Text Mining, Transforming Text into Knowledge

Ayoub Bagheri



Today

- Introduction
- Examples
- Applications
- Regular expressions

Introduction



Ayoub Bagheri a.bagheri@uu.nl Course coordinator



Anastasia Giachanou

a.giachanou@uu.nl

Lecturer – feature selection & sentiment analysis



Timo van der Kuil

<u>t.h.vanderkuil@uu.nl</u>

Lab teacher





Course website

https://textminingcourse.nl/

202400006



Materials of Text Mining, Transforming Text into Knowledge

Text Mining, Transforming Text into Knowledge

This webpage contains all materials required for the Applied Data Science course Text Mining, Transforming Text into Knowledge (course code: 202400006).

The materials on this website are CC-BY-4.0 licensed.

 \odot

Syllabus

You can find the course syllabus as a web page here or as a pdf here. The course schedule with required reading materials is in the syllabus as well, specifically here.

Lectures

Course overview

Week Topic

- 1 Intro to text mining & regular expressions
- 2 Text preprocessing
- 3 Text classification
- 4 Feature selection
- 5 Clustering & topic modeling
- 6 Word embedding
- 7 Deep learning & LLMs
- 8 Sentiment analysis
- 9 Responsible text mining & applications

Course proceedings

- 1 lecture per week (Every Monday at 15:15)
 - Read required readings before lecture!
 - Ask questions!
- 1 lab session per week (Every Monday at 17:15)
 - Exercises during the lab session
- 2 assignments
 - 10% of your grade
 - Week 4 & week 8
- 1 **exam** (and a resit)

Readings about text mining

- Jurafsky & Martin (2024). Speech and Language Processing. <u>https://web.stanford.edu/~jurafsky/slp3/</u>
- Eisenstein (2019). Natural Language Processing. https://github.com/jacobeisenstein/gt-nlpclass/blob/master/notes/eisenstein-nlpnotes.pdf

SPEECH AND Language processing

An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition



DANIEL JURAFSKY & JAMES H. MARTIN



JACOB EISENSTEIN

Read the syllabus! A lot of your questions will be answered by reading the syllabus

Text Mining

Text mining: definition 1

• "The discovery by computer of **new, previously unknown information**, by automatically extracting information from different written resources", Hearst (1999).

Example 1: Who was the best Friend?

https://rss.onlinelibrary.wiley.com/doi/epdf/10.1111/1740-9713.01574





Text mining: definition 2

• Text mining is about looking for **patterns in text**, in a similar way that data mining can be loosely described as looking for patterns in data.

Example 2: Did a poet with donkey ears write the oldest anthem in the world?

https://dh2017.adho.org/abstracts/079/079.pdf



Text mining: definition 3

• Text mining describes a set of linguistic, statistical, and machine learning techniques that model and **structure the information content of textual sources**. (Wikipedia)

Example 3: Automatic detection of disease codes in cardiology discharge letters

https://www.nature.com/articles/s41746-021-00404-9



Box 1: An example of a Dutch discharge letter from the dataset

Bovengenoemde patiënt was opgenomen op <DATUM-1> op de <PERSOON-1> voor het specialisme Cardiologie. Reden van opname STEMI inferior Cardiale voorgeschiedenis. Blanco Cardiovasculaire risicofactoren: Roken(-) Diabetes(-) Hypertensie(?) Hypercholesterolemie (?) Anamnese. Om 18.30 pijn op de borst met uitstraling naar de linkerarm, zweten, misselijk. Ambulance gebeld en bij aansluiten monitor beeld van acuut onderwandinfarct. AMBU overdracht. 500 mg aspegic iv, ticagrelor 180 mg oraal, heparine, zofran eenmalig, 3× NTG spray. HD stabiel gebleven.Medicatie bij presentatie.Geen. Lichamelijk onderzoek. Grauw, vegetatief, Halsvenen niet gestuwd. Cor s1 s2 geen souffles.Pulm schoon. Extr warm en slank. Aanvullend onderzoek. AMBU ECG: Sinusritme, STEMI inferior III)II C/vermoedelijk RCA. Coronair angiografie. (...). Conclusie angio: 1-vatslijden..PCI Conclusie en beleid Bovengenoemde <LEEFTIJD-1> jarige man, blanco cardiale voorgeschiedenis, werd gepresenteerd vanwege een STEMI inferior waarvoor een spoed PCI werd verricht van de mid-RCA. Er bestaan geen relevante nevenletsels. Hij kon na de procedure worden overgeplaatst naar de CCU van het <INSTELLING-2>...Dank voor de snelle overname...Medicatie bij overplaatsing. Acetylsalicylzuur dispertablet 80 mg; oraal; 1× per dag 80 milligram; <DATUM-1>. Ticagrelor tablet 90 mg; oraal; 2× per dag 90 milligram; <DATUM-1>. Metoprolol tablet 50 mg; oraal; 2× per dag 25 milligram; $\langle DATUM-1 \rangle$. Atorvastatine tablet 40 mg (als ca-zout-3-water); oraal; 1× per dag 40 milliaram: <DATUM-1> Samenvatting Hoofddiagnose: STEMI inferior wv PCI RCA. Geen nevenletsels. Nevendiagnoses:

Complicaties: geen Ontslag naar: CCU <INSTELLING-2>.

