Data Analysis - Lecture 7

Introduction to Text Mining

Dr. Jérémie Sublime

LISITE Laboratory - DaSSIP Team - ISEP
LIPN - CNRS UMR 7030

jeremie.sublime@isep.fr

Plan

1. Introduction
2. Text mining tasks and processes
3. Analyzing similarities between and within texts
4. Latent Dirichlet Allocation
5. Word2vec
6. Conclusion

Outline

1. Introduction
2. Text mining tasks and processes
3. Analyzing similarities between and within texts
4. Latent Dirichlet Allocation
5. Word2vec
6. Conclusion

Working with texts is important and hard !

Text Mining: Definition

Text Mining or Text data Mining or approximately **Text Analytics**, is the process of deriving high-quality information from texts.

What texts ?

- Books
- Web pages
- Social networks
- Science articles
- ...

Very different contents, with different formats and potentially large amount of information to be extracted.

Data analysis so far

In our previous classes, we worked only with numerical data:

- Numerical vectors
- Numerical time series

We did a bit of clustering, visualization and classification:

- All of them are based on distance or similarity functions

Data analysis so far

In our previous classes, we worked only with numerical data:

- Numerical vectors
- Numerical time series

We did a bit of clustering, visualization and classification:

- All of them are based on distance or similarity functions

Problem: text data represent the largest amount of data !

How do we do all of these with text data ?

- Similarity between words, sentences, or texts ?
- Classification or clustering of texts ?
- Text time series analysis ?

Stylometry: an early attempt at text analysis

- Stylometry is a science used in case of disputed authorship of a text: it aims at analysis writing patterns.
- Thomas C. Mendenhall pioneered stylometry in 1887 by proposing to analyze and compare the "**word spectrum**" of texts.
- He was a physicist and a meteorologist at The Ohio State University from 1873 and later became President of the Worcester Polytechnic Institute.

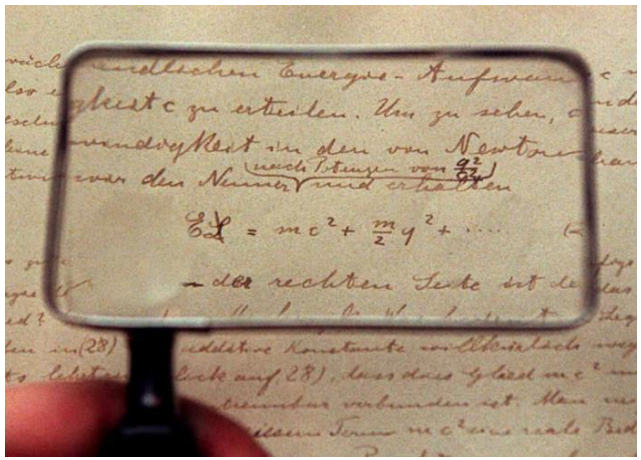
SCIENCE.-

FRIDAY, MARCH 11, 1887.

THE CHARACTERISTIC CURVES OF COM-POSITION.

The age of information overload

- Mendenhall's techniques and the idea of "word spectrum" are still used in modern text mining algorithms. But we added a few tools since:
  - Word semantic analysis
  - Sentiment analysis of words
  - Similarity links and measures between words
- The main difference between 1887 and now, is that nowadays there is way too much information coming in real time for a manual analysis to be possible !



Examples of text analysis tasks

The most common uses of text mining include but are not limited to:

- Machine translation
- Part-of-speech tagging
- Information extraction
- Sentiment analysis from text
- Automatic quizz answering
- Text categorization
- Stylometry and plagiarism analysis

Examples of text analysis tasks

The most common uses of text mining include but are not limited to:

- Machine translation
- Part-of-speech tagging
- Information extraction
- Sentiment analysis from text
- Automatic quizz answering
- Text categorization
- Stylometry and plagiarism analysis

Some of these applications are very important for world leaders.

Text Mining VS Natural Language Processing

Text Mining

It deals with the text itself

- Frequency count of words
- Sentence length
- Word spectrum
- Presence/absence of key words
- etc.

Natural Language Processing

It deals with the underlying metadata

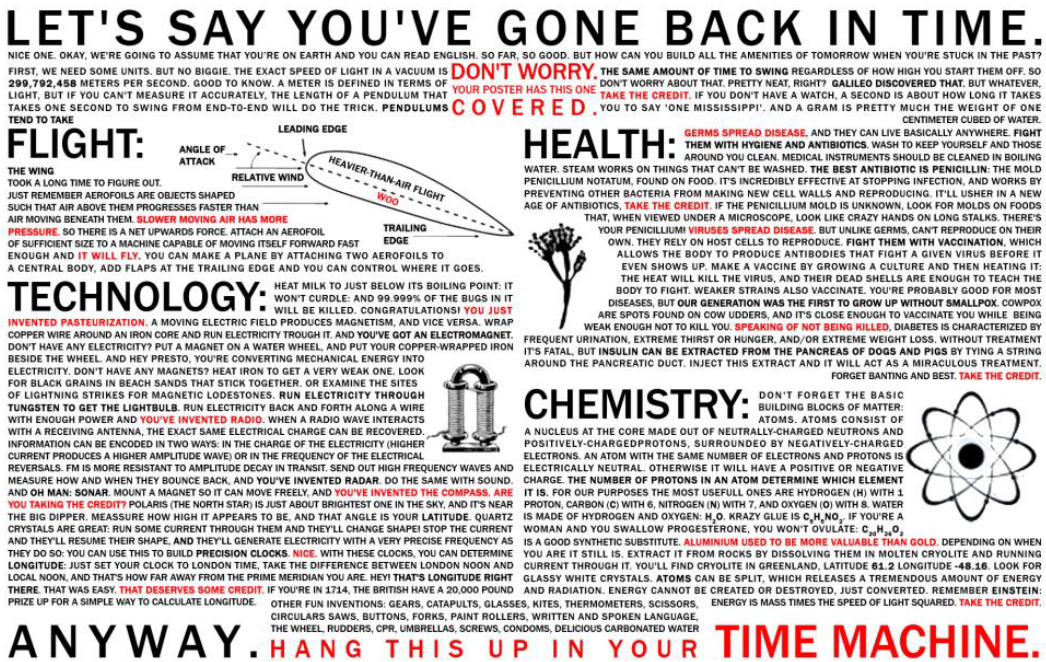
- Linguistics
- Computational linguistics
- Content analysis
- Stylistics
- etc.

