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

Text preprocessing

Text, Web and Social Media Analytics Lab

Prof. Dr. Diana Hristova

Survey

How far did you get with Exercise 1?

a. Only finished Question 1 (Google Colab)

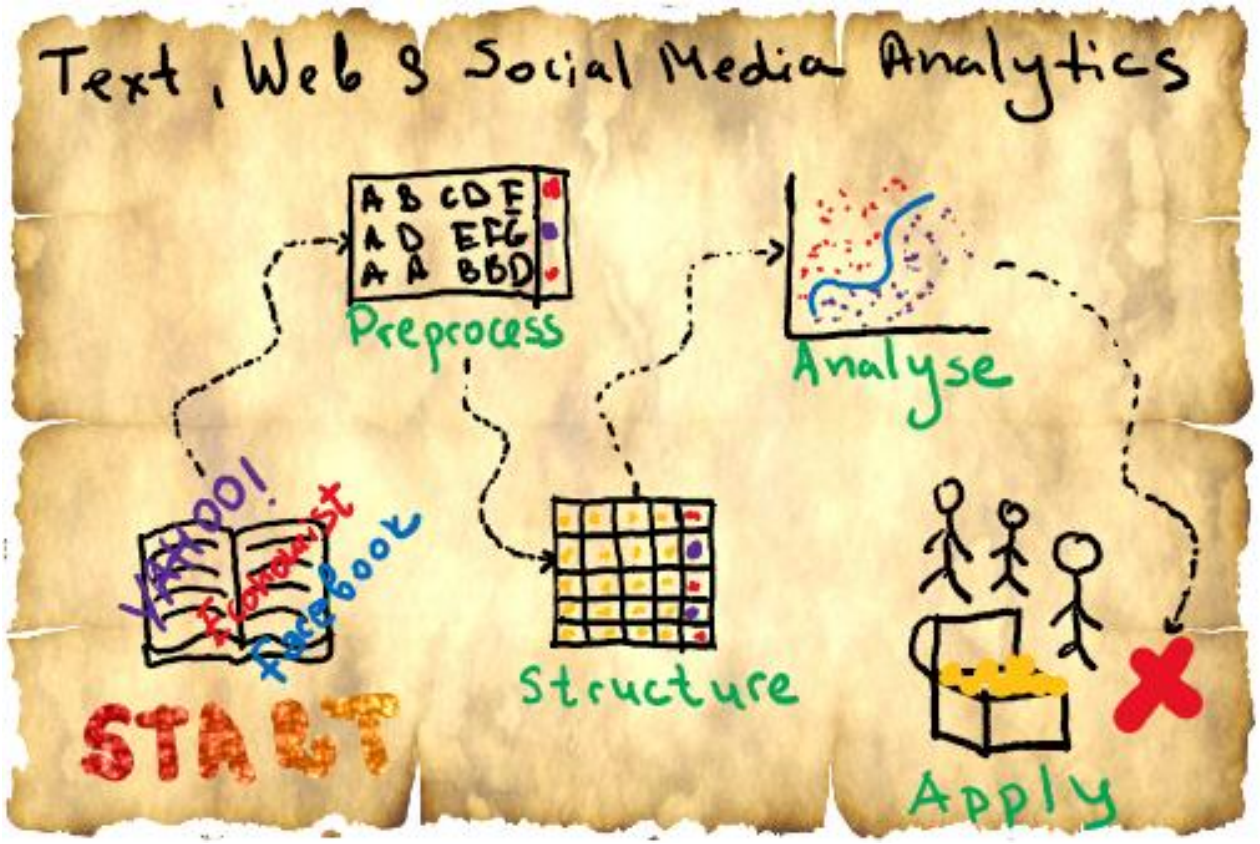
b. Finished Question 2 b. (Twitter Developer account+ Twitter App)

c. Finished Question 2 c. (Code runs)

Summary

What did we learn last week?