ChatGPT

You What is Text Mining?

ChatGPT

Text mining, also known as text data mining or text analytics, is the process of extracting meaningful information and insights from large amounts of unstructured text data. This process involves a combination of techniques from linguistics, data mining, and computer science, particularly natural language processing (NLP). The main steps in text mining typically include:

- 1. **Text Collection**: Gathering a large corpus of text from various sources such as books, articles, websites, social media, etc.
- 2. **Preprocessing**: Cleaning and organizing the text. This step may include removing irrelevant data (like HTML tags), converting text to a uniform format, eliminating stop words (common words that don't contribute much meaning), and reducing words to their base or root form through stemming or lemmatization.

Why text mining?

- **Text data is everywhere**, websites (e.g., news), social media (e.g., twitter), databases (e.g., doctors' notes), digital scans of printed materials, ...
- A lot of world's data is in **unstructured text format**
- Text is high dimensional
- Applications in industry: search, machine translation, sentiment analysis, question answering, ...
- Applications in science: cognitive modeling, understanding bias in language, automated systematic literature reviews, ...

High dimensional data

• All possible words & phrases

Complex & subtle relationships in text

- "Jumbo merges with Hema"
- "Jumbo is bought by Hema"

Ambiguity & context sensitivity

- car = automobile = vehicle
- kapsalon (hairdresser) or kapsalon (fast food)

Homographs: same words can mean different things

• Bat (sports, animal, ...)

Synonyms

Misspellings

Abbreviations

Negations

Spelling variations

LANGUAGE!

What to do with the TM challenges? We will answer this question during the course.



Text mining process

- Data: Text
- **Text Preprocessing:** is the process of cleaning, normalizing, and structuring raw text data into a format suitable for analysis or input into NLP models. (week 2)
- **Text transformation, feature generation:** involves converting text data into a different format or structure, such as numerical vectors or simplified forms, to make it suitable for analysis or modeling. (weeks 1, 2, 3, 6, 7, 8)
- **Feature selection:** is the process of identifying and selecting the most relevant features from a dataset to improve model performance and reduce complexity. (week 4)
- **Data mining, pattern discovery:** is the process of extracting meaningful patterns and knowledge from text. (weeks 3, 5, 7, 8, 9)
- Interpretation / Evaluation: is the process of understanding and explaining the model and patterns / is the assessment process to measure performance and quality. (weeks 3-9)

Many tasks

- Feature selection
- Text classification
- Text clustering
- Topic modelling
- Sentiment analysis
- Responsible text mining
- Bias detection
- Text summarization
- Translation
- Automatic reviewing: ASReview
- Chatbot design (conversational AI)
- And more..



Regular Expressions You can design a chatbot at the end of the course, but we need to start with the basics!



http://xkcd.com/208/

Regular expressions (regex)

- **Powerful** and a **very useful** tool for text (pre)processing
- Used in pretty much every pipeline involving text
- Typical applications:
 - Extracting numbers, emails, IP-addresses, etc.
 - Validating text inputs in GUIs
 - Reformatting annoying incorrect dates (everything not yyyy-mm-dd)
 - Scrubbing names and addresses for pseudonimization
- Cryptic & takes a lot of practice!

Basic matching using the 're' library

• Find all occurrences of a pattern in the string.

```
text = "cat, bat, rat, mat"
matches = re.findall(r"\b[a-z]at\b", text)
print("Matches:", matches) # Output: ['cat', 'bat', 'rat', 'mat']
```

Basic matching using the 're' library

• Replace a pattern with a new string.

```
text = "The rain in Spain"
result = re.sub(r"ain", "123", text)
print("Result:", result) # Output: The r123 in Sp123
```

Basic matching using the 're' library

Anchors: Start and End of String

- By default, regular expressions will match any part of a string. It's often useful to **anchor** the regular expression so that it matches from the start or end of the string. You can use:
 - ^ matches the start of the string.
 - \$ matches the end of the string

```
text = "Hello, world!"
print(bool(re.search(r"^Hello", text))) # Output: True
print(bool(re.search(r"world!$", text))) # Output: True
```

How to write regular expressions

1. Simple regex searches

RE	Example Patterns Matched
/woodchucks/	"interesting links to woodchucks and lemurs"
/a/	"Mary Ann stopped by Mona's"
/!/	"You've left the burglar behind again!" said Nori

