Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Text Representation
Part 1 Bag-of-words

Text, Web and Social Media Analytics Lab

Prof. Dr. Diana Hristova

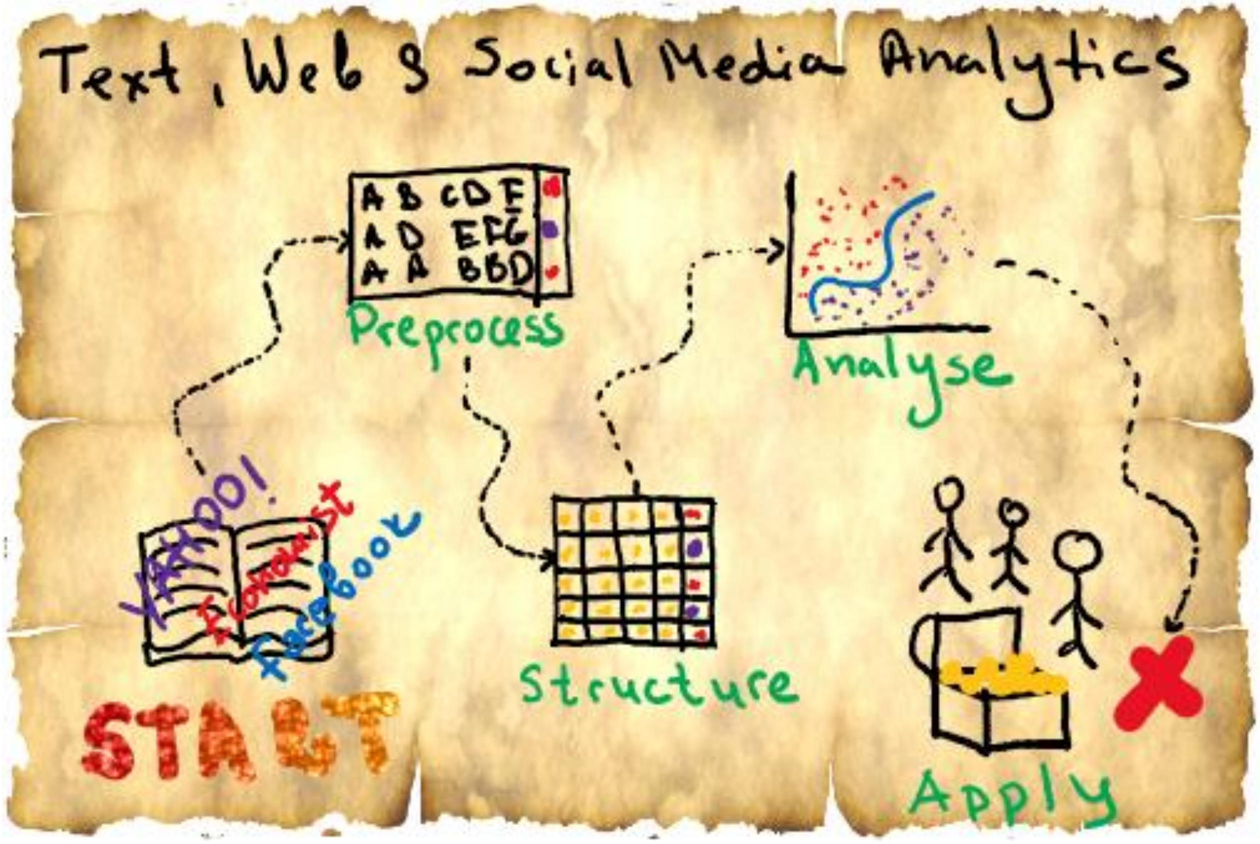
Exercise 2: Preprocessing

Who would like to present?

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

What did we learn last week?