Roughly summed up:

- Text Mining is statistics oriented.
- NLP is analysis oriented.

Mining the texts: what are we trying to understand ?

- **The text themselves ?**
- **The authors of the texts ?**
  - The author as a writer ?
  - The author as a person in the world ?
- **Things in the world ?**
  - Described by the text ?
  - Linked to the text ?



LET'S SAY YOU'VE GONE BACK IN TIME.

NICE ONE. OKAY, WE'RE GOING TO ASSUME THAT YOU'RE ON EARTH AND YOU CAN READ ENGLISH. SO FAR, SO GOOD. BUT HOW CAN YOU BUILD ALL THE AMENITIES OF TOMORROW WHEN YOU'RE STUCK IN THE PAST?

FIRST, WE NEED SOME UNITS. BUT NO BIGGIE. THE EXACT SPEED OF LIGHT IN A VACUUM IS **299,792,458** METERS PER SECOND. GOOD TO KNOW. A METER IS DEFINED IN TERMS OF LIGHT, BUT IF YOU CAN'T MEASURE IT ACCURATELY, THE LENGTH OF A PENDULUM THAT TAKES ONE SECOND TO SWING FROM END-TO-END WILL DO THE TRICK. **PENDULUMS TEND TO TAKE** DON'T WORRY. YOUR POSTER HAS THIS ONE COVERED. **THE SAME AMOUNT OF TIME TO SWING** REGARDLESS OF HOW HIGH YOU START THEM OFF. SO DON'T WORRY ABOUT THAT. PRETTY NEAT, RIGHT? **GALILEO DISCOVERED THAT.** BUT WHATEVER, TAKE THE CREDIT. IF YOU DON'T HAVE A WATCH, A SECOND IS ABOUT HOW LONG IT TAKES YOU TO SAY 'ONE MISSISSIPPI'. AND A GRAM IS PRETTY MUCH THE WEIGHT OF ONE CENTIMETER CUBED OF WATER.

FLIGHT:

THE WING TOOK A LONG TIME TO FIGURE OUT. JUST REMEMBER AEROFOILS ARE OBJECTS SHAPED SUCH THAT AIR ABOVE THEM PROGRESSES FASTER THAN AIR MOVING BENEATH THEM. SLOWER MOVING AIR HAS MORE PRESSURE. SO THERE IS A NET UPWARDS FORCE. ATTACH AN AEROFOIL OF SUFFICIENT SIZE TO A MACHINE CAPABLE OF MOVING ITSELF FORWARD FAST ENOUGH AND IT WILL FLY. YOU CAN MAKE A PLANE BY ATTACHING TWO AEROFOILS TO A CENTRAL BODY, ADD FLAPS AT THE TRAILING EDGE AND YOU CAN CONTROL WHERE IT GOES.

HEALTH: **GERMS SPREAD DISEASE.** AND THEY CAN LIVE BASICALLY ANYWHERE. **FIGHT THEM WITH HYGIENE AND ANTIBIOTICS.** WASH TO KEEP YOURSELF AND THOSE AROUND YOU CLEAN. MEDICAL INSTRUMENTS SHOULD BE CLEANED IN BOILING WATER. STEAM WORKS ON THINGS THAT CAN'T BE WASHED. **THE BEST ANTIBIOTIC IS PENICILLIN:** THE MOLD PENICILLIUM NOTATUM, FOUND ON FOOD. IT'S INCREDIBLY EFFECTIVE AT STOPPING INFECTION, AND WORKS BY PREVENTING OTHER BACTERIA FROM MAKING NEW CELL WALLS AND REPRODUCING. IT'LL USHER IN A NEW AGE OF ANTIBIOTICS, TAKE THE CREDIT. IF THE PENICILLIUM MOLD IS UNKNOWN, LOOK FOR MOLDS ON FOODS THAT, WHEN VIEWED UNDER A MICROSCOPE, LOOK LIKE CRAZY HANDS ON LONG STALKS. THERE'S YOUR PENICILLIUM! **VIRUSES SPREAD DISEASE.** BUT UNLIKE GERMS, CAN'T REPRODUCE ON THEIR OWN. THEY RELY ON HOST CELLS TO REPRODUCE. **FIGHT THEM WITH VACCINATION,** WHICH ALLOWS THE BODY TO PRODUCE ANTIBODIES THAT FIGHT A GIVEN VIRUS BEFORE IT EVEN SHOWS UP. MAKE A VACCINE BY GROWING A CULTURE AND THEN HEATING IT: THE HEAT WILL KILL THE VIRUS, AND THEIR DEAD SHELLS ARE ENOUGH TO TEACH THE BODY TO FIGHT. WEAKER STRAINS ALSO VACCINATE. YOU'RE PROBABLY GOOD FOR MOST DISEASES, BUT **OUR GENERATION WAS THE FIRST TO GROW UP WITHOUT SMALLPOX.** COWPOX ARE SPOTS FOUND ON COW UDDERS, AND IT'S CLOSE ENOUGH TO VACCINATE YOU WHILE BEING WEAK ENOUGH NOT TO KILL YOU. SPEAKING OF NOT BEING KILLED, DIABETES IS CHARACTERIZED BY FREQUENT URINATION, EXTREME THIRST OR HUNGER, AND/OR EXTREME WEIGHT LOSS. WITHOUT TREATMENT IT'S FATAL. BUT **INSULIN CAN BE EXTRACTED FROM THE PANCREAS OF DOGS AND PIGS** BY TYING A STRING AROUND THE PANCREATIC DUCT. INJECT THIS EXTRACT AND IT WILL ACT AS A MIRACULOUS TREATMENT. FORGET BANTING AND BEST. TAKE THE CREDIT.

