Introduction to intelligent systems

Natural language processing

Mikkel N. Schmidt

Technical University of Denmark,
DTU Compute, Department of Applied Mathematics and Computer Science.

Overview

1. Text processing

2. Demo: List manipulation in Python

3. Tasks

Feedback group

- Marius Drachmann Niss
- Joseph An Duy Nguyen
- Tobias Rodrigues Bjerre
- Marcus Zabell Olssen

Learning objectives

I Data representation in the computer.

I Bag of words representation.

II Term frequency-inverse document frequency (TF-IDF).

I Understand the concepts and definitions, and know their application. Reason about the concepts in the context of an example. Use correct technical terminology.

II As above plus: Read, manipulate, and work with technical definitions and expressions (mathematical and Python code). Carry out practical computations. Interpret and evaluate results.

Text processing

Natural language processing

Syntax Part-of-speech tagging, parsing, grammar induction etc.

Semantics Lexical semantics, translation, named entity recognition, sentiment analysis, topic analysis, etc.

Discourse Summarization, discourse analysis

Steven Bird, Ewan Klein, and Edward Loper, "Natural Language Processing with Python—Analyzing Text with the Natural Language Toolkit", http://www.nltk.org/book/

Wikipedia, https://en.wikipedia.org/wiki/Natural_language_processing

Levels of analysis

- Sequence of characters
- Sequence of words
- Sequence of sentences

Document search

Corpus A set of documents (text)

Query A text string

Goal Return top-N relevant documents according to the query

Which document is most relevant

Consider the search query:

`what is a cat`

Which of the following “documents” (sentences) is most relevant, and why?

1. `Cats are small domesticated mammals with soft fur and retractile claws.`
2. `We need to promote direct foreign investment, which is what a free trade agreement would do.`
3. `Is a banana a fruit or a herb? The banana plant is technically regarded as a herb, because the stem does not contain woody tissue.`
4. `The Caterpillar Inc. (CAT) stock price is now well beyond what most analysts predicted.`

(Words matching the query are highlighted, case and punctuation ignored.)

Desiderata

- Word endings should be ignored
- Common words should be ignored and rare words should be given emphasis
- System should distinguish between homographs (words spelled the same with different meaning)
- User should write a better query
- ... more?

Stemming

- Reduce words to their word stem, base or root form.
- Related words map to the same stem
- Many search engines treat words with the same stem as synonyms (conflation)

Example: **`argue`**, **`argued`**, **`argues`**, **`arguing`** all reduce to the stem **`argu`**

Example of stemming

Before stemming

```
Zebras are several species of African equids (horse family)
united by their distinctive black and white striped coats.
```

After stemming

```
zebra are sever speci of african equid hors famili unit by
their distinct black and white stripe coat
```

Bag-of-words representation

	Doc. 1	Doc. 2
african	1	0
although	0	1
and	1	0
are	1	0
bear	0	1
black	1	0
by	1	0
close	0	1
coat	1	0
distinct	1	0
equid	1	0
famili	1	0
giraff	0	1
hors	1	0
is	0	1
it	0	1
mark	0	1
most	0	1
of	1	1
okapi	0	1
relat	0	1
reminisc	0	1
sever	1	0
speci	1	0
stripe	1	1
the	0	2
their	1	0
to	0	1
unit	1	0
white	1	0
zebra	1	1

Sentences

1. Zebras are several species of African equids (horse family) united by their distinctive black and white striped coats.
2. Although the okapi bears striped markings reminiscent of zebras it is most closely related to the giraffe.

- A bag-of-words sentence/document can be seen as a point in a high-dimensional vector space

Exercise: Dot product

We can use the dot product between the word occurence vectors as a measure of similarity between documents

- Compute the dot product between the two sentences
- Can you think of pros and cons of using the dot product to measure similarity?