Treasury map: Text Representation



Course structure

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Date	Lecture	Exercise
12.04.2021	Introduction	Technical Installation
19.04.2021	Text Preprocessing	Projects kick-off
26.04.2021	**Text Representation**	**Preprocessing Newsgroups**
03.05.2021	Text Representation (2)	**Text Representation Newsgroups**
10.05.2021	Text Classification	Text Representation Newsgroups (2)
17.05.2021	Text Clustering	Newsgroups Topic Classification
31.05.2021	Text Mining in Social Media	Newsgroups Topic Clustering
07.06.2021	Mining Social Graphs	Sentiment Analysis and Time Series in Twitter
14.06.2021	Projects Status Update	Projects Status Update
21.06.2021	Web Analytics	Mining Social Graphs in Twitter
28.06.2021	Mock Exam	Web Analytics in E-commerce
05.07.2021	Final Presentation	Final Presentation
19.07.2021	Submit Code & Written report	
t.b.a.	Exam	

What will we learn today?

Hochschule für
Wirtschaft und Recht Berlin
Berlin School of Economics and Law

At the end of this lecture, you will:

1. Know the motivation for deriving structured text representation
2. Understand and apply the following main types of methods for text representation as well as know their pros and cons:
   - Bag-of-words: One-hot encoding, Relative, Absolute and Tf-Idf frequencies
   - N-grams

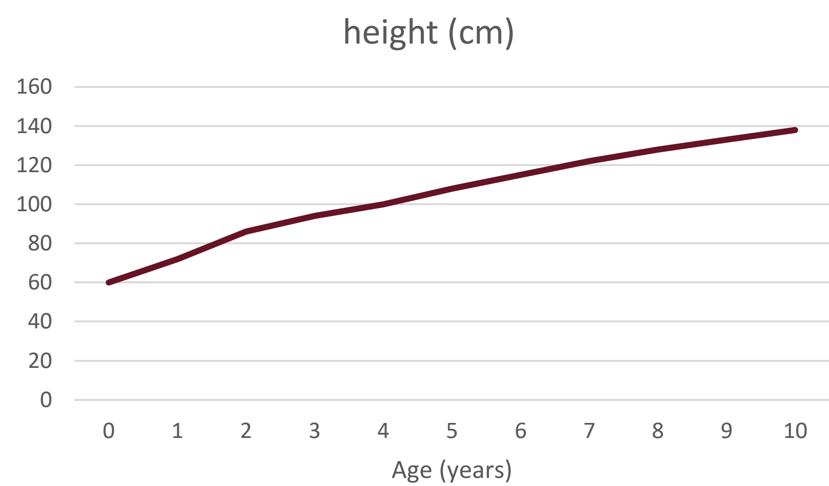
Motivation: Why can‘t we use text data directly as input for algorithms?

Any ideas?

Motivation: Why can‘t we use text data directly as input for algorithms?

- Existing machine learning/statistical approaches require structured input.
- Example: linear regression for the height of girls depending on their age

age (years)	Gender (f=female)	height (cm)
0	f	60
1	f	72
2	f	86
3	f	94
4	f	100
5	f	108
6	f	115
7	f	122
8	f	128
9	f	133
10	f	138



Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Motivation: Why can‘t we use text data directly as input for algorithms? (2)

- **But:** texts are unstructured, we as humans structure them in our heads.
- How can we structure texts, so that machine learning approaches can be applied to them?
- By using ***vectorization*** i.e., the process of representing texts numerically as vectors consisting of features.

Preprocessed text

car organization university maryland college park line wonder enlighten car see day door sport car look late early call bricklin door small addition bumper separate rest body know tellme model engine specs year production car history info funky look car mail thank bring neighborhood

0.1
1
0.3
5
0.7
2.8
4
0
1.1
3.8

Text representation: Dictionary

- A **dictionary** is the set of all words used in a corpus. It is similar to language vocabulary.
- **Note:** usually the dictionary is determined after preprocessing
- Example: let the corpus consist of the following sentences:
  - Doc1: “I like apples, but don’t like oranges.” $\xrightarrow{\text{preprocess}}$ “like appl like orang”
  - Doc2: “I like oranges.” ➔ “like orang”
  - Doc3: “I prefer apples to oranges.” ➔ “prefer appl orang”