Course structure: Treasury map



Course structure

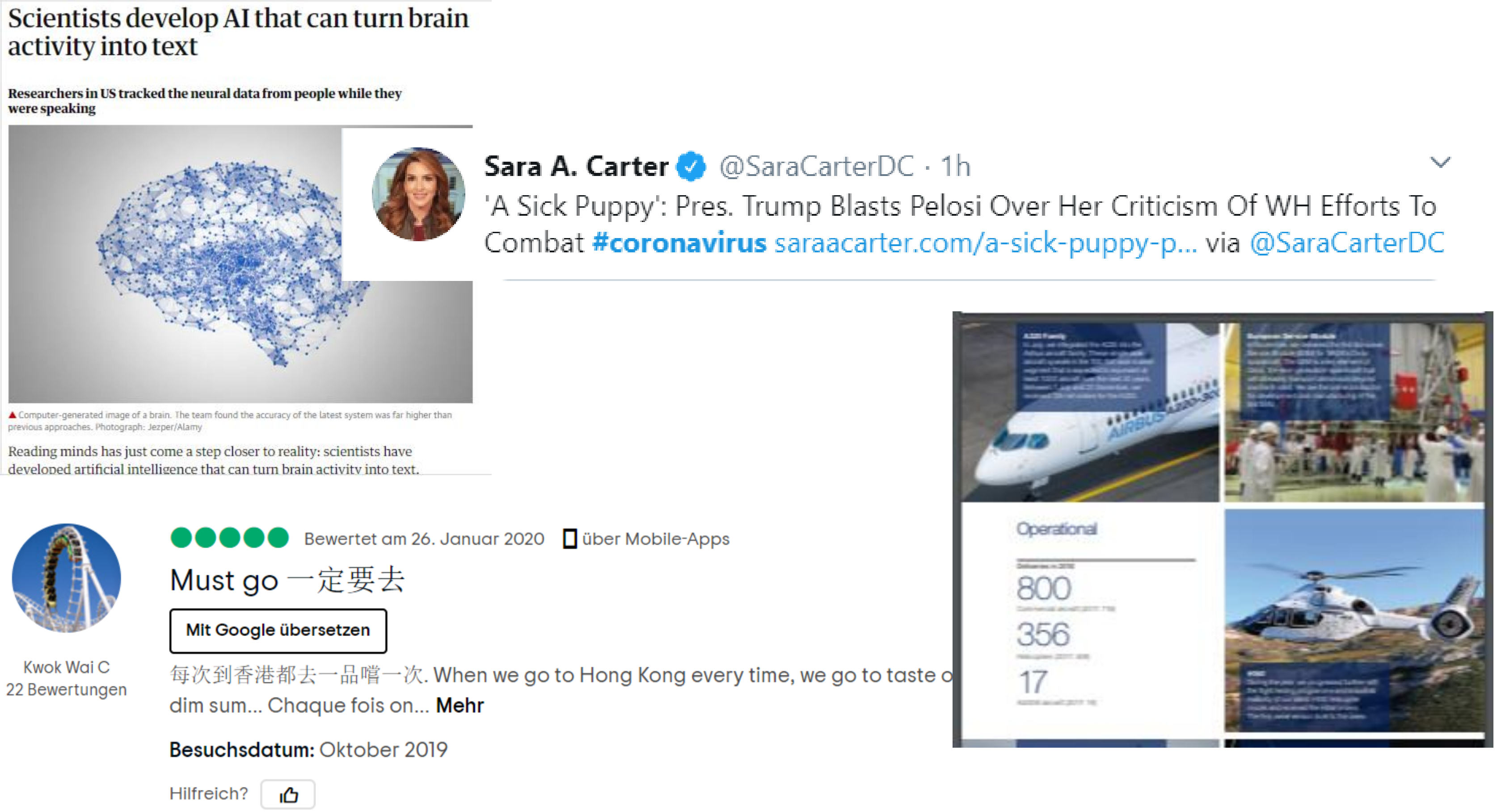
Date	Lecture	Exercise
12.04.2021	Introduction	Technical Installation
19.04.2021	Text Preprocessing	Projects kick-off
26.04.2021	Text Representation	Preprocessing Newsgroups
03.05.2021	Text Representation (2)	Text Representation Newsgroups
10.05.2021	Text Classification	Text Representation Newsgroups (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?

At the end of this lecture, you will:

1. Know why you should preprocess your text before deriving structured representation from it
2. Be able to apply and understand the aim of the following preprocessing steps:
   - Removing non-alphabetical parts such as numbers and punctuation
   - Transforming letters to lower-case ones
   - Removing non-informative words
   - Stemming and lemmatization

Motivation: Why do we need text preprocessing?



Scientists develop AI that can turn brain activity into text

Researchers in US tracked the neural data from people while they were speaking

▲ Computer-generated image of a brain. The team found the accuracy of the latest system was far higher than previous approaches. Photograph: Jezper/Alamy

Reading minds has just come a step closer to reality: scientists have developed artificial intelligence that can turn brain activity into text.

Sara A. Carter @SaraCarterDC · 1h
'A Sick Puppy': Pres. Trump Blasts Pelosi Over Her Criticism Of WH Efforts To Combat **#coronavirus** saraacarter.com/a-sick-puppy-p... via @SaraCarterDC

●●●●● Bewertet am 26. Januar 2020 über Mobile-Apps

Must go 一定要去

Mit Google übersetzen

每次到香港都去一品嚐一次. When we go to Hong Kong every time, we go to taste o dim sum... Chaque fois on... **Mehr**

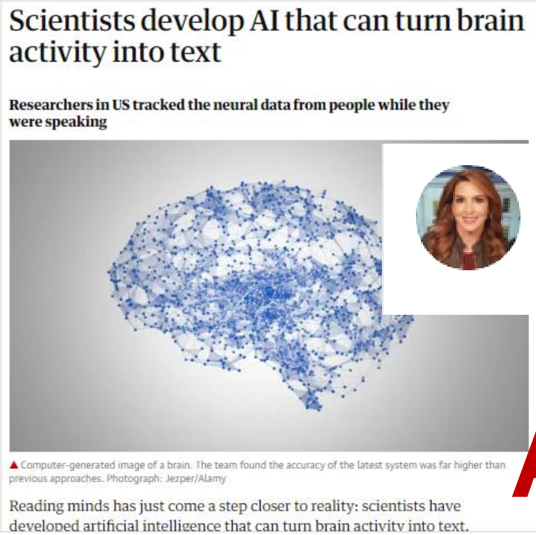
Besuchsdatum: Oktober 2019

Hilfreich?

Kwok Wai C
22 Bewertungen

Sources: www.airbus.com › corporate-topics › financial-and-company-information
https://www.theguardian.com/science/2020/mar/30/scientists-develop-ai-that-can-turn-brain-activity-into-text

Motivation: Why do we need text preprocessing?

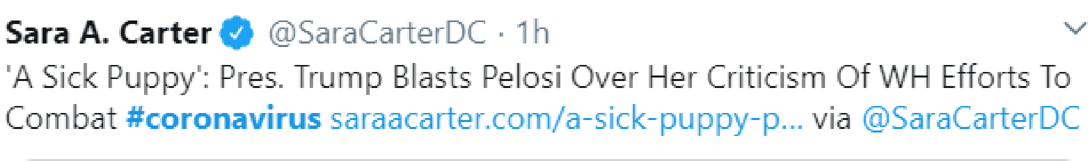


Scientists develop AI that can turn brain activity into text

Researchers in US tracked the neural data from people while they were speaking

▲ Computer-generated image of a brain. The team found the accuracy of the latest system was far higher than previous approaches. Photograph: Jezper/Alamy

Reading minds has just come a step closer to reality: scientists have developed artificial intelligence that can turn brain activity into text.



Sara A. Carter @SaraCarterDC · 1h

'A Sick Puppy': Pres. Trump Blasts Pelosi Over Her Criticism Of WH Efforts To Combat **#coronavirus** saraacarter.com/a-sick-puppy-p... via @SaraCarterDC

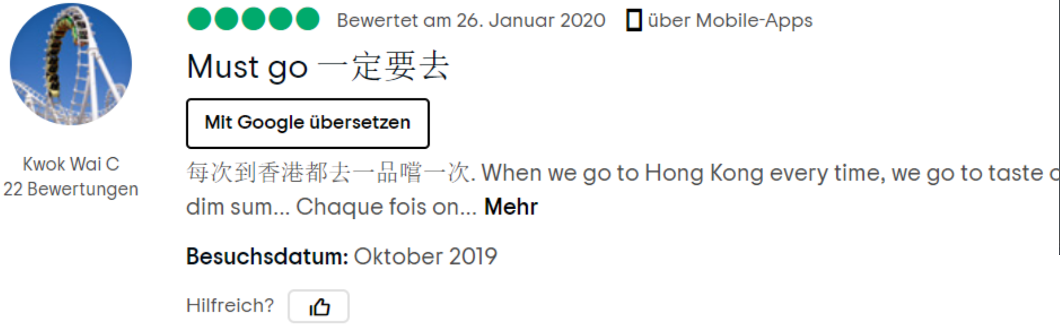
Any ideas?

Operational

800

356

17



●●●●● Bewertet am 26. Januar 2020 über Mobile-Apps

Kwok Wai C
22 Bewertungen

Must go 一定要去

Mit Google übersetzen

每次到香港都去一品嚐一次. When we go to Hong Kong every time, we go to taste c dim sum... Chaque fois on... **Mehr**

Besuchsdatum: Oktober 2019

Hilfreich?

Sources: www.airbus.com › corporate-topics › financial-and-company-information
https://www.theguardian.com/science/2020/mar/30/scientists-develop-ai-that-can-turn-brain-activity-into-text

Motivation: Why do we need text preprocessing? (2)

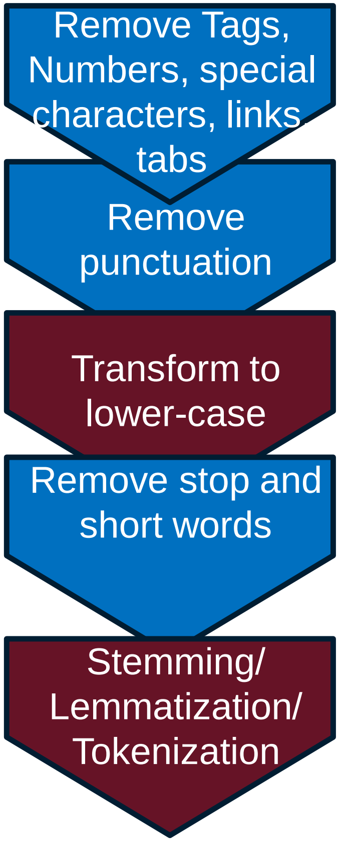
- Just as every data, text data contains a lot of noise:
  - ✓ Texts may contain non-linguistic parts such as pictures, emojis, html-tags, links
  - ✓ Even numbers can add noise, if not relevant to the task at hand.
  - ✓ Texts contain a lot of words that make them more readable, but are not informative (e.g., ‘a’, ‘the’).

