	61803.26
6	health	68820	health	68820	0.80	trump	48949	54972.23
7	work	66888	report	55795	0.76	test	61854	52976.68
8	day	63634	state	50672	0.73	case	79659	49868.62
9	year	63349	government	80035	0.70	health	68820	48706.36
10	test	61854	biden	16017	0.66	photograph	34677	47455.74
11	week	60939	police	24582	0.64	people	132673	38714.62
12	report	55795	australia	24652	0.64	outbreak	23588	38517.27
13	home	52189	virus	39680	0.63	government	80035	36597.14
14	country	51617	school	30373	0.63	infection	24983	35126.29
15	update	51063	australian	22109	0.62	guardian	21497	33563.35
16	state	50672	people	132673	0.62	crisis	28603	31787.43
17	trump	48949	president	26034	0.61	biden	16017	29891.74
18	good	48912	care	28376	0.60	mask	19477	29778.42
19	make	48763	hospital	25983	0.60	restriction	22614	29613.70
20	pandemic	47939	uk	45338	0.59	covid	15014	29339.01

What do you think? Exact evaluation of behavior can only be achieved by testing on **standardized evaluation datasets**. See remarks slide for papers and evaluation datasets.

Remarks:

- Lot's of approaches to extract terms...
- Corpus comparison methods are usually better than frequency based methods. **Disadvantage?: terms are observed independently of each other**
- LL and "characteristic elements diagnostic" well established in corpus linguistic literature
- **Extension Difference Analysis:** Method for determining discriminatory terms, in which the different distribution of word forms in texts is evaluated. The *basis is a set of reference texts* against which a target text is compared.
  - Analysis of keywords in press releases ("words of the day")
  - Analysis of web pages
  - Analysis of social media messages
  - Digital Humanities
  - ...
- More recent and related algorithms are:
  - Yake!: a light-weight unsupervised automatic keyword extraction method which rests on statistical text features extracted from single documents to select the most relevant keywords of a text. **Extracts phrases!** [Campos et. al. 2020]
  - TextRank: proposes two innovative unsupervised methods for graph based (concordances) keyword and sentence extraction. [Mihalcea, Tarau 2004]
  - Rake: is more computationally efficient than TextRank while achieving higher precision and comparable recall scores. [Rose et. al. 2010]
  - Single Rank: proposes to use a small number of nearest neighbor documents to provide more knowledge to improve single document keyphrase extraction. [Wan, Xiau 2008]
  - **CopyRnn**: generative model for keyphrase prediction with an encoder-decoder framework [Meng et. al. 2017]