Sentences

1. `Zebras are several species of African equids (horse family) united by their distinctive black and white striped coats.`
2. `Although the okapi bears striped markings reminiscent of zebras it is most closely related to the giraffe.`

Words in common in the sentences

	Doc. 1	Doc. 2
`of`	1	1
`stripe`	1	1
`zebra`	1	1

TF and IDF

TF: Term frequency How *frequently* a term occurs in a document. Since documents can have different length. TF is often normalized by the document length.

$$\mathrm{TF}(t,d) = \frac{\#\text{occurences of term } t \text{ in document } d}{\#\text{words in document } d}$$

IDF: Inverse document frequency How *important* a term is. Some terms like *`is`*, *`that`*, and *`the`* may appear a lot, but have little importance.

$$\mathrm{IDF}(t) = \log\left(\frac{\#\text{documents in corpus}}{\#\text{documents with term } t}\right)$$

Exercise: TF-IDF

- Consider a document that contains 100 words, wherein
  - the word *the* appears 3 times and
  - the word *cat* appears 3 times
- The document is part of a 10 000 document corpus, wherein
  - 4 900 of the documents contain the word *the* and
  - 123 of the documents contain the word *cat*

Compute the TF and IDF for the terms *the* and *cat*

TF and IDF

$$TF = \frac{n_{t,d}}{n_d} \qquad IDF = \log\left(\frac{N}{n_t}\right)$$

$n_{t,d}$	Number of occurences of term t in document d
n_d	Number of terms in document d
n_t	Number of documents with term t
N	Total number of documents

Exercise: TF-IDF

- Consider a document that contains 100 words, wherein
  - the word *the* appears 3 times and
  - the word *cat* appears 3 times
- The document is part of a 10 000 document corpus, wherein
  - 4 900 of the documents contain the word *the* and
  - 123 of the documents contain the word *cat*

Compute the TF and IDF for the terms *the* and *cat*

TF and IDF

$$TF = \frac{n_{t,d}}{n_d} \qquad IDF = \log\left(\frac{N}{n_t}\right)$$

$n_{t,d}$ Number of occurences of term t in document d

n_d Number of terms in document d

n_t Number of documents with term t

N Total number of documents

Solution

the

$$\mathrm{TF} = \frac{3}{100} = 0.03$$

$$\mathrm{IDF} = \log\left(\frac{10\,000}{4\,900}\right) \approx 0.7133$$

cat

$$\mathrm{TF} = \frac{3}{100} = 0.03$$

$$\mathrm{IDF} = \log\left(\frac{10\,000}{123}\right) \approx 4.398$$

TF-IDF score

$$\text{TF-IDF}(d, q) = \underbrace{\sum_{t \in q}}_{\text{Sum over all terms in query } q} \frac{n_{t,d}}{n_d} \cdot \log\left(\frac{N}{n_t}\right)$$

$n_{t,d}$	Number of occurences of term t in document d
n_d	Number of terms in document d
n_t	Number of documents with term t
N	Total number of documents

Exercise: TF-IDF

- What happens if no documents contain one of the search terms?

TF-IDF

$$\text{TF-IDF}(d, q) = \sum_{t \in q} \frac{n_{t,d}}{n_d} \cdot \log\left(\frac{N}{n_t}\right)$$

$n_{t,d}$ Number of occurences of term t in document d

n_d Number of terms in document d

n_t Number of documents with term t

N Total number of documents

Exercise: TF-IDF

- What happens if no documents contain one of the search terms?

TF-IDF

$$\text{TF-IDF}(d, q) = \sum_{t \in q} \frac{n_{t,d}}{n_d} \cdot \log\left(\frac{N}{n_t}\right)$$

$n_{t,d}$ Number of occurences of term t in document d

n_d Number of terms in document d

n_t Number of documents with term t

N Total number of documents

Solution: Division by zero!