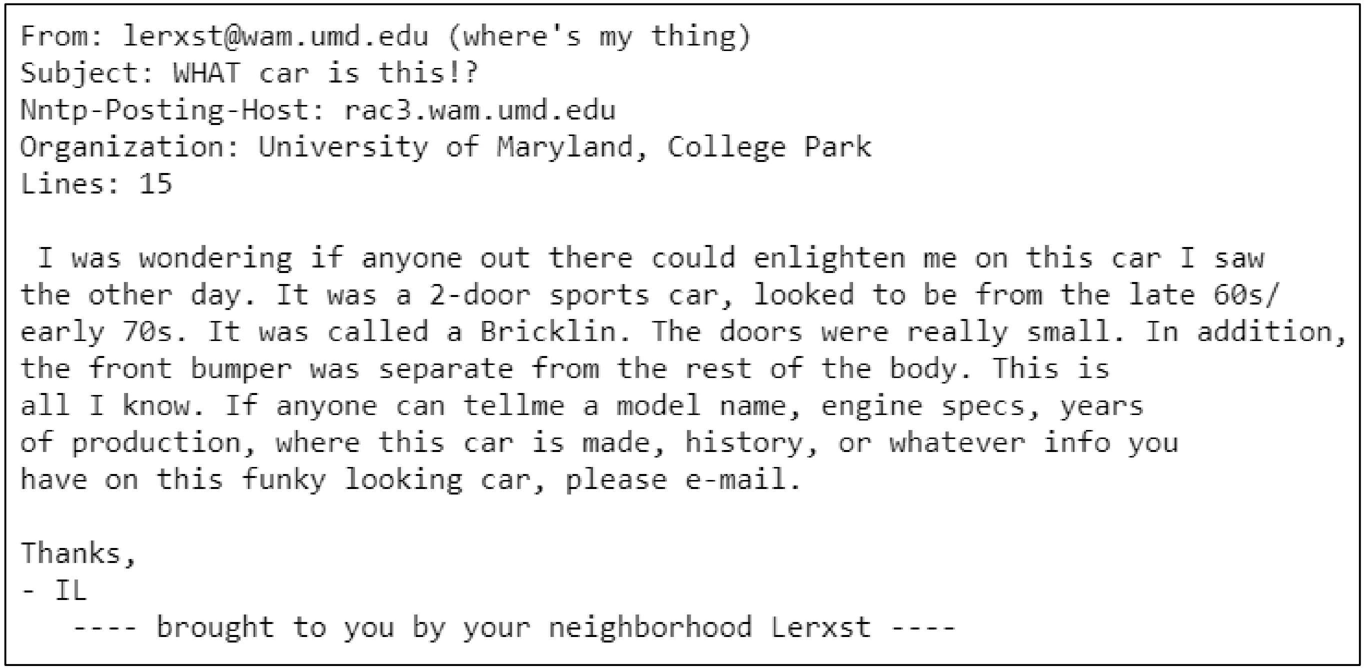
  ➔ clean data to avoid the “Garbage-in-garbage-out.” effect.
- Also, many words have a same (or related) meaning, but a machine cannot identify that
  - ✓ Case-sensitivity i.e. ‘happy’ vs. ‘Happy’ vs. ‘HAPPY’
  - ✓ A machine doesn’t know that ‘risk’ and ‘risks’ have the same linguistic root

  ➔ normalise data to avoid huge feature space without explanatory power.

▶ Text preprocessing (cleaning and normalisation)

Text preprocessing: Overview


Remove Tags, Numbers, special characters, links, tabs
Remove punctuation
Transform to lower-case
Remove stop and short words
Stemming/ Lemmatization/ Tokenization



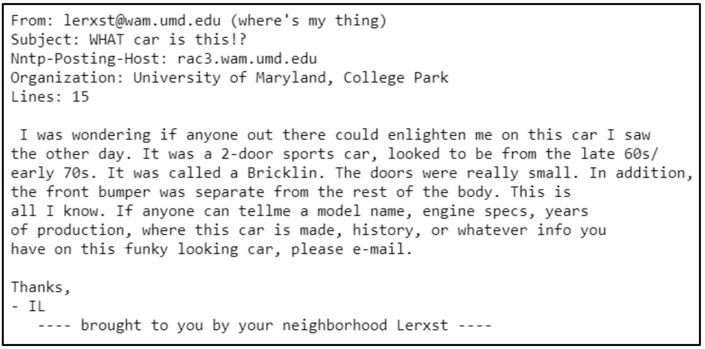
```
From: lerxst@wam.umd.edu (where's my thing)
Subject: WHAT car is this!?
Nntp-Posting-Host: rac3.wam.umd.edu
Organization: University of Maryland, College Park
Lines: 15

 I was wondering if anyone out there could enlighten me on this car I saw
the other day. It was a 2-door sports car, looked to be from the late 60s/
early 70s. It was called a Bricklin. The doors were really small. In addition,
the front bumper was separate from the rest of the body. This is
all I know. If anyone can tellme a model name, engine specs, years
of production, where this car is made, history, or whatever info you
have on this funky looking car, please e-mail.

Thanks,
- IL
   ---- brought to you by your neighborhood Lerxst ----
```

http://qwone.com/~jason/20Newsgroups/

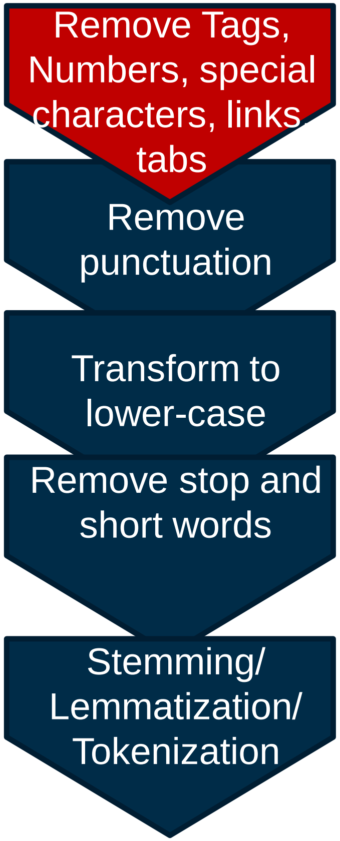
Text preprocessing: Remove non-linguistic parts



```
From: lerxst@wam.umd.edu (where's my thing)
Subject: WHAT car is this!?
Nntp-Posting-Host: rac3.wam.umd.edu
Organization: University of Maryland, College Park
Lines: 15

 I was wondering if anyone out there could enlighten me on this car I saw
the other day. It was a 2-door sports car, looked to be from the late 60s/
early 70s. It was called a Bricklin. The doors were really small. In addition,
the front bumper was separate from the rest of the body. This is
all I know. If anyone can tellme a model name, engine specs, years
of production, where this car is made, history, or whatever info you
have on this funky looking car, please e-mail.

Thanks,
- IL
   ---- brought to you by your neighborhood Lerxst ----
```

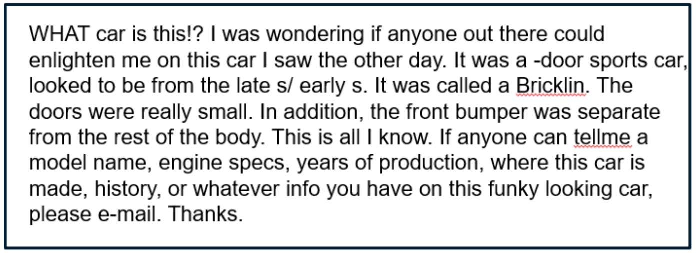

Remove Tags, Numbers, special characters, links, tabs
Remove punctuation
Transform to lower-case
Remove stop and short words
Stemming/ Lemmatization/ Tokenization

Why are we doing this?

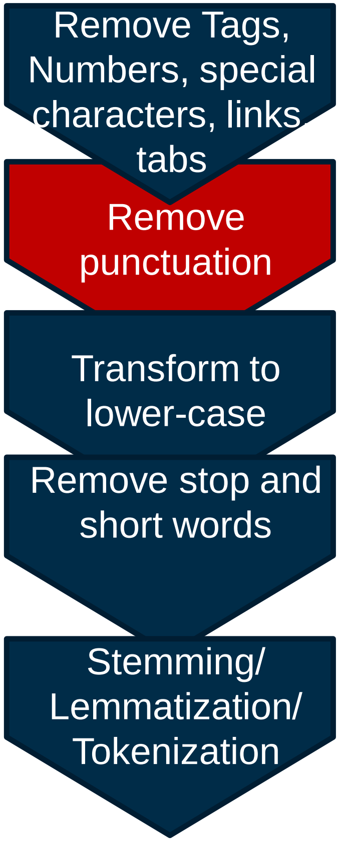
- Tags, tabs/ new lines, links (e-mails) and special characters rarely convey information relevant for machine learning analysis.
- In most cases, text mining focuses on textual information, numbers are usually already available in structured form.

WHAT car is this!? I was wondering if anyone out there could enlighten me on this car I saw the other day. It was a -door sports car, looked to be from the late s/ early s. It was called a Bricklin. The doors were really small. In addition, the front bumper was separate from the rest of the body. This is all I know. If anyone can tellme a model name, engine specs, years of production, where this car is made, history, or whatever info you have on this funky looking car, please e-mail. Thanks.

Text preprocessing: Remove punctuation



WHAT car is this!? I was wondering if anyone out there could enlighten me on this car I saw the other day. It was a -door sports car, looked to be from the late s/ early s. It was called a Bricklin. The doors were really small. In addition, the front bumper was separate from the rest of the body. This is all I know. If anyone can tellme a model name, engine specs, years of production, where this car is made, history, or whatever info you have on this funky looking car, please e-mail. Thanks.

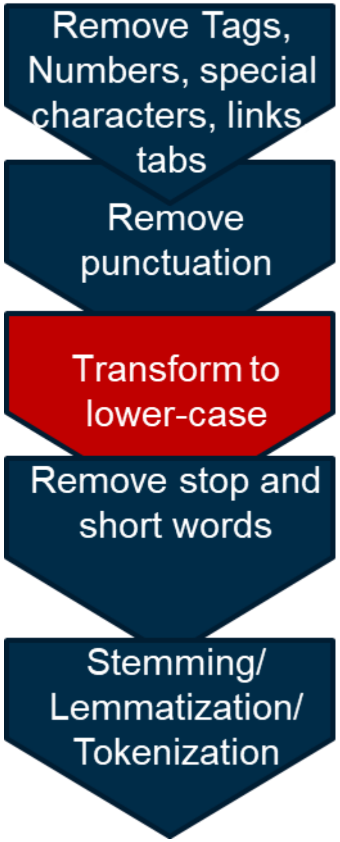

Remove Tags, Numbers, special characters, links, tabs
Remove punctuation
Transform to lower-case
Remove stop and short words
Stemming/ Lemmatization/ Tokenization

Why are we doing this?

- Punctuation improves the readability of the text, but often does not convey relevant information for text analysis.
- Still, this is an optional step, might be omitted if you are interested in the sentence structure.

WHAT car is this I was wondering if anyone out there could enlighten me on this car I saw the other day It was a door sports car looked to be from the late s early s It was called a Bricklin The doors were really small In addition the front bumper was separate from the rest of the body This is all I know If anyone can tellme a model name engine specs years of production where this car is made history or whatever info you have on this funky looking car please e mail Thanks

Text preprocessing: Transform to lower-case



Remove Tags, Numbers, special characters, links tabs

Remove punctuation

Transform to lower-case

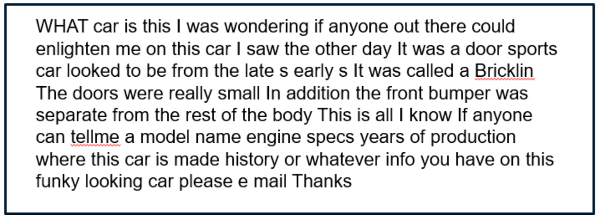
Remove stop and short words

Stemming/ Lemmatization/ Tokenization

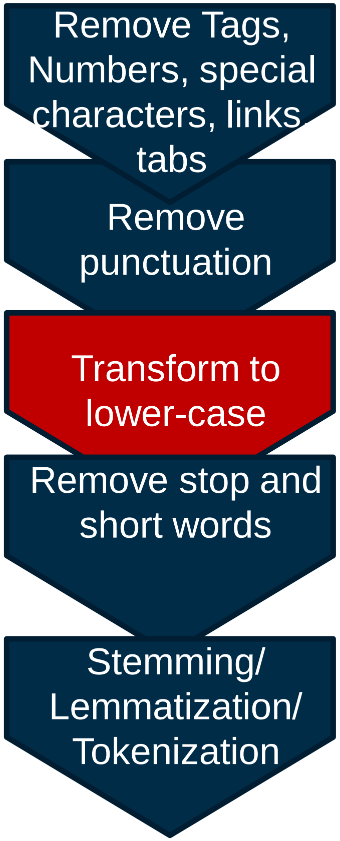
Why should words be transformed to lower-case?

Text preprocessing: Transform to lower-case

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



WHAT car is this I was wondering if anyone out there could enlighten me on this car I saw the other day It was a door sports car looked to be from the late s early s It was called a Bricklin The doors were really small In addition the front bumper was separate from the rest of the body This is all I know If anyone can tellme a model name engine specs years of production where this car is made history or whatever info you have on this funky looking car please e mail Thanks



Remove Tags, Numbers, special characters, links, tabs
Remove punctuation
Transform to lower-case
Remove stop and short words
Stemming/ Lemmatization/ Tokenization

Why are we doing this?

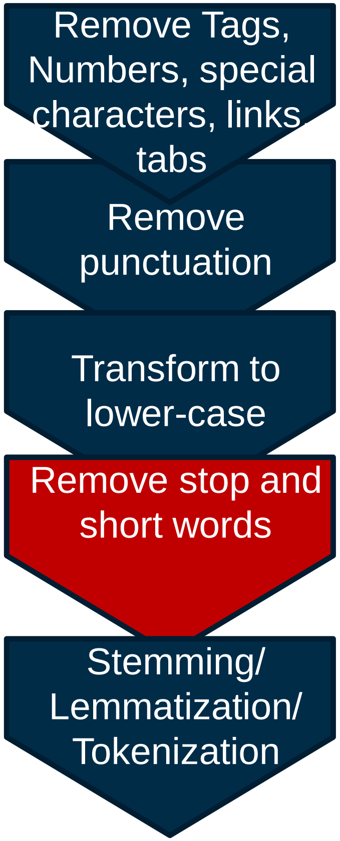
- We as humans know that ‘What’, ‘WHAT’ and ‘what’ mean the same.
- For a machine those are three different words

➔ Tell the machine they are the same by making everything lower case (normalisation).

what car is this i was wondering if anyone out there could enlighten me on this car i saw the other day it was a door sports car looked to be from the late s early s it was called a bricklin the doors were really small in addition the front bumper was separate from the rest of the body this is all i know if anyone can tellme a model name engine specs years of production where this car is made history or whatever info you have on this funky looking car please e mail thanks

Text preprocessing: Remove stopwords and short words

what car is this i was wondering if anyone out there could enlighten me on this car i saw the other day it was a door sports car looked to be from the late s early s it was called a bricklin the doors were really small in addition the front bumper was separate from the rest of the body this is all i know if anyone can tellme a model name engine specs years of production where this car is made history or whatever info you have on this funky looking car please e mail thanks


Remove Tags, Numbers, special characters, links, tabs
Remove punctuation
Transform to lower-case
Remove stop and short words
Stemming/ Lemmatization/ Tokenization

Why are we doing this?

