

# Introduction

## Applied Text Mining

Ayoub Bagheri

## Contents

.....	3
.....	3
.....	4
<b>Course Logistics</b>	<b>4</b>
Course materials .....	4
Teachers .....	4
Program .....	6
Goal of the course .....	6
<b>Python?</b>	<b>6</b>
How familiar are you with Python? .....	6
Python IDE? .....	6
Google Colab? .....	7
Python .....	7
Google Colab .....	8
<b>What is Text Mining?</b>	<b>8</b>
Text mining in an example .....	8
.....	9
Example .....	10
Example .....	10
Language is hard! .....	11
<b>Text Mining to the Rescue!</b>	<b>12</b>
Text mining .....	12
Language is hard! .....	12

<b>Process &amp; Tasks</b>	<b>12</b>
Text mining process . . . . .	12
Text mining tasks . . . . .	12
And more in NLP . . . . .	13
<b>Text Preprocessing</b>	<b>13</b>
Text preprocessing . . . . .	13
Typical steps . . . . .	13
Tokenization/Segmentation . . . . .	13
N-grams . . . . .	14
Part Of Speech (POS) tagging . . . . .	14
<b>Vector Space Model</b>	<b>14</b>
Basic idea . . . . .	14
How to represent a document . . . . .	15
Vector space model . . . . .	15
Vector space model . . . . .	15
Vector space model . . . . .	15
An illustration of VS model . . . . .	16
VSM: How do we represent vectors? . . . . .	16
Bag of Words (BOW) . . . . .	16
Binary . . . . .	16
Term Frequency . . . . .	17
TF: Document - Term Matrix (DTM) . . . . .	17
TFiDF . . . . .	17
TFiDF: Document - Term matrix (DTM) . . . . .	18
How to define a good similarity metric? . . . . .	18
How to define a good similarity metric? . . . . .	19
Next . . . . .	19
<b>Summary</b>	<b>19</b>
Summary . . . . .	19
<b>Practical 1</b>	<b>19</b>

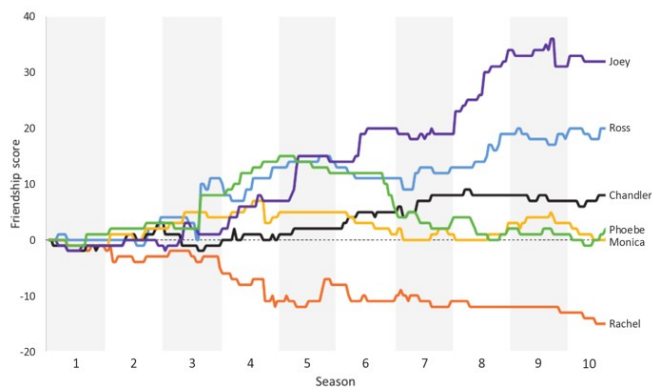
# Did a poet with donkey ears write the oldest anthem in the world?

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



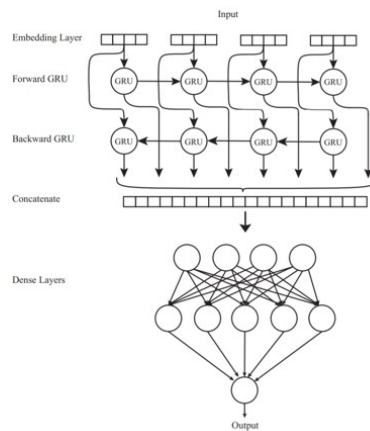
# Who was the best Friend?

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



# Automatic detection of ICD10 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, zweeten, misselijk. Ambulance gebeld en bij aansluiten monitor beeld van acuut onderwandinfarct.  
**AMBU overdracht** 500 mg aspegic iv, ticagrelor 180 mg oraal, heparine, zofran eenmalig, 3x 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/III C/vermoedelijk RCA.  
**Coronair angiografie** (...). Conclusie angio: 1-vatslijden. PCI  
**Conclusie en beleid**  
Bovengenoemde <LEEFTIJD-1>jarige man, blanco cardiale voorgeschiedenis, werd gerepresenteerd 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 dispersietablet 80 mg; oraal: 1x per dag 80 milligram; <DATUM-1>; Ticagrelor tablet 90 mg; oraal: 2x per dag 90 milligram; <DATUM-1>; Metoprolol tablet 50 mg; oraal: 2x per dag 25 milligram; <DATUM-1>; Atorvastatine tablet 40 mg (als ca-zout-3-water); oraal: 1x per dag 40 milligram; <DATUM-1>  
**Samenvatting**  
Hoofdiagnose: STEMI inferior wv PCI RCA. Geen nevenletsels. Nevenaandoeningen: geen.  
Complicaties: geen Ontslag naar: CCU <INSTELLING-2>.

## Course Logistics

### Course materials

You can access the course materials quickly from

[https://ayoubbagheri.nl/applied\\_tm/](https://ayoubbagheri.nl/applied_tm/)

### Teachers



Anastasia



Arjan



Luka



Dong



Daniel

## Program

Time	Monday	Tuesday	Wednesday	Thursday	Friday
<b>9:00 - 10:30</b>	<b>Lecture 1</b>	<b>Lecture 3</b>	<b>Lecture 5</b>	<b>Lecture 7</b>	<b>Lecture 9</b>
	Break	Break	Break	Break	Break
<b>10:45 - 11:45</b>	Practical 1	Practical 3	Practical 5	Practical 7	Practical 9
<b>11:45 - 12:15</b>	Discussion 1	Discussion 3	Discussion 5	Discussion 7	Discussion 9
	Lunch	Lunch	Lunch	Lunch	Lunch
<b>13:45 - 15:15</b>	<b>Lecture 2</b>	<b>Lecture 4</b>	<b>Lecture 6</b>	<b>Lecture 8</b>	<b>Lecture 10</b>
	Break	Break	Break	Break	Break
<b>15:30 - 16:30</b>	Practical 2	Practical 4	Practical 6	Practical 8	Practical 10
<b>16:30 - 17:00</b>	Discussion 2	Discussion 4	Discussion 6	Discussion 8	Discussion 10

## Goal of the course


- Text data are everywhere!
- A lot of world's data are in the format of unstructured text
- This course teaches
  - text mining techniques
  - using Python
  - on a variety of applications
  - in many domains.

## Python?


### How familiar are you with Python?

- What is your experience level with Python?





- 1 Connect to [www.wooclap.com/DXFCZD](http://www.wooclap.com/DXFCZD)
- 2 You can participate



- 1 Not yet connected? Send [@DXFCZD](#) to **0970 1420 2908**
- 2 Send your answer to the same number

## Python IDE?

- Which Python IDE do you mostly use? If you use more than one environment fill in the other text boxes.



- 1 Connect to [www.wooclap.com/DXFCZD](https://www.wooclap.com/DXFCZD)
- 2 You can participate



- 1 Not yet connected? Send @DXFCZD to 0970 1420 2908
- 2 Send your answer to the same number

## Google Colab?

- How familiar are you with Google Colab? (1: limited to 5: expert)



- 1 Connect to [www.wooclap.com/DXFCZD](https://www.wooclap.com/DXFCZD)
- 2 You can participate



- 1 Not yet connected? Send @DXFCZD to 0970 1420 2908
- 2 Send your answer to the same number

## Python

- Latest: Python 3.10
- Follow the tutorial on Python in Google Colab for the Applied Text Mining course: link
- Python For Beginners
  - <https://www.python.org/about/gettingstarted/>
- The Python Language Reference
  - <https://docs.python.org/3/reference/>
- Python 3.9.1 documentation
  - <https://docs.python.org/3/>

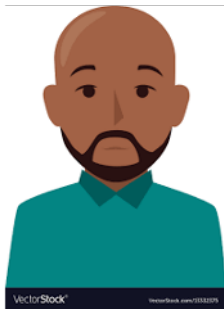
## Google Colab

- Colaboratory, or “Colab” for short, allows you to write and execute Python in your browser, with
  - Zero configuration required
  - Free access to GPUs
  - Easy sharing
- [Intro](https://colab.research.google.com/notebooks/intro.ipynb)
- Cheat-sheet for Google Colab
- Keyboard shortcuts:

1	Actions	Colab	Jupyter
2	show keyboard shortcuts	Ctrl/Cmd M H	H
3	Insert code cell above	Ctrl/Cmd M A	A
4	Insert code cell below	Ctrl/Cmd M B	B
5	Delete cell/selection	Ctrl/Cmd M D	DD
6	Interrupt execution	Ctrl/Cmd M I	II
7	Convert to code cell	Ctrl/Cmd M Y	Y
8	Convert to text cell	Ctrl/Cmd M M	M
9	Split at cursor	Ctrl/Cmd M -	Ctrl Shift -

## What is Text Mining?

### Text mining in an example



- This is **Garry**!
- **Garry** works at Bol.com (a webshop in the Netherlands)
- He works in the dep of **Customer relationship management**.
- He uses Excel to read and search customers' reviews, extract aspects they wrote their reviews on, and identify their sentiments.
- Curious about his job? See two examples!



This is a nice book for both young and old. It gives beautiful life lessons in a fun way. Definitely worth the money!

+ Educational

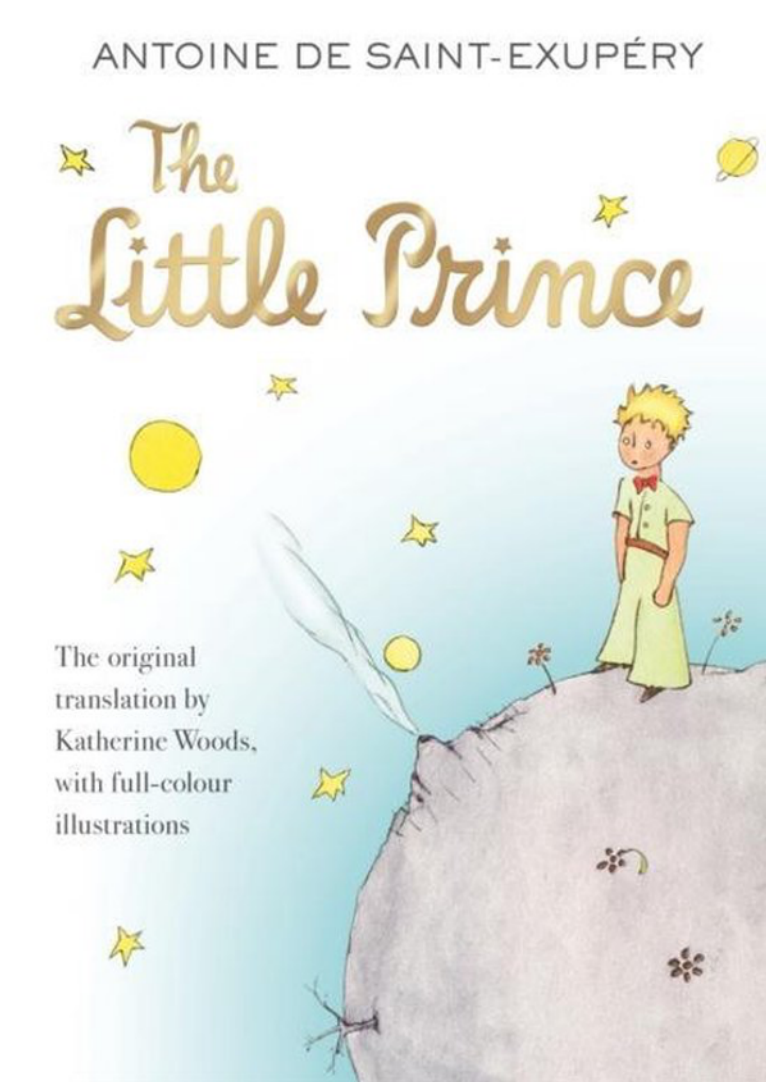
+ Funny

+ Price

Nice story for older children.

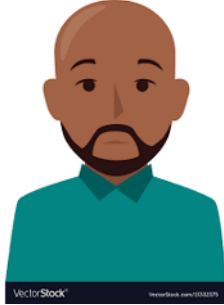
+ Funny

- Readability



## Example

---

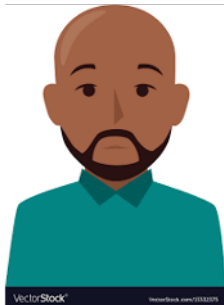


- Garry likes his job a lot, but sometimes it is frustrating!
- This is mainly because their company is expanding quickly!
- Garry decides to hire **Larry** as his assistant.



## Example

---





- Still, a lot to do for two people!
- Garry has some budget left to hire another assistant for couple of years!
- He decides to hire **Harry** too!
- Still, manual labeling using Excel is labor-intensive!



## Language is hard!

- Different things can mean more or less the same (“data science” vs. “statistics”)
- Context dependency (“You have very nice shoes”);
- Same words with different meanings (“to sanction”, “bank”);
- Lexical ambiguity (“we saw her duck”)
- Irony, sarcasm (“That’s just what I needed today!”, “Great!”, “Well, what a surprise.”)
- Figurative language (“He has a heart of stone”)
- Negation (“not good” vs. “good”), spelling variations, jargon, abbreviations
- All the above are different over languages, 99% of work is on English!

# Text Mining to the Rescue!

## Text mining

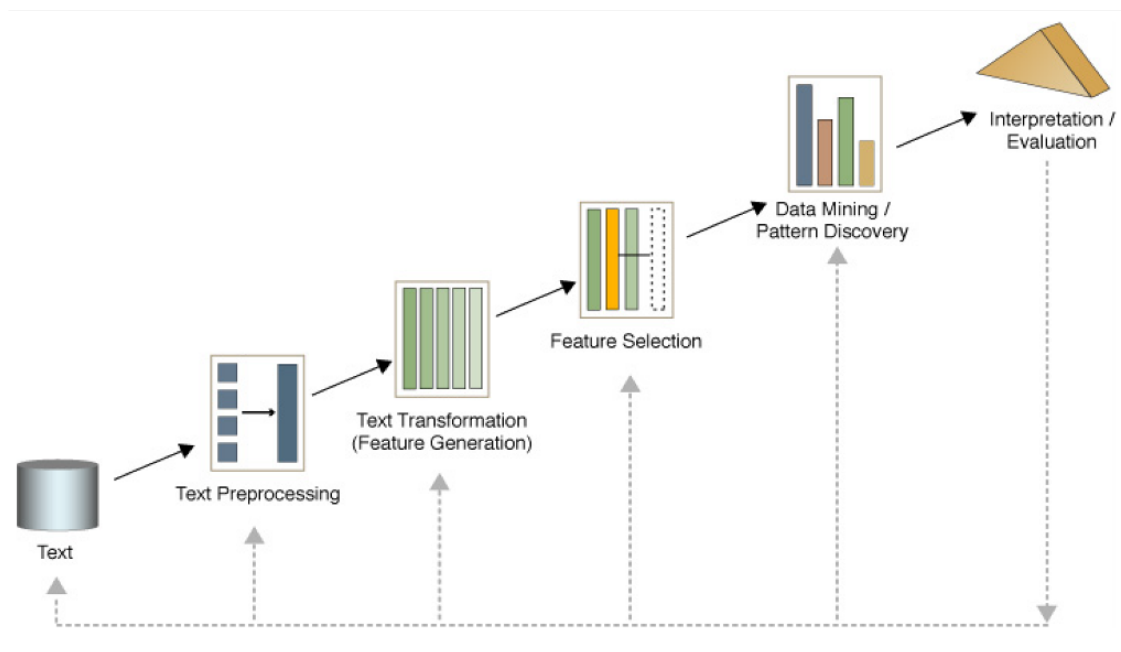
- “the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources” Hearst (1999)
- Text mining is about looking for patterns in text, in a similar way that data mining can be loosely described as looking for patterns in data.
- Text mining describes a set of linguistic, statistical, and machine learning techniques that model and structure the information content of textual sources. (Wikipedia)

## Language is hard!

- We won’t solve linguistics ...
- In spite of the problems, text mining can be quite effective!

## Process & Tasks

### Text mining process

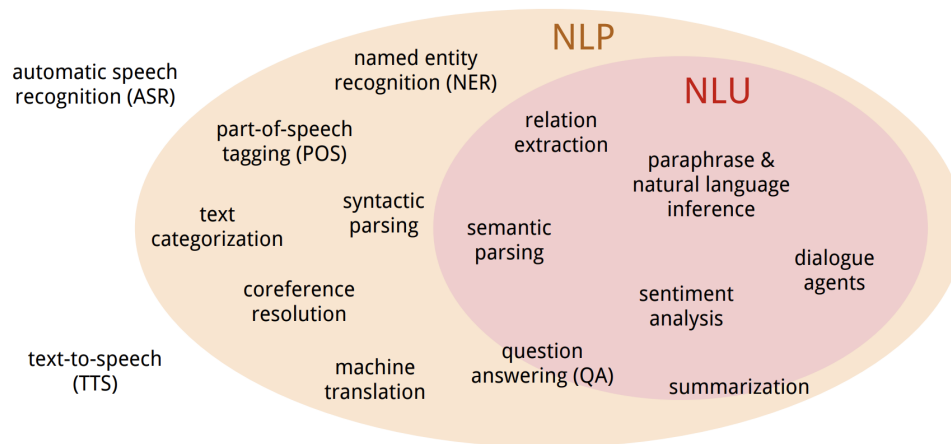


### Text mining tasks

- Text classification
- Text clustering
- Sentiment analysis
- Feature selection
- Topic modelling

- Responsible text mining
- Text summarization

## And more in NLP



source: <https://nlp.stanford.edu/~wcmac/papers/20140716-UNLU.pdf>

## Text Preprocessing

### Text preprocessing

- is an approach for cleaning and noise removal of text data.
- brings your text into a form that is analyzable for your task.
- transforms text into a more digestible form so that machine learning algorithms can perform better.

### Typical steps

- Tokenization (“text”, “ming”, “is”, “the”, “best”, “!”)
- Stemming (“lungs”→“lung”) or Lemmatization (“were”→“is”)
- Lowercasing (“Disease”→“disease”)
- Stopword removal (“text ming is best!”)
- Punctuation removal (“text ming is the best”)
- Number removal (“I42”→“I”)
- Spell correction (“hart”→“heart”)

Not all of these are appropriate at all times!

### Tokenization/Segmentation

- Split text into words and sentences

There was an earthquake  
near D.C. I've even felt it in  
Philadelphia, New York,  
etc.

There + was + an +  
earthquake + near + D.C.

I + ve + even + felt + it + in +  
Philadelphia, + New + York, +  
etc.

## N-grams

- N-grams: a contiguous sequence of N tokens from a given piece of text
  - E.g., ‘Text mining is to identify useful information.’
  - Bigrams: ‘text\_mining’, ‘mining\_is’, ‘is\_to’, ‘to\_identify’, ‘identify\_useful’, ‘useful\_information’, ‘information\_’.
- Pros: capture local dependency and order
- Cons: increase the vocabulary size

## Part Of Speech (POS) tagging

- Annotate each word in a sentence with a part-of-speech.

I ate the spaghetti with meatballs.  
Pro V Det N Prep N

John saw the saw and decided to take it to the table.  
PN V Det N Con V Part V Pro Prep Det N

- Useful for subsequent syntactic parsing and word sense disambiguation.

## Vector Space Model

### Basic idea

- Text is “unstructured data”
- How do we get to something structured that we can compute with?
- **Text must be represented somehow**
- Represent the text as something that makes sense to a computer

## How to represent a document

- Represent by a string?
  - No semantic meaning
- Represent by a list of sentences?
  - Sentence is just like a short document (recursive definition)
- Represent by a vector?
  - A vector is an ordered finite list of numbers.

## Vector space model

- A vector space is a collection of vectors
- Represent documents by concept vectors
  - Each concept defines one dimension
  - k concepts define a high-dimensional space
  - Element of vector corresponds to concept weight

## Vector space model

- Distance between the vectors in this concept space
  - Relationship among documents
- The process of converting text into numbers is called Vectorization

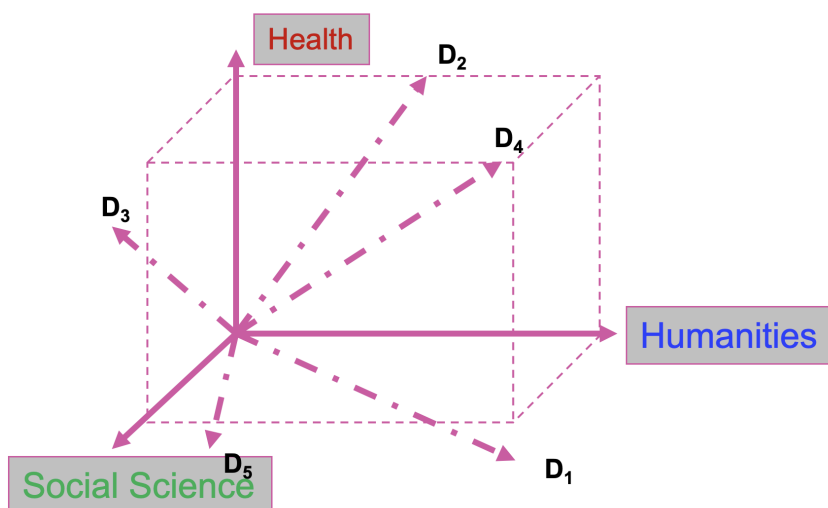
## Vector space model

- Terms are generic features that can be extracted from text
- Typically, terms are single words, keywords, n-grams, or phrases
- Documents are represented as vectors of terms
- Each dimension (concept) corresponds to a separate term

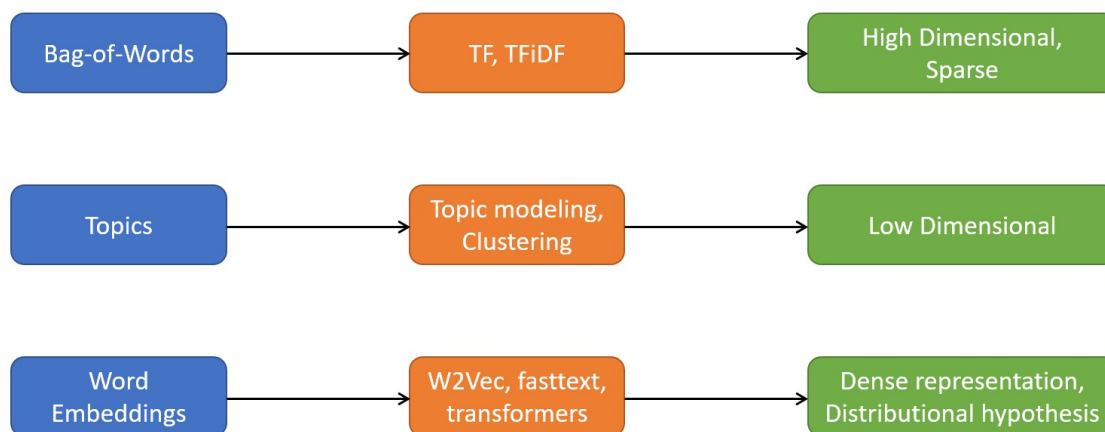
$$d = (w_1, \dots, w_n)$$

## An illustration of VS model

- All documents are projected into this concept space



## VSM: How do we represent vectors?



## Bag of Words (BOW)

- *Terms* are words (more generally we can use n-grams)
- *Weights* are number of occurrences of the terms in the document
  - Binary
  - Term Frequency (TF)
  - Term Frequency inverse Document Frequency (TFIDF)

## Binary

- Doc1: Text mining is to identify useful information.



- Doc2: Useful information is mined from text.
- Doc3: Apple is delicious.

	text	information	identify	mining	mined	is	useful	to	from	apple	delicious
Doc1	1	1	1	1	0	1	1	1	0	0	0
Doc2	1	1	0	0	1	1	1	0	1	0	0
Doc3	0	0	0	0	0	1	0	0	0	1	1

## Term Frequency

- Idea: a term is more important if it occurs more frequently in a document
- TF formulas
  - Let  $t(c, d)$  be the frequency count of term  $t$  in doc  $d$
  - Raw TF:  $tf(t, d) = c(t, d)$

## TF: Document - Term Matrix (DTM)

### Bag of words

- d1: "And God said, Let there be light: and there was light."
- d2: "And God saw the light, that it was good: and God divided the light from the darkness."
- d3: "And God called the light Day, and the darkness he called Night. And the evening and the morning were the first day."

### "Document - Term matrix" (DTM) (raw word counts)

	light	god	darkness	called	day	let	said	divided	good	saw	evening	first	morning	night
d1	2	1	0	0	0	1	1	0	0	0	0	0	0	0
d2	2	2	1	0	0	0	0	1	1	1	0	0	0	0
d3	1	1	1	2	2	0	0	0	0	0	1	1	1	1

## TFiDF

- Idea: a term is more discriminative if it occurs a lot but only in fewer documents

Let  $n_{d,t}$  denote the number of times the  $t$ -th term appears in the  $d$ -th document.

$$TF_{d,t} = \frac{n_{d,t}}{\sum_i n_{d,i}}$$

Let  $N$  denote the number of documents and  $N_t$  denote the number of documents containing the  $t$ -th term.

$$IDF_t = \log\left(\frac{N}{N_t}\right)$$

TFiDF weight:

$$w_{d,t} = TF_{d,t} \cdot IDF_t$$

## TFiDF: Document - Term matrix (DTM)

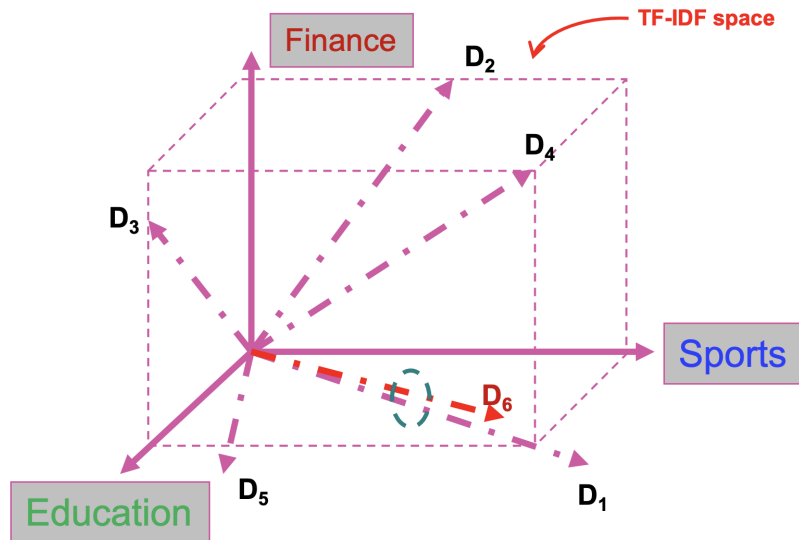
### Bag of words

- d1: "And God said, Let there be light: and there was light."
- d2: "And God saw the light, that it was good: and God divided the light from the darkness."
- d3: "And God called the light Day, and the darkness he called Night. And the evening and the morning were the first day."

### "Document - Term matrix" (DTM) (tf-idf)

	light	god	darkness	called	day	let	said	divided	good	saw	evening	first	morning	night
d1	0	0	0.000	0.0	0.0	1.1	1.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
d2	0	0	0.405	0.0	0.0	0.0	0.0	1.1	1.1	1.1	0.0	0.0	0.0	0.0
d3	0	0	0.405	2.2	2.2	0.0	0.0	0.0	0.0	0.0	1.1	1.1	1.1	1.1

## How to define a good similarity metric?



## How to define a good similarity metric?

- Euclidean distance

$$dist(d_i, d_j) = \sqrt{\sum_{t \in V} [tf(t, d_i)idf(t) - tf(t, d_j)idf(t)]^2}$$

- Longer documents will be penalized by the extra words
- We care more about how these two vectors are overlapped

- Cosine similarity

- Angle between two vectors:

$$cosine(d_i, d_j) = \frac{V_{d_i}^T V_{d_j}}{|V_{d_i}|_2 \times |V_{d_j}|_2} \leftarrow \text{TF-IDF vector}$$

- Documents are normalized by length

## Next

- Text classification

## Summary

### Summary

- Text data are everywhere!
- Language is hard!
- The basic problem of text mining is that text is not a neat data set
- Solution: text pre-processing & VSM

## Practical 1