UNIVERSITY OF AMSTERDAM
Economics & Business

Data Science Minor – 2020/2021

Text Retrieval - Text Mining
Week 1 : Lexical Representations

Julien ROSSI

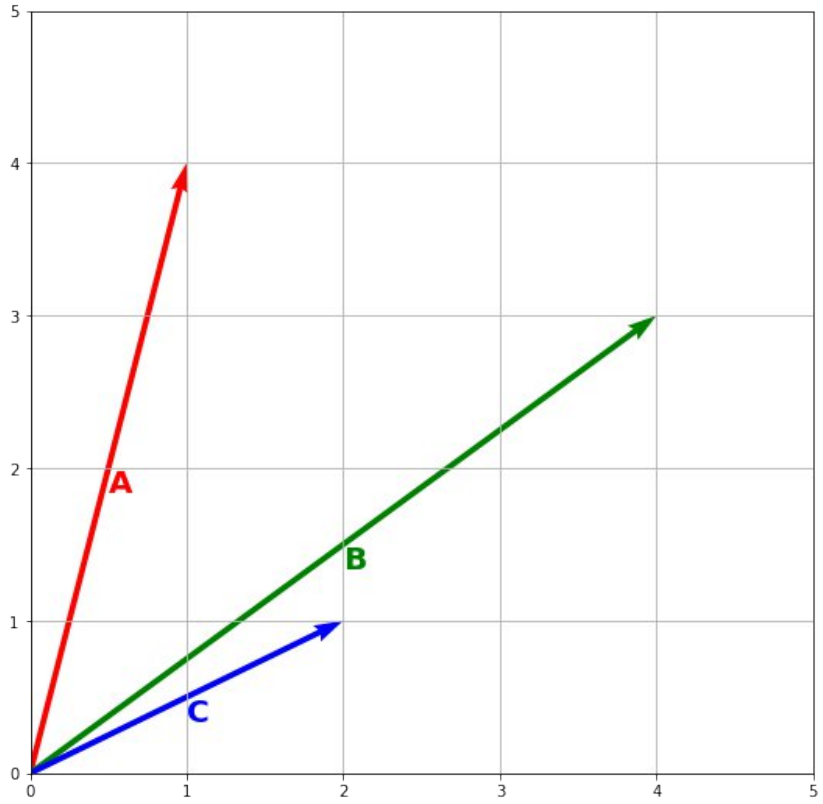
Lexical Representations

- Understand the **Bag of Words (BoW)**
- Understand **cosine similarity**, its impact on **BoW**
- Understand the problems behind **tokenizing, stemming, stopping**
- Know the limitations behind lexical representations

- In a nutshell:
  - Two texts are similar if they share the same words
  - Focus on **how** it is written, rather than **what** it is about

Cosine Similarity

- See the Notebook “cosine similarity”



Tokenization

- Split a text in bits and pieces called tokens
- “I love Amsterdam” → “I” “love” “Amsterdam”

- Token ≠ Word
  - “I was there (in Moscow)”
  - → | I | was | there | (| in | Moscow |) |

- This is the first processing applied to any text

Bag of Words

1. Observe all unique tokens appearing in texts:
   - This is the **Vocabulary** or **Dictionary**
   - Its size is **V**, the number of tokens
2. For each document
   - Create a vector of dimension **V**
   - In position 1, write the number of times the token at position 1 in the dictionary appears in the document
   - Continue for each position in the vector
3. This vector of dimension **V** is the **Bag of Words** representation of the document

Bag of Words

- Sentence 1: ”the cat sat on the hat"
- Sentence 2: ”the dog ate the cat and the hat”

- Vocabulary: and, ate, cat, dog, hat, on, sat, the

- BoW 1: [0, 0, 1, 0, 1, 1, 1, 2]
- BoW 2: [1, 1, 1, 1, 1, 0, 0, 3]

- At first position:
  - “and” does not appear in sentence 1
  - “and” appears one time in sentence 2

TF-IDF

- Raw counts over-exaggerate the importance of words
- Raw counts only focus on one document