TECHNOLOGY: HEAT MILK TO JUST BELOW ITS BOILING POINT: IT WON'T CURDLE- AND 99.999% OF THE BUGS IN IT WILL BE KILLED. CONGRATULATIONS! YOU JUST INVENTED PASTEURIZATION. A MOVING ELECTRIC FIELD PRODUCES MAGNETISM, AND VICE VERSA. WRAP COPPER WIRE AROUND AN IRON CORE AND RUN ELECTRICITY TROUGH IT. AND **YOU'VE GOT AN ELECTROMAGNET.** DON'T HAVE ANY ELECTRICITY? PUT A MAGNET ON A WATER WHEEL, AND PUT YOUR COPPER-WRAPPED IRON BESIDE THE WHEEL. AND HEY PRESTO, YOU'RE CONVERTING MECHANICAL ENERGY INTO ELECTRICITY. DON'T HAVE ANY MAGNETS? HEAT IRON TO GET A VERY WEAK ONE. LOOK FOR BLACK GRAINS IN BEACH SANDS THAT STICK TOGETHER. OR EXAMINE THE SITES OF LIGHTNING STRIKES FOR MAGNETIC LODESTONES. **RUN ELECTRICITY THROUGH TUNGSTEN TO GET THE LIGHTBULB.** RUN ELECTRICITY BACK AND FORTH ALONG A WIRE WITH ENOUGH POWER AND YOU'VE INVENTED RADIO. WHEN A RADIO WAVE INTERACTS WITH A RECEIVING ANTENNA, THE EXACT SAME ELECTRICAL CHARGE CAN BE RECOVERED. INFORMATION CAN BE ENCODED IN TWO WAYS: IN THE CHARGE OF THE ELECTRICITY (HIGHER CURRENT PRODUCES A HIGHER AMPLITUDE WAVE) OR IN THE FREQUENCY OF THE ELECTRICAL REVERSALS. FM IS MORE RESISTANT TO AMPLITUDE DECAY IN TRANSIT. SEND OUT HIGH FREQUENCY WAVES AND MEASURE HOW AND WHEN THEY BOUNCE BACK, AND **YOU'VE INVENTED RADAR.** DO THE SAME WITH SOUND. AND **OH MAN: SONAR.** MOUNT A MAGNET SO IT CAN MOVE FREELY, AND YOU'VE INVENTED THE COMPASS. ARE YOU TAKING THE CREDIT? POLARIS (THE NORTH STAR) IS JUST ABOUT BRIGHTEST ONE IN THE SKY, AND IT'S NEAR THE BIG DIPPER. MEASSURE HOW HIGH IT APPEARS TO BE, AND THAT ANGLE IS YOUR **LATITUDE.** QUARTZ CRYSTALS ARE GREAT: RUN SOME CURRENT THROUGH THEM AND THEY'LL CHANGE SHAPE! STOP THE CURRENT AND THEY'LL RESUME THEIR SHAPE, **AND** THEY'LL GENERATE ELECTRICITY WITH A VERY PRECISE FREQUENCY AS THEY DO SO: YOU CAN USE THIS TO BUILD **PRECISION CLOCKS.** NICE. WITH THESE CLOCKS, YOU CAN DETERMINE **LONGITUDE:** JUST SET YOUR CLOCK TO LONDON TIME, TAKE THE DIFFERENCE BETWEEN LONDON NOON AND LOCAL NOON, AND THAT'S HOW FAR AWAY FROM THE PRIME MERIDIAN YOU ARE. HEY! **THAT'S LONGITUDE RIGHT THERE.** THAT WAS EASY. THAT DESERVES SOME CREDIT. IF YOU'RE IN 1714, THE BRITISH HAVE A 20,000 POUND PRIZE UP FOR A SIMPLE WAY TO CALCULATE LONGITUDE.