Okapi BM25

$$\text{BM25}(d,q) = \sum_{t \in q} \frac{n_{t,d} \cdot (k_1 + 1)}{n_{t,d} + k_1 \cdot (1 - b + b \cdot \frac{n_d}{\text{avgdl}})} \cdot \log\left(\frac{N - n_t + 0.5}{n_t + 0.5}\right)$$

$n_{t,d}$ Number of occurences of term t in document d

n_d Number of terms in document d

n_t Number of documents with term t

N Total number of documents

avgdl Average document length $\frac{1}{N}\sum_d n_d$

b Parameter ($b \in [0,1]$, default $b = 0.75$)

k_1 Parameter ($k_1 > 0$, default $k_1 = 1.2$)

Britta Weber, “BM25 demystified”, https://www.youtube.com/watch?v=v3Ko0CwgTZ0

Stephen Robertson and Hugo Zaragoza, “The probabilistic relevance framework: BM25 and beyond”

Wikipedia, https://en.wikipedia.org/wiki/Okapi_BM25

Okapi BM25 parameters

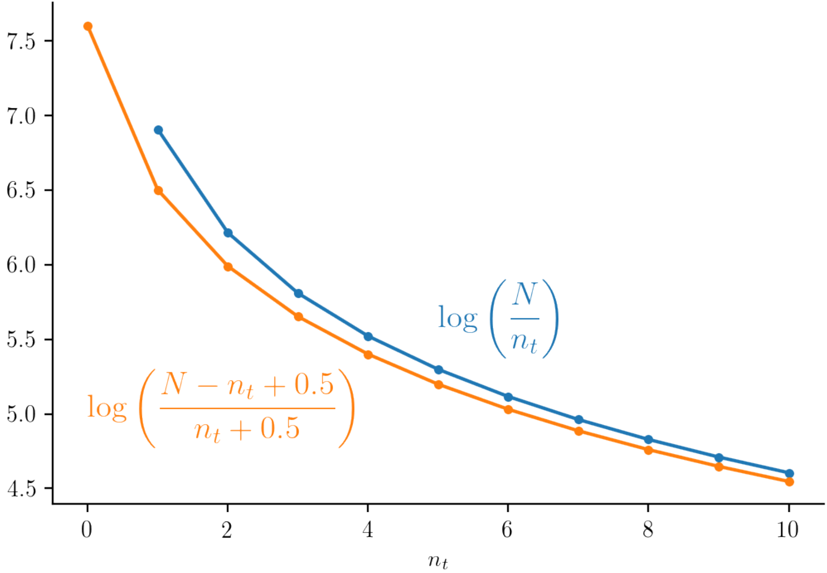
$$\text{BM25}(d, q) = \sum_{t \in q} \frac{n_{t,d} \cdot (k_1 + 1)}{n_{t,d} + k_1 \cdot (1 - b + b \cdot \frac{n_d}{\text{avgdl}})} \cdot \log\left(\frac{N - n_t + 0.5}{n_t + 0.5}\right)$$

What do the parameters b and k_1 control?

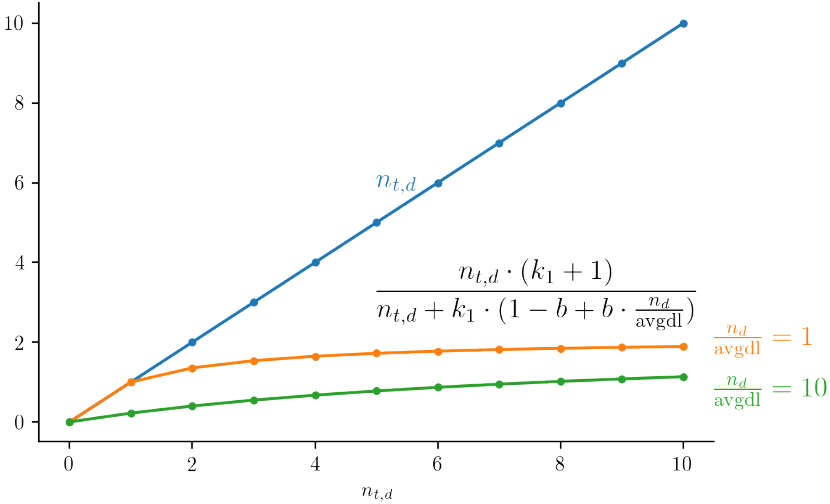
k_1 How much to weigh the term-frequency
$k_1 = 0$: No term-frequency. $k_1 \to \infty$: Raw term-frequency.

b How much to scale with the document length
$b = 0$: No scaling. $b = 1$: Full scaling.