- Stopwords: a, the, which, to, himself
- They increase readability, but do not substantially add information

➔ Noise in the data

After stopwords removal

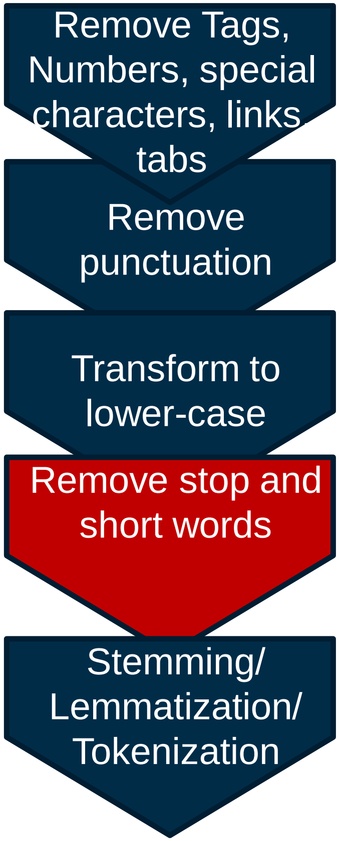
car wondering enlighten car saw day door sports car looked late s early s called bricklin doors small addition bumper separate rest body know tellme model engine specs years production car history info funky looking car e mail thanks

Text preprocessing: Overview

Is preprocessing language-specific?

Text preprocessing: Remove stopwords and short words (2)

car wondering enlighten car saw day door sports car looked late s early s called bricklin doors small addition bumper separate rest body know tellme model engine specs years production car history info funky looking car e mail thanks


Remove Tags, Numbers, special characters, links, tabs
Remove punctuation
Transform to lower-case
Remove stop and short words
Stemming/ Lemmatization/ Tokenization

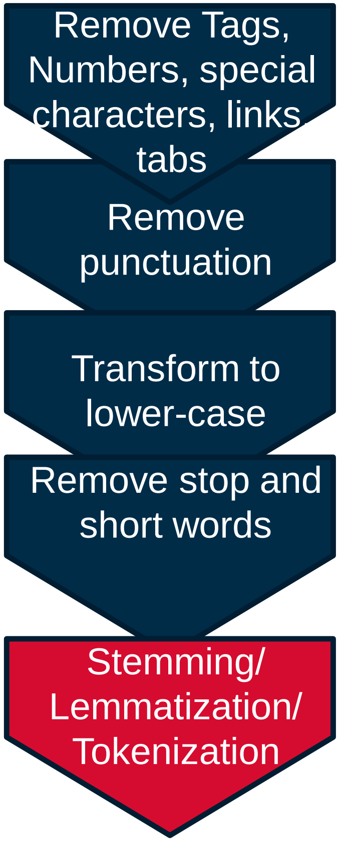
Why are we doing this?

- Short words: words with a few characters e.g., s, co
- They usually do not convey important information and either were part of the text or resulted from the previous preprocessing steps (e.g., often after stemming).

After short words removal (< 3 characters)

car wondering enlighten car saw day door sports car looked late early called bricklin doors small addition bumper separate rest body know tellme model engine specs years production car history info funky looking car mail thanks

Text preprocessing: Stemming and Lemmatization


Remove Tags, Numbers, special characters, links, tabs
Remove punctuation
Transform to lower-case
Remove stop and short words
Stemming/ Lemmatization/ Tokenization

Why are we doing this?

- Words are sometimes in plural (e.g. results) or in past tense (e.g. wanted).
- We know that they have similar meaning, but a machine doesn’t.

➔ Try to convert all words to their root (normalisation, language specific).

Stemming: convert the word to a root form following a set of slicing rules (e.g. gardening ➔ garden)

Lemmatization: determine the linguistic root of the word, based on look-up dictionaries (e.g. went ➔ go)

Text preprocessing: Transform to lower-case

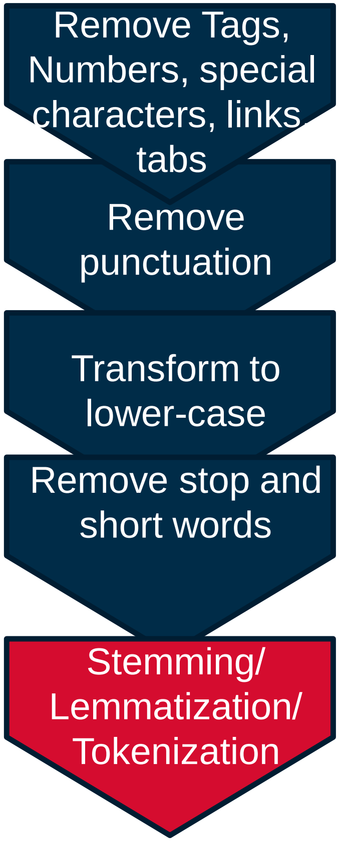
What are the *advantages* of stemming as opposed to lemmatization?

a. **Linguistic root**

b. **Quick and simple calculation**

c. **No real word as output**

Text preprocessing: Stemming and Lemmatization



Remove Tags, Numbers, special characters, links, tabs

Remove punctuation

Transform to lower-case

Remove stop and short words

Stemming/ Lemmatization/ Tokenization

Why are we doing this?

- Words are sometimes in plural (e.g. results) or in past tense (e.g. wanted).
- We know that they have similar meaning, but a machine doesn’t.

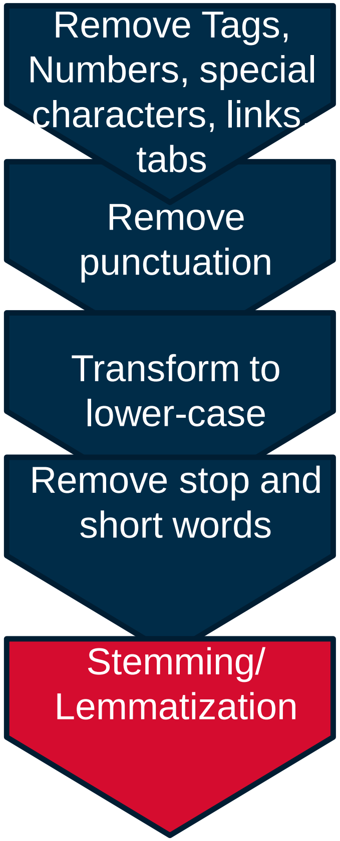
➔ Try to convert all words to their root (language-specific).

Stemming: convert the word to a root form following a set of slicing rules (e.g. gardening ➔ garden)
+ Quick and simple
- Resulting word may not exist

Lemmatization: determine the linguistic root of the word, based on look-up dictionaries (e.g. went ➔ go)
+ True root
- Slower, less words for grouping

Text preprocessing: Stemming and Lemmatization (2)

car wondering enlighten car saw day door sports car looked late early called bricklin doors small addition bumper separate rest body know tellme model engine specs years production car history info funky looking car mail thanks


Remove Tags, Numbers, special characters, links, tabs

Remove punctuation

Transform to lower-case

Remove stop and short words

Stemming/ Lemmatization

After Stemming

car wonder enlighten car **saw dai** door sport car look late earli call bricklin door small addit bumper separ rest bodi know tellm model engin spec year product car histori info funki look car mail thank

After Lemmatization

car 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

Stemming: Porter Stemmer

- The **Porter stemmer** is one of the most commonly applied stemming algorithms.
- It consecutively removes **suffixes** from the words, following certain rules:
- **Definitions:**
  - ✓ A consonant is a letter other than A, E, I, O or U and other than Y preceded by a consonant. All other letters are vowels.
  - ✓ $*v*$ and $*c*$ mean that the word contains a vowel and consonant respectively.
  - ✓ m is the preceding string length.
- Phase 1 a: (Plurals)

Rule	Example
$SSES \rightarrow SS$	$caresses \rightarrow caress$
$IES \rightarrow I$	$ponies \rightarrow poni$
$SS \rightarrow SS$	$caress \rightarrow caress$
$S \rightarrow$	$cats \rightarrow cat$

Stemming: Porter Stemmer (2)

- **Definitions:** $*v*$ and $*c*$ mean that the word contains a vowel and consonant respectively. m is the preceding string length.

Phase 1 b (verbs):

Rule	Example
$(m > 0)EED \rightarrow EE$	$agreed \rightarrow agree$
$(*v*)ED \rightarrow$	$plastered \rightarrow plaster$
$(*v*)ING \rightarrow$	$motoring \rightarrow motor$

Phase 1 c (adverbs):

Rule	Example
$(*v*)Y \rightarrow I$	$happy \rightarrow happi$

Stemming: Porter Stemmer (3)

Phase 2 (adjectives and nouns):

Rule	Example
$(m > 0)ATIONAL \rightarrow ATE$	$relational \rightarrow relate$
$(m > 0)TIONAL \rightarrow TION$	$conditional \rightarrow condition$
$(m > 0)IVENESS \rightarrow IVE$	$decisiveness \rightarrow decisive$
…	…

Check the whole algorithm here:

http://snowball.tartarus.org/algorithms/porter/stemmer.html
http://snowball.tartarus.org/algorithms/porter/stem_ISO_8859_1.sbl

Text preprocessing: What is a corpus?

- A **corpus** is a collection of similar documents relevant for the task at hand i.e. a normal dataset consisting of texts as data points.
- **Examples:** a collection of news on a given topic, tweets, e-mails, call centre protocols, annual reports.
- A corpus can be annotated i.e. consist of **labelled** data (e.g. news category for each news) or unannotated i.e. consist of **unlabelled** data (e.g. tweets).
- Usually annotated corpora are used in supervised learning and unannotated ones are used in unsupervised learning.
- **IMPPORTANT:** always store your corpus when preprocessing is done to avoid doing this costly process over and over again.

Summary and Outlook

Summary:

- Text data may contain a lot of noise making analysis difficult.
- Text preprocessing is done by:
  1. Removing non-linguistic parts, short and stop words
  2. Transforming texts to lower case
  3. Stemming and lemmatization
- The Porter Stemmer is one of the most commonly applied stemming algorithms.
- Datasets for text analytics are usually called corpus. They should always be stored after preprocessing to reduce analysis effort.
- **Outlook:** The preprocessed corpus is transformed in a structured form by using an appropriate document representation technique (see next lecture).

Questions?

Exercise 2

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 2 (in Moodle).