2. Disjunction (OR)

The use of the brackets [] to specify a disjunction of characters:

RE	Match	Example Patterns
/[wW]oodchuck/	Woodchuck or woodchuck	"Woodchuck"
/[abc]/	'a', 'b', <i>or</i> 'c'	"In uomini, in sold <u>a</u> ti"
/[1234567890]/	any digit	"plenty of <u>7</u> to 5"

The pipe sympol | is also for disjunction:

Pattern	Matches
groundhog woodchuck	
yours mine	yours mine
abc	= [abc]

3. Brackets and dash

The use of the brackets [] plus the dash - to specify a range:

RE	Match	Example Patterns Matched
/[A-Z]/	an upper case letter	"we should call it 'Drenched Blossoms' "
/[a-z]/	a lower case letter	"my beans were impatient to be hoed!"
/[0-9]/	a single digit	"Chapter 1: Down the Rabbit Hole"

4. Negation

The caret ^ for negation or just to mean ^:

RE	Match (single characters)	Example Patterns Matched
/[^A-Z]/	not an upper case letter	"Oyfn pripetchik"
/[^Ss]/	neither 'S' nor 's'	"I have no exquisite reason for't"
/[^.]/	not a period	"our resident Djinn"
/[e^]/	either 'e' or '^'	"look up _ now"
/a^b/	the pattern 'a b'	"look up <u>a^ b</u> now"

5. Question and period marks

The question mark ? marks optionality of the previous expression:

RE	Match	Example Patterns Matched
/woodchucks?/	woodchuck or woodchucks	"woodchuck"
/colou?r/	color or colour	" <u>color</u> "

The use of the period . to specify any character:

RE	Match	Example Matches
/beg.n/	any character between beg and n	begin, beg'n, begun

6. Anchors

RE	Match
^	start of line
\\$	end of line
∖b	word boundary
∖B	non-word boundary

7. Common sets

Aliases for common sets of characters:

RE	Expansion	Match	First Matches
\d	[0-9]	any digit	Party_of_5
\D	[^0-9]	any non-digit	Blue_moon
\w	[a-zA-Z0-9_]	any alphanumeric/underscore	<u>D</u> aiyu
$\backslash W$	[^\w]	a non-alphanumeric	<u>1</u> !!!
\s	$[_{r}t n f]$	whitespace (space, tab)	
\S	[^\s]	Non-whitespace	in_Concord

The backslash for escaping!

8. Operators for counting

Aliases for common sets of characters:

RE	Expansion	Match	First Matches
\d	[0-9]	any digit	Party_of_5
\D	[^0-9]	any non-digit	Blue_moon
\w	[a-zA-Z0-9_]	any alphanumeric/underscore	<u>D</u> aiyu
\mathbb{W}	[^\w]	a non-alphanumeric	<u>1</u> !!!
\s	[_\r\t\n\f]	whitespace (space, tab)	
\S	[^\s]	Non-whitespace	<u>i</u> n_Concord

The backslash for escaping!

9. Other

Some characters that need to be backslashed:

RE	Match	First Patterns Matched
/*	an asterisk "*"	"K <u>*</u> A*P*L*A*N"
١.	a period "."	"Dr. Livingston, I presume"
\?	a question mark	"Why don't they come and lend a hand?"
\n	a newline	
\t	a tab	

10. Operator precedence

Parenthesis () Counters * + ? {} Sequences and anchors the ^my end\$ Disjunction |

Examples

Regex

hello
gray|grey
gr(a|e)y
gr[ae]y
b[aeiou]bble
[b-chm-pP]at|ot

Matches any string that contains {hello} contains {gray, grey} contains {gray, grey} contains {gray, grey} contains {babble, bebble, bibble, bobble, bubble} contains {bat, cat, hat, mat, nat, oat, pat, Pat, ot}

colou?r

contains {color, colour}

More complicated examples

Regex

\d 1\d{10} \d+(\.\d\d)?

^dog dog\$ ^dog\$ Matches any string that contains {0,1,2,3,4,5,6,7,8,9} contains an 11-digit string starting with a 1 contains a positive integer or a floating point number with exactly two characters after the decimal point. begins with "dog" ends with "dog" is exactly "dog"

Regular expression conclusion

- You have now *heard of* regular expressions
- And might have a basic idea of what you might do with them
- The only way to really learn, however, is practice
- Next time you encounter some text you need to work on think "can I do this using regular expressions?"
- The answer is probably "yes".

Challenge problem: regex crossword.com/



Conclusion

- Text is everywhere!
- Text Mining and NLP is the topic of the day!
- Regular expression is a good start.

Practical 1: Read text data and perform

regular expression. Using a text dataset for online reviews.