The “inverse document frequency” term in Okapi BM25



The “term-frequency” in Okapi BM25



Exercise: Okapi BM25

BM25

$$\mathrm{BM25}(d, q) = \sum_{t \in q} \frac{n_{t,d} \cdot (k_1 + 1)}{n_{t,d} + k_1 \cdot (1 - b + b \cdot \frac{n_d}{\mathrm{avgdl}})} \cdot \log\left(\frac{N - n_t + 0.5}{n_t + 0.5}\right)$$

- Consider a document that contains 100 words, wherein
  - the word *the* appears 3 times and
  - the word *cat* appears 3 times
- The document is part of a 10 000 document corpus, wherein
  - 4 900 of the documents contain the word *the* and
  - 123 of the documents contain the word *cat*
- The average document length in the corpus is 150

$n_{t,d}$	Number of occurences of term t in document d
n_d	Number of terms in document d
n_t	Number of documents with term t
N	Total number of documents
avgdl	Average document length
b	$b = 0.75$
k_1	$k_1 = 1.2$

Compute the BM25-score for the query *the cat*

Exercise: Okapi BM25

Solution

$$
\begin{aligned}
\mathrm{BM25}(d, q) &= \sum_{t \in q} \frac{n_{t,d} \cdot (k_1 + 1)}{n_{t,d} + k_1 \cdot (1 - b + b \cdot \frac{n_d}{\mathrm{avgdl}})} \cdot \log\left(\frac{N - n_t + 0.5}{n_t + 0.5}\right) \\
&= \frac{3 \cdot (1.2 + 1)}{3 + 1.2 \cdot (1 - 0.75 + 0.75 \cdot \frac{100}{150})} \cdot \log\left(\frac{10\,000 - 4\,900 + 0.5}{4\,900 + 0.5}\right) + \\
&\quad \frac{3 \cdot (1.2 + 1)}{3 + 1.2 \cdot (1 - 0.75 + 0.75 \cdot \frac{100}{150})} \cdot \log\left(\frac{10\,000 - 123 + 0.5}{123 + 0.5}\right) \\
&\approx 1.692 \cdot 0.040 + 1.692 \cdot 4.382 \approx \underline{7.483}
\end{aligned}
$$

Demo: List manipulation in Python

Demo: List manipulation in Python

```
>>> my_list = [1, 1, 2, 3, 5, 8, 13]
```

Indexing (accessing elements)

```
>>> my_list[0]
1
```

```
>>> my_list[4]
5
```

Length

```
>>> len(my_list)
7
```

Check if element occurs

```
>>> 1 in my_list
True
```

```
>>> 17 in my_list
False
```

Count occurences

```
>>> my_list.count(1)
2
```

```
>>> my_list.count(2)
1
```

```
>>> my_list.count(17)
0
```

List comprehension

```
>>> [x**2 for x in my_list]
[1, 1, 4, 9, 25, 64, 169]
```

Demo: List manipulation in Python

```
>>> my_list = [1, 1, 2, 3, 5, 8, 13]
>>> my_query = [3, 4, 5]
```

How many of the numbers in `my_query` occur in `my_list`?
For-loop

```
>>> occurences = 0
>>> for q in my_query:
...     if q in my_list:
...         occurences += 1
```

List comprehension

```
>>> occurences = [q in my_list for q in my_query].count(True)
```

Tasks

Tasks

Today

1. Implement document search for the *animals.txt* data set
    - Work through the tasks in the script `05-DocumentSearch.ipynb` (contains a solution template)
    - Given a query (one or more words), your program must return the top-5 best matching documents
    - Start by implementing TF-IDF and then move on to Okapi BM25
2. Today's feedback group
    - Marius Drachmann Niss
    - Joseph An Duy Nguyen
    - Tobias Rodrigues Bjerre
    - Marcus Zabell Olssen

Lab report

- Lab 2: Symbolic AI (Deadline: Thursday 28 September 20:00)