➔ **Dictionary: ?**

Text representation: Dictionary

- A **dictionary** is the set of all words used in a corpus. It is similar to language vocabulary.
- **Note:** usually the dictionary is determined after preprocessing
- Example: let the corpus consist of the following sentences:
  - Doc1: “I like apples, but don’t like oranges.” $\xrightarrow{\text{preprocess}}$ “like appl like orang”
  - Doc2: “I like oranges.” ➔ “like orang”
  - Doc3: “I prefer apples to oranges.” ➔ “prefer appl orang”

➔ **Dictionary:** like, appl, orang, prefer

Text representation: Bag-of-words

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

- The Bag-of-words approach uses a vector where each feature represents one word in the dictionary.
- The feature values are the word importances
- Example: let the preprocessed corpus consist of the following sentences:
  - Doc1: “like appl like orang”
  - Doc2: “like orang”
  - Doc3: “prefer appl orang”

Bag-of-words

Doc	like	appl	orang	prefer
Doc1				
Doc2				

? How to determine the feature importances?

Text representation: Bag-of-words

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

? How to determine the feature importances?

Any ideas?

Bag-of-words: One-hot encoding

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

- **One-hot encoding:** use binary values depending on the appearance of the word.
- **Example:** let the preprocessed corpus consist of the following sentences:
  - Doc1: “like appl like orang”
  - Doc2: “like orang”
  - Doc3: “prefer appl orang”

Doc	like	appl	orang	prefer
Doc1	1	1	1	0
Doc2	1	0	1	0
Doc3	0	1	1	1

Bag-of-words: One-hot encoding

What is a *disadvantage* of one-hot encoding?

a. **Not possible to compare documents of different length**

b. **All words are considered equally important**

c. **Not useful for short texts**

Bag-of-words: One-hot encoding

- **One-hot encoding:** use binary values depending on the appearance of the word.
- **Example:** let the preprocessed corpus consist of the following sentences:
  - Doc1: “like appl like orang”
  - Doc2: “like orang”
  - Doc3: “prefer appl orang”

Doc	like	appl	orang	prefer
Doc1	1	1	1	0
Doc2	1	0	1	0
Doc3	0	1	1	1

- **Advantage:** easy to calculate and interpret, useful for short texts
- **Disadvantage:** words that appear often are treated in the same way as rare words.

Bag-of-words: Absolute frequency

- **Absolute term frequency:** value is the number of appearances of a word.

➔ More common words are more important.

- **Example:** let the preprocessed corpus consist of the following sentences:
  - Doc1: “like appl like orang”
  - Doc2: “like orang”
  - Doc3: “prefer appl orang”

Doc	like	appl	orang	prefer
Doc1	2	1	1	0
Doc2	1	0	1	0
Doc3	0	1	1	1

- **Advantage:** considers word importance, interpretable
- **Disadvantage:** not comparable between documents (i.e. long vs. short ones).

Bag-of-words: Relative frequency

- **Relative term frequency:** absolute term frequency/length of document
- **Example:** let the preprocessed corpus consist of the following sentences:
  - Doc1: “like appl like orang”
  - Doc2: “like orang”
  - Doc3: “prefer appl orang”

Doc	like	appl	orang	prefer	Doc length
Doc1	0.5	0.3	0.3	0.0	4
Doc2	0.5	0.0	0.5	0.0	2
Doc3	0.0	0.3	0.3	0.3	3

- **Advantage:** considers document length, interpretable
- **Disadvantage:** common words among all documents are overweighed (e.g. ‘annual’ in annual reports).

Why is it important not to overweigh common words?

Bag-of-words: TF-IDF frequency

- **TF-IDF frequency:** considers the occurrence among all documents

$$tf_{idf}(word) = freq(word) * \boxed{log(\frac{total\ number\ of\ docs}{number\ of\ docs\ with\ word})}\ \text{IDF}$$

- If a word appears in all documents

$$log\left(\frac{total\ number\ of\ docs}{number\ of\ docs\ with\ word}\right) = log(1) = 0 \Rightarrow tf_{idf}(word) = 0$$

- If a word appears in only 1% of the documents

$$log\left(\frac{total\ number\ of\ docs}{number\ of\ docs\ with\ word}\right) = log(100/1) = 4.6$$

$$\Rightarrow tf_{idf}(word) = 4.6 freq(word)$$

- In this case the frequency is increased approx. 5 times.
- **Note:** Implementations in Python differ with respect to the used frequency and the form of the IDF term ➔ Always use only one package at a time.