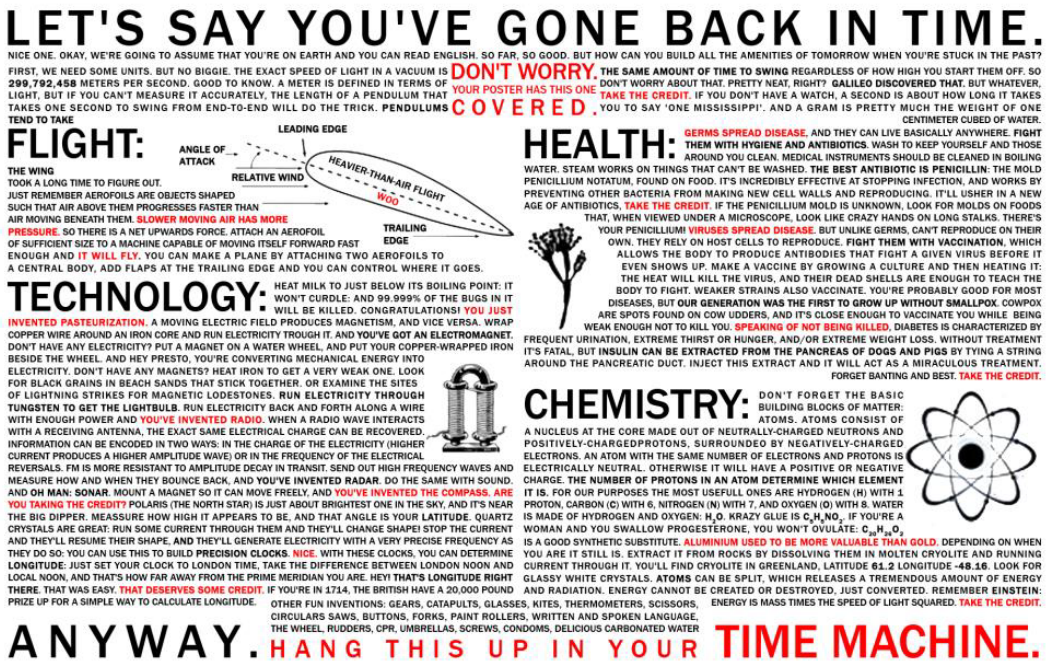
CHEMISTRY: DON'T FORGET THE BASIC BUILDING BLOCKS OF MATTER: ATOMS. ATOMS CONSIST OF A NUCLEUS AT THE CORE MADE OUT OF NEUTRALLY-CHARGED NEUTRONS AND POSITIVELY-CHARGEDPROTONS, SURROUNDEO BY NEGATIVELY-CHARGED ELECTRONS. AN ATOM WITH THE SAME NUMBER OF ELECTRONS AND PROTONS IS ELECTRICALLY NEUTRAL. OTHERWISE IT WILL HAVE A POSITIVE OR NEGATIVE CHARGE. **THE NUMBER OF PROTONS IN AN ATOM DETERMINE WHICH ELEMENT IT IS.** FOR OUR PURPOSES THE MOST USEFULL ONES ARE HYDROGEN (H) WITH 1 PROTON, CARBON (C) WITH 6, NITROGEN (N) WITH 7, AND OXYGEN (O) WITH 8. WATER IS MADE OF HYDROGEN AND OXYGEN: H_2O. KRAZY GLUE IS $C_4H_5NO_2$. IF YOU'RE A WOMAN AND YOU SWALLOW PROGESTERONE, YOU WON'T OVULATE: $C_{21}H_{30}O_2$. IS A GOOD SYNTHETIC SUBSTITUTE. ALUMINIUM USED TO BE MORE VALUABLE THAN GOLD. DEPENDING ON WHEN YOU ARE IT STILL IS. EXTRACT IT FROM ROCKS BY DISSOLVING THEM IN MOLTEN CRYOLITE AND RUNNING CURRENT THROUGH IT. YOU'LL FIND CRYOLITE IN GREENLAND, LATITUDE **61.2** LONGITUDE **-48.16**. LOOK FOR GLASSY WHITE CRYSTALS. **ATOMS** CAN BE SPLIT, WHICH RELEASES A TREMENDOUS AMOUNT OF ENERGY AND RADIATION. ENERGY CANNOT BE CREATED OR DESTROYED, JUST CONVERTED. REMEMBER **EINSTEIN:** ENERGY IS MASS TIMES THE SPEED OF LIGHT SQUARED. TAKE THE CREDIT.

OTHER FUN INVENTIONS: GEARS, CATAPULTS, GLASSES, KITES, THERMOMETERS, SCISSORS, CIRCULARS SAWS, BUTTONS, FORKS, PAINT ROLLERS, WRITTEN AND SPOKEN LANGUAGE, THE WHEEL, RUDDERS, CPR, UMBRELLAS, SCREWS, CONDOMS, DELICIOUS CARBONATED WATER

ANYWAY. HANG THIS UP IN YOUR TIME MACHINE.

Mining the texts: what are we trying to understand ?

- **The text themselves ?**
- **The authors of the texts ?**
  - The author as a writer ?
  - The author as a person in the world ?
- **Things in the world ?**
  - Described by the text ?
  - Linked to the text ?



LET'S SAY YOU'VE GONE BACK IN TIME.

NICE ONE. OKAY, WE'RE GOING TO ASSUME THAT YOU'RE ON EARTH AND YOU CAN READ ENGLISH. SO FAR, SO GOOD. BUT HOW CAN YOU BUILD ALL THE AMENITIES OF TOMORROW WHEN YOU'RE STUCK IN THE PAST?

FIRST, WE NEED SOME UNITS. BUT NO BIGGIE. THE EXACT SPEED OF LIGHT IN A VACUUM IS **299,792,458** METERS PER SECOND. GOOD TO KNOW. A METER IS DEFINED IN TERMS OF LIGHT, BUT IF YOU CAN'T MEASURE IT ACCURATELY, THE LENGTH OF A PENDULUM THAT TAKES ONE SECOND TO SWING FROM END-TO-END WILL DO THE TRICK. **PENDULUMS TEND TO TAKE**

DON'T WORRY. YOUR POSTER HAS THIS ONE **COVERED.**

THE SAME AMOUNT OF TIME TO SWING REGARDLESS OF HOW HIGH YOU START THEM OFF. SO DON'T WORRY ABOUT THAT. PRETTY NEAT, RIGHT? **GALILEO DISCOVERED THAT.** BUT WHATEVER, TAKE THE CREDIT. IF YOU DON'T HAVE A WATCH, A SECOND IS ABOUT HOW LONG IT TAKES YOU TO SAY 'ONE MISSISSIPPI'. AND A GRAM IS PRETTY MUCH THE WEIGHT OF ONE CENTIMETER CUBED OF WATER.

FLIGHT:

LEADING EDGE
ANGLE OF ATTACK
RELATIVE WIND
HEAVIER-THAN-AIR FLIGHT
WOO
TRAILING EDGE

THE WING TOOK A LONG TIME TO FIGURE OUT. JUST REMEMBER AEROFOILS ARE OBJECTS SHAPED SUCH THAT AIR ABOVE THEM PROGRESSES FASTER THAN AIR MOVING BENEATH THEM. **SLOWER MOVING AIR HAS MORE PRESSURE.** SO THERE IS A NET UPWARDS FORCE. ATTACH AN AEROFOIL OF SUFFICIENT SIZE TO A MACHINE CAPABLE OF MOVING ITSELF FORWARD FAST ENOUGH AND **IT WILL FLY.** YOU CAN MAKE A PLANE BY ATTACHING TWO AEROFOILS TO A CENTRAL BODY, ADD FLAPS AT THE TRAILING EDGE AND YOU CAN CONTROL WHERE IT GOES.