- Consider a document in a corpus
- TF is Term Frequency: the number of times the term appears in the document
- DF is Document Frequency: the number of times the term appears in the whole corpus
- IDF is an inverse of DF

- TF-IDF(term, document, corpus) = TF * IDF

TF-IDF

- TF-IDF(term, document, corpus) = TF * IDF
- **High Value:** term that appears in the document, but not a lot overall in the corpus
- **Low Value:** term that appears in the document, but also in a lot of other documents in the corpus

TF-IDF

- Many Formulas

- Pick one per column to compose a TF-IDF coefficient
- Sklearn default formula:

  natural * idf * no

Term frequency		Document frequency		Normalization	
natural	$\mathrm{tf}(t;d)$				
logarithm	$1 + \log \mathrm{tf}(t;d)$	no	1		
augmented	$\frac{1}{2} + \frac{1}{2}\frac{\mathrm{tf}(t;d)}{\max\{t':\mathrm{tf}(t';d)\}}$	idf	$\log \frac{n}{\mathrm{df}(t)}$	no	1
boolean	$\begin{cases} 1 & \text{if } \mathrm{tf}(t;d) > 0 \\ 0 & \text{otherwise} \end{cases}$	prob idf	$\max\{0, \log \frac{n}{n-\mathrm{df}(t)}\}$	eucl	$\frac{1}{\sqrt{\sum_{t \in d} \mathrm{w}(t,d)}}$
log avg	$\frac{1+\log \mathrm{tf}(t;d)}{1+\log avg_{t' \in d}\mathrm{tf}(t';d)}$				

UNIVERSITY OF AMSTERDAM
Economics & Business

Bag of Words / TF-IDF

- See the Notebook “Bag of Words”

```
Text: "To Sherlock Holmes she is always the woman."
BoW Shape: (1, 127)
BoW Vector: [[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0
  1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0]]
```

```
abhorrent     : 0 | actions       : 0 | adjusted      : 0 | adler         : 0 | admirable     : 0 | admirably     : 0 | admit         : 0 | akin          : 0
all           : 0 | always        : 1 | and           : 0 | any           : 0 | as            : 0 | balanced      : 0 | be            : 0 | but           : 0
cold          : 0 | crack         : 0 | delicate      : 0 | distracting   : 0 | disturbing    : 0 | doubt         : 0 | drawing       : 0 | dubious       : 0
eclipses      : 0 | emotion       : 0 | emotions      : 0 | excellent     : 0 | eyes          : 0 | factor        : 0 | false         : 0 | felt          : 0
finely        : 0 | for           : 0 | from          : 0 | gibe          : 0 | grit          : 0 | has           : 0 | have          : 0 | he            : 0
heard         : 0 | her           : 0 | high          : 0 | him           : 0 | himself       : 0 | his           : 0 | holmes        : 1 | in            : 0
instrument    : 0 | into          : 0 | introduce     : 0 | intrusions    : 0 | irene         : 0 | is            : 1 | it            : 0 | late          : 0
lenses        : 0 | love          : 0 | lover         : 0 | machine       : 0 | memory        : 0 | men           : 0 | mental        : 0 | mention       : 0
might         : 0 | mind          : 0 | more          : 0 | most          : 0 | motives       : 0 | name          : 0 | nature        : 0 | never         : 0
not           : 0 | observer      : 0 | observing     : 0 | of            : 0 | one           : 0 | or            : 0 | other         : 0 | own           : 0
particularly  : 0 | passions      : 0 | perfect       : 0 | placed        : 0 | position      : 0 | power         : 0 | precise       : 0 | predominates  : 0
questionable  : 0 | reasoner      : 0 | reasoning     : 0 | results       : 0 | save          : 0 | seen          : 0 | seldom        : 0 | sensitive     : 0
sex           : 0 | she           : 1 | sherlock      : 1 | sneer         : 0 | softer        : 0 | spoke         : 0 | strong        : 0 | such          : 0
take          : 0 | temperament   : 0 | than          : 0 | that          : 0 | the           : 1 | there         : 0 | they          : 0 | things        : 0
throw         : 0 | to            : 1 | trained       : 0 | under         : 0 | upon          : 0 | veil          : 0 | was           : 0 | were          : 0
which         : 0 | whole         : 0 | with          : 0 | woman         : 1 | world         : 0 | would         : 0 | yet           : 0
```

Limitations

- **Cosine Similarity**
  - Similar words in different dimensions will hurt similarity
  - Like “cat” and “cats”

- **Vocabulary**
  - Grows very fast, but mainly because of noise
  - Think typos, numbers, OCR artefacts, …
  - Some words don’t bring information (*and, or, I, …*)

- **Composition**
  - Order of words does not matter for BoW
  - Some groups of words carry information (*New York, Black Sea, …*)

Text Processing

Stemming

- Reduce a token to a **stem**
- "cats" → "cat", "making" → "mak"

- All different forms of a word will be counted under the same dimension
  - Plurals
  - Conjugations
- Stem ≠ Word (ex: "mak" is the stem of "making", but is not an English word)

- **Different stemmers:** Porter, Snowball / Arabic, Chinese, German, …

Text Processing

Lemmatizing

- Reduce a token to a **lemma**
- "cats" → "cat", "making" → "make"

- All different forms of a word will be counted under the same dimension
  - Plurals
  - Conjugations

- Lemma = Word

- **Different lemmatizers:** WordNet, SpaCy, TextBlob, StanfordCoreNLP, etc… (See also)

Text Processing

Stopping

- Remove tokens
- Any text contains “syntactic sugar”
- Those words are literally everywhere, poisoning **cosine similarity**

- How much information in words like:
  - the, a, an, my, …
  - “to be” (think “my cat is blue” versus “cat – blue”)

- Based on pre-established lists

Text Processing

Filtering by Document Frequencies

- The document frequency of a token is the number of documents in which a token appears

- Token is everywhere: does not matter for similarity
- Token is hardly there: very likely a typo, or a single (number, name, …)

Text Processing

N-Grams

- BoW ignores word order

- Collect groups of N consecutive tokens in text
- See which ones repeat

- 2-grams: “New York”, “Black sea”, “Joe Biden”
- 3-grams: “New York City”, “Great Wall of China”

- Two texts with “New York” will have higher **cosine similarity** as this 2-gram has a dimension in BoW

Text Processing

- See the Notebook “Text Processing”

: 0	returned	: 0	returns	: 0	rich	: 0
: 0	room	: 0	rooms	: 0	rose	: 0
: 0	running	: 0	rush	: 0	rushed	: 0
: 0	sat	: 0	save	: 0	saw	: 0
: 0	scissors	: 0	scissors grinder	: 0	secure	: 0
: 0	send	: 0	sent	: 0	seriously	: 0
: 0	servant	: 0	seven	: 0	shall	: 0
: 0	sherlock	: 1	sherlock holmes	: 1	shoulders	: 0
: 0	signal	: 0	silence	: 0	simple	: 0
: 0	sitting room	: 0	situation	: 0	sliding	: 0
: 0	smoke rocket	: 0	soon	: 0	sort	: 0
: 0	sovereign reach	: 0	spare	: 0	spoke	: 0
: 0	st monica	: 0	stage	: 0	stairs	: 0
: 0	stay	: 0	steel	: 0	stepped	: 0
: 0	strange	: 0	strange visitor	: 0	street	: 0
: 0	study	: 0	successful	: 0	sure	: 0
: 0	taken	: 0	taking	: 0	tell	: 0
: 0	thing	: 0	things	: 0	think	: 0
: 0	tie	: 0	time	: 0	times	: 0
: 0	took heavy	: 0	tore	: 0	trained	: 0

Take Away

- **Bag of Words** produces vectors from text
- Text is split in **tokens**
- Unique tokens form a **Vocabulary**
- **Stemming** / **Lemmatizing** transform tokens
- **Stopping** / **Filtering** remove tokens

- BoW vectors count tokens, or **N-grams**
- **TF-IDF** corrects for term frequency in corpus