Bag-of-words: TF-IDF frequency (2)

- **Example:** let the preprocessed corpus co
  - Doc1: “like appl like orang”
  - Doc2: “like orang”
  - Doc3: “prefer appl orang”

Relative frequency				
Doc	like	appl	orang	prefer
Doc1	0.5	0.3	0.3	0.0
Doc2	0.5	0.0	0.5	0.0
Doc3	0.0	0.3	0.3	0.3

Doc	like	appl	orang	prefer
Doc1	0.09	0.05	0.00	0.00
Doc2	0.09	0.00	0.00	0.00
Doc3	0.00	0.05	0.00	0.14
Number of docs with word	2	2	3	1
Log(Total/Number of docs with word)	0.18	0.18	0	0.48

- **Advantage:** Considers the occurrence among all documents (important for classification)
- **Disadvantage:** Difficult interpretation

What are disadvantages of the Bag-of-words approach?

Vectorization: N-grams

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

- Bag-of-words approach considers each word in isolation

➔ The context is completely ignored.

Example:

- Doc1: “The meal is horrible, but the service is great.”
- Doc2: “The service is horrible, but the meal is great.”

➔ Regardless of the chosen frequency, Doc1 and Doc2 will have the same Bag-of-words representation

▶ Use combinations of consecutive words

N-grams (2)

Example:

- Doc1: “The meal is horrible, but the service is great.”
- Doc2: “The service is horrible, but the meal is great.”
- Doc3: “Both the service and the meal are great.”

Preprocessed corpus:

- Doc1: “meal horribl servic great”
- Doc2: “servic horribl meal great”
- Doc3: “servic meal great”

N-grams (3)

- **N-grams**: a set of n consecutive words

    - Doc1: “meal horribl servic great”
    - Doc2: “servic horribl meal great”
    - Doc3: “servic meal great”

↓

horribl meal	**horribl servic**	**meal great**	**meal horribl**	**servic great**	**servic horribl**	**servic meal**

- You can use any type of the frequencies from Bag-of-words (e.g. One-hot, TF-IDF)
- **Advantage:** Considers the context of the word
- **Disadvantage: ?**

N-grams (3)

- **N-grams**: a set of n consecutive words

  - Doc1: “meal horribl servic great”
  - Doc2: “servic horribl meal great”
  - Doc3: “servic meal great”

horribl meal	horribl servic	meal great	meal horribl	servic great	servic horribl	servic meal

- You can use any type of the frequencies from Bag-of-words (e.g. One-hot, TF-IDF)
- **Advantage:** Considers the context of the word
- **Disadvantage:** They generate a much bigger dictionary with many zero values

Vectorization: Removing common and rare words

Detour

- When analysing structured data, part of the data preparation is feature selection i.e., reduce the input features to the relevant ones.
- This is done to reduce noise in the data provided as input to the models (see Text Classification and Clustering).
- One simple way to apply feature selection in text analytics is to remove words that appear very commonly (e.g. ‘news’ in a news corpus) or very rarely (e.g. some named entities) among all documents.
- **Implementation in Python (sklearn)**
  - max_df: remove words from the corpus that appear in more than *max_df* documents
  - min_df: remove words from the corpus that appear in less than *min_df* documents

Summary and Outlook

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Summary:

- Structuring text data is important as most data analysis algorithms require structured data as input.
- Vectorization is the process of representing texts numerically as vectors consisting of features.
- The Bag-of-words vectorization approach uses single words as features. The values can be calculated by:
  - ✓ One-hot encoding
  - ✓ Absolute, Relative and Tf-IDF frequencies
- N-grams consider two consecutive words instead of one word in isolation.
- **Outlook:** Embedding-based approaches offer an alternative to bag-of-words and n-grams that better considers context information.

Questions?

Exercise 3

In a minute, six break-out rooms will be created. Choose the room that corresponds to your group in Moodle e.g. Room 1= Group 1. In your project group discuss and document the solution for Exercise 3 (in Moodle).