HEALTH: **GERMS SPREAD DISEASE.** AND THEY CAN LIVE BASICALLY ANYWHERE. **FIGHT THEM WITH HYGIENE AND ANTIBIOTICS.** WASH TO KEEP YOURSELF AND THOSE AROUND YOU CLEAN. MEDICAL INSTRUMENTS SHOULD BE CLEANED IN BOILING WATER. STEAM WORKS ON THINGS THAT CAN'T BE WASHED. **THE BEST ANTIBIOTIC IS PENICILLIN:** THE MOLD PENICILLIUM NOTATUM, FOUND ON FOOD. IT'S INCREDIBLY EFFECTIVE AT STOPPING INFECTION, AND WORKS BY PREVENTING OTHER BACTERIA FROM MAKING NEW CELL WALLS AND REPRODUCING. IT'LL USHER IN A NEW AGE OF ANTIBIOTICS, **TAKE THE CREDIT.** IF THE PENICILLIUM MOLD IS UNKNOWN, LOOK FOR MOLDS ON FOODS THAT, WHEN VIEWED UNDER A MICROSCOPE, LOOK LIKE CRAZY HANDS ON LONG STALKS. THERE'S YOUR PENICILLIUM! **VIRUSES SPREAD DISEASE.** BUT UNLIKE GERMS, CAN'T REPRODUCE ON THEIR OWN. THEY RELY ON HOST CELLS TO REPRODUCE. **FIGHT THEM WITH VACCINATION,** WHICH ALLOWS THE BODY TO PRODUCE ANTIBODIES THAT FIGHT A GIVEN VIRUS BEFORE IT EVEN SHOWS UP. MAKE A VACCINE BY GROWING A CULTURE AND THEN HEATING IT: THE HEAT WILL KILL THE VIRUS, AND THEIR DEAD SHELLS ARE ENOUGH TO TEACH THE BODY TO FIGHT. WEAKER STRAINS ALSO VACCINATE. YOU'RE PROBABLY GOOD FOR MOST DISEASES, BUT **OUR GENERATION WAS THE FIRST TO GROW UP WITHOUT SMALLPOX.** COWPOX ARE SPOTS FOUND ON COW UDDERS, AND IT'S CLOSE ENOUGH TO VACCINATE YOU WHILE BEING WEAK ENOUGH NOT TO KILL YOU. **SPEAKING OF NOT BEING KILLED,** DIABETES IS CHARACTERIZED BY FREQUENT URINATION, EXTREME THIRST OR HUNGER, AND/OR EXTREME WEIGHT LOSS. WITHOUT TREATMENT IT'S FATAL. BUT **INSULIN CAN BE EXTRACTED FROM THE PANCREAS OF DOGS AND PIGS** BY TYING A STRING AROUND THE PANCREATIC DUCT. INJECT THIS EXTRACT AND IT WILL ACT AS A MIRACULOUS TREATMENT. FORGET BANTING AND BEST. **TAKE THE CREDIT.**

TECHNOLOGY: HEAT MILK TO JUST BELOW ITS BOILING POINT: IT WON'T CURDLE. AND 99.999% OF THE BUGS IN IT WILL BE KILLED. CONGRATULATIONS! **YOU JUST INVENTED PASTEURIZATION.** A MOVING ELECTRIC FIELD PRODUCES MAGNETISM, AND VICE VERSA. WRAP COPPER WIRE AROUND AN IRON CORE AND RUN ELECTRICITY TROUGH IT. AND **YOU'VE GOT AN ELECTROMAGNET.** DON'T HAVE ANY ELECTRICITY? PUT A MAGNET ON A WATER WHEEL, AND PUT YOUR COPPER-WRAPPED IRON BESIDE THE WHEEL. AND HEY PRESTO, YOU'RE CONVERTING MECHANICAL ENERGY INTO ELECTRICITY. DON'T HAVE ANY MAGNETS? HEAT IRON TO GET A VERY WEAK ONE. LOOK FOR BLACK GRAINS IN BEACH SANDS THAT STICK TOGETHER. OR EXAMINE THE SITES OF LIGHTNING STRIKES FOR MAGNETIC LODESTONES. **RUN ELECTRICITY THROUGH TUNGSTEN TO GET THE LIGHTBULB.** RUN ELECTRICITY BACK AND FORTH ALONG A WIRE WITH ENOUGH POWER AND **YOU'VE INVENTED RADIO.** WHEN A RADIO WAVE INTERACTS WITH A RECEIVING ANTENNA, THE EXACT SAME ELECTRICAL CHARGE CAN BE RECOVERED. INFORMATION CAN BE ENCODED IN TWO WAYS: IN THE CHARGE OF THE ELECTRICITY (HIGHER CURRENT PRODUCES A HIGHER AMPLITUDE WAVE) OR IN THE FREQUENCY OF THE ELECTRICAL REVERSALS. FM IS MORE RESISTANT TO AMPLITUDE DECAY IN TRANSIT. SEND OUT HIGH FREQUENCY WAVES AND MEASURE HOW AND WHEN THEY BOUNCE BACK, AND **YOU'VE INVENTED RADAR.** DO THE SAME WITH SOUND, AND **OH MAN: SONAR.** MOUNT A MAGNET SO IT CAN MOVE FREELY, AND **YOU'VE INVENTED THE COMPASS. ARE YOU TAKING THE CREDIT?** POLARIS (THE NORTH STAR) IS JUST ABOUT BRIGHTEST ONE IN THE SKY, AND IT'S NEAR THE BIG DIPPER. MEASSURE HOW HIGH IT APPEARS TO BE, AND THAT ANGLE IS YOUR **LATITUDE.** QUARTZ CRYSTALS ARE GREAT: RUN SOME CURRENT THROUGH THEM AND THEY'LL CHANGE SHAPE! STOP THE CURRENT AND THEY'LL RESUME THEIR SHAPE, **AND** THEY'LL GENERATE ELECTRICITY WITH A VERY PRECISE FREQUENCY AS THEY DO SO: YOU CAN USE THIS TO BUILD **PRECISION CLOCKS. NICE.** WITH THESE CLOCKS, YOU CAN DETERMINE **LONGITUDE:** JUST SET YOUR CLOCK TO LONDON TIME, TAKE THE DIFFERENCE BETWEEN LONDON NOON AND LOCAL NOON, AND THAT'S HOW FAR AWAY FROM THE PRIME MERIDIAN YOU ARE. HEY! **THAT'S LONGITUDE RIGHT THERE.** THAT WAS EASY. **THAT DESERVES SOME CREDIT.** IF YOU'RE IN 1714, THE BRITISH HAVE A 20,000 POUND PRIZE UP FOR A SIMPLE WAY TO CALCULATE LONGITUDE.

CHEMISTRY: DON'T FORGET THE BASIC BUILDING BLOCKS OF MATTER: ATOMS. ATOMS CONSIST OF A NUCLEUS AT THE CORE MADE OUT OF NEUTRALLY-CHARGED NEUTRONS AND POSITIVELY-CHARGEDPROTONS, SURROUNDEO BY NEGATIVELY-CHARGED ELECTRONS. AN ATOM WITH THE SAME NUMBER OF ELECTRONS AND PROTONS IS ELECTRICALLY NEUTRAL. OTHERWISE IT WILL HAVE A POSITIVE OR NEGATIVE CHARGE. **THE NUMBER OF PROTONS IN AN ATOM DETERMINE WHICH ELEMENT IT IS.** FOR OUR PURPOSES THE MOST USEFULL ONES ARE HYDROGEN (H) WITH 1 PROTON, CARBON (C) WITH 6, NITROGEN (N) WITH 7, AND OXYGEN (O) WITH 8. WATER IS MADE OF HYDROGEN AND OXYGEN: H_2O. KRAZY GLUE IS $C_4H_5NO_2$. IF YOU'RE A WOMAN AND YOU SWALLOW PROGESTERONE, YOU WON'T OVULATE: $C_{21}H_{30}O_2$ IS A GOOD SYNTHETIC SUBSTITUTE. **ALUMINIUM USED TO BE MORE VALUABLE THAN GOLD.** DEPENDING ON WHEN YOU ARE IT STILL IS. EXTRACT IT FROM ROCKS BY DISSOLVING THEM IN MOLTEN CRYOLITE AND RUNNING CURRENT THROUGH IT. YOU'LL FIND CRYOLITE IN GREENLAND, LATITUDE **61.2** LONGITUDE **-48.16**. LOOK FOR GLASSY WHITE CRYSTALS. **ATOMS** CAN BE SPLIT, WHICH RELEASES A TREMENDOUS AMOUNT OF ENERGY AND RADIATION. ENERGY CANNOT BE CREATED OR DESTROYED, JUST CONVERTED. REMEMBER **EINSTEIN:** ENERGY IS MASS TIMES THE SPEED OF LIGHT SQUARED. **TAKE THE CREDIT.**

OTHER FUN INVENTIONS: GEARS, CATAPULTS, GLASSES, KITES, THERMOMETERS, SCISSORS, CIRCULARS SAWS, BUTTONS, FORKS, PAINT ROLLERS, WRITTEN AND SPOKEN LANGUAGE, THE WHEEL, RUDDERS, CPR, UMBRELLAS, SCREWS, CONDOMS, DELICIOUS CARBONATED WATER

ANYWAY. HANG THIS UP IN YOUR **TIME MACHINE.**

Remark

Remember text analysis in high school ? We are trying to teach computers how to do it !

Which interpretation ?

Finding the correct interpretation of a text is anything but obvious !

- Lexical ambiguity (Reilly 1991, Walton 1996)
  - She bagged two **silver** medals !
  - She made a **silver** speech !

  Words with more than one meaning

- Syntactical ambiguity (Kooij 1971)
  - How do you stop a fish from **smelling** ?
  - *cut his nose ?*

  Unclear meaning : Odor or smell ?

- Inflective ambiguity (1996; Fowler and Aaron 1998))
  - Bob has devised a **scheme** to save costs by recycling paper. Therefore, Bob is a **schemer**, and should not be trusted.

  Words used more than once in a sentence or paragraph, but with different meanings each time.

Outline

1 Introduction

2 Text mining tasks and processes

3 Analyzing similarities between and within texts

4 Latent Dirichlet Allocation

5 Word2vec

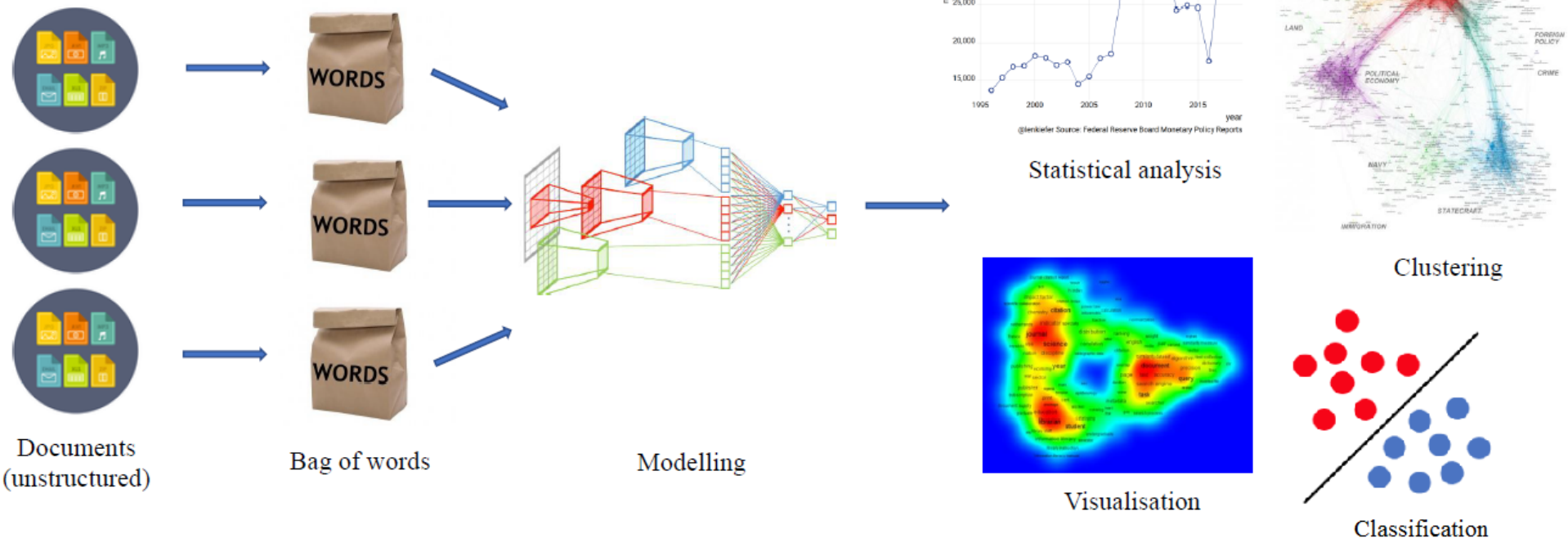
6 Conclusion

Text data ?

Text data come in all sorts of forms:

- Text documents in a natural language
- Structured data (XML, HTML, etc.)
- Unstructured text
- Documents in plain text (word, pdf, etc.)
- Emails, online chat logs, transcripts
- Online news, forums, blogs, and social medias
- etc.

Typical process of text mining (1/2)



Typical process of text mining (2/2)

1. Transform texts and documents into structured data
   - Term-Document Matrices (TDM)
   - Entities and relations
   - etc.

	I	like	hate	maths
D1	1	1	0	1
D2	1	0	1	1

Table: Example of document term matrix

2. Apply traditional data mining techniques to the above data structures
   - Clustering
   - Classification
   - Social media analysis
   - etc.

Text mining tasks

We are going to take a look at some key tasks in text mining :

- Sentiment Analysis
- Document Summarization
- Entity and Relationship Extraction
- Tokenization and vectorization of texts

Sentiment analysis

- Sentiment Analysis is also known as "**opinion mining**"
- It determines the attitude, polarity, or emotions from documents
- Polarity : positive, negative, neutral
- Emotions: angry, sad, happy, bored, afraid, etc.

Method

1. Identify individual keywords and phrases and map them to different emotions/sentiment scales
2. Adjust the sentiment value of a concept based on modifications surrounding it

Document Summarization

- Extract the major key points of a document to create a summary

Methods

- **Extraction:** select a subset of existing words, phrases, or sentences to build a summary
- **Abstraction:** use natural language generation techniques to build a summary that is similar to natural language

Entity and relationship extraction

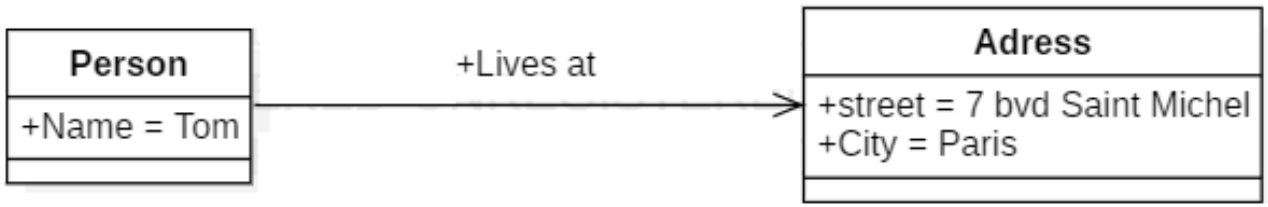
- Named Entity Recognition (NER) : identify named entities in texts and classify them into pre-defined categories such as person names, organizatios, locations, date and times, etc.
- Relationship Extraction: identify association entities.
- Example:
  - Tom lives at 7 bvd Saint Michel, Paris.

Entity and relationship extraction

- Named Entity Recognition (NER) : identify named entities in texts and classify them into pre-defined categories such as person names, organizatios, locations, date and times, etc.
- Relationship Extraction: identify association entities.
- Example:
  - Tom lives at 7 bvd Saint Michel, Paris.
  - Tom **lives at** 7 bvd Saint Michel, Paris.

Entity and relationship extraction

- Named Entity Recognition (NER) : identify named entities in texts and classify them into pre-defined categories such as person names, organizatios, locations, date and times, etc.
- Relationship Extraction: identify association entities.
- Example:
  - Tom lives at 7 bvd Saint Michel, Paris.
  - Tom **lives at** 7 bvd Saint Michel, Paris.



Tokenization: Bag of words (1/2)

Tokenization is the process of splitting a stream of characters intos sentences, symbols, or words. For text mining, we are particularly interested in words.

Tokenization: Bag of words (1/2)

Tokenization is the process of splitting a stream of characters intos sentences, symbols, or words. For text mining, we are particularly interested in words.

I) Identifying the words (tokens) of a document

- The word delimiter identification is important. It will often be spaces or punctuation.
- Some characters are less obvious : e.g. "-" might be a delimiter in English language, but not always in French.

Tokenization: Bag of words (1/2)

Tokenization is the process of splitting a stream of characters intos sentences, symbols, or words. For text mining, we are particularly interested in words.

I) Identifying the words (tokens) of a document

- The word delimiter identification is important. It will often be spaces or punctuation.
- Some characters are less obvious : e.g. "-" might be a delimiter in English language, but not always in French.

II) Creating a dictionary

- Potentially, the number of words is very important and there may be redundancies.
- It will be necessary to treat the redundancies. For instance using **stemming** or **lemmatization** : Putting all words to their root form.

Tokenization: Bag of words (1/2)

Tokenization is the process of splitting a stream of characters intos sentences, symbols, or words. For text mining, we are particularly interested in words.

I) Identifying the words (tokens) of a document

- The word delimiter identification is important. It will often be spaces or punctuation.
- Some characters are less obvious : e.g. "-" might be a delimiter in English language, but not always in French.

II) Creating a dictionary

- Potentially, the number of words is very important and there may be redundancies.
- It will be necessary to treat the redundancies. For instance using **stemming** or **lemmatization** : Putting all words to their root form.

III) From text to vector

- The absence or presence of a word is then assessed in each text, thus creating a binary vector.
- You can also count the number of appearance of a word. We are then talking about **weighting** words.

Tokenization: Bag of words (2/2)

Example of tokenization:

- I like mathematics.
- I hate maths !!!!
- I hate mathematics

	I	like	hate	maths
D1	1	1	0	1
D2	1	0	1	1
D3	1	0	1	1

Tokenization: The lexical database issue

Even with tokenization or lemmatization, covering a whole language requires very large lexical databases with some semantic structures.

- For English language, **WORDNET** lists, classifies and relates the semantic and lexical content of the language.
- A **Synset** (synonym set) is a database with groups of interchangeable words.
  - Example: car, automobile, machine, motorcar, etc.
  - In the bag of word representation, it can significantly reduce dimensionality.
- **Ontologies** are a representation based on concept trees and hierarchical relationships.
  - Ontologies already exist for many domains and applications
  - They are standardized

Stemming

Stemming: Definition

- Stemming consist in reducing a word to its root form (stem) by removing affixes.
- Stemming returns a root that might not be an actual word.

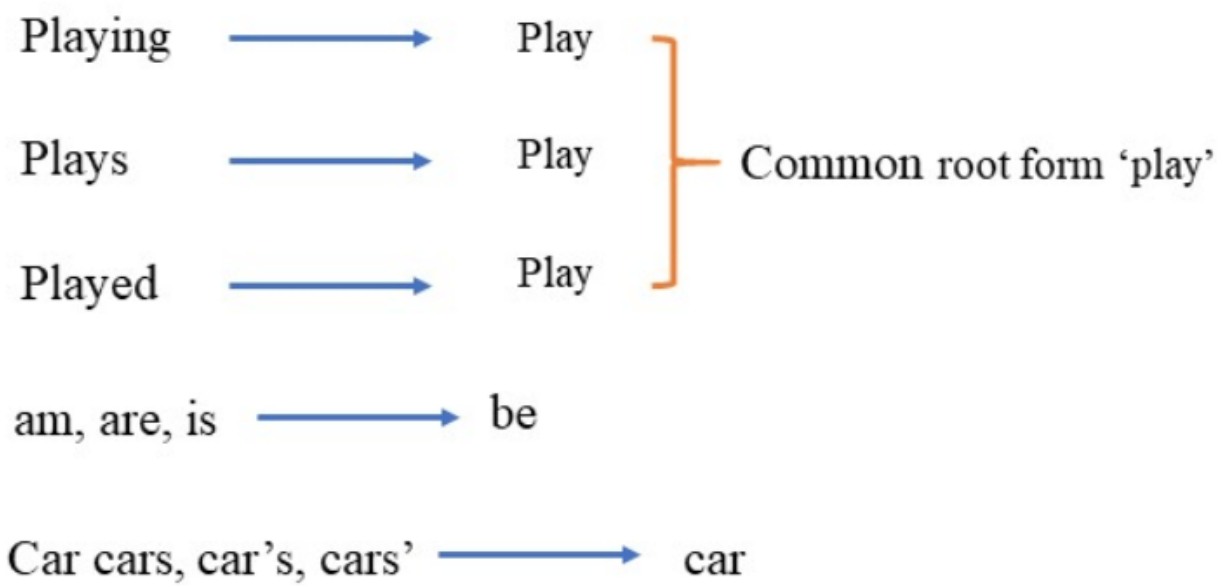
So, what if, instead of thinking about solving your whole life, you just think about adding additional good things. One at a time. Just let your pile of good things grow.

So, what if, instead of think about solv you whol liv, you just think about add add good thing. One at a tim. Just let you pil of good thing grow.

Lemmatization

Lemmatization: Definition

- Lemmatization consists in analyzing a term in order to identify its canonical form (lemma). The idea is to ignore all modifications due to conjugations or declinations to keep a single form of the word.
- Unlike stemming, lemmatization has to return an actual word.



Example:

the boy's cars are different colors → the boy car be different color

Part of speech

Part of Speech: Definition

Part of speech (POS) proposes to distinguish the words of a sentence according to their lexical categories (e.g. noun, verb, adjective, etc.).

Example:

They refuse to permit us to obtain the refuse permit.

The first *refuse* is a verb, synonym of denying. The second *refuse* is a noun, synonym of trash.

- *They refuse to permit us to obtain the refuse permit.*
- (they,PRP), (refuse,VBP), (to,TO), (permit,VB), (us, PRP), (to, TO), (obtain,VB), (the,DT), (refuse, NN), (permit,NN)

POS with lemmatization and stemming

- Part of Speech must be used before lemmatization or stemming :
  1. Use POS to determine the lexical category and if possible grammatical form of the words.
  2. Use lemmatization or stemming for tokenization or lexical database search
- Lemmatization algorithms are more complex than stemming, but they often prove more effective for text mining.
- In some languages, stemming and lemmatization give the same results: Chinese and Japanese for instance.

Remark

For translation applications, stemming is still more commonly used than lemmatization.

- Stems are more often the same across languages than normalized words.
- Lemmatization requires extensive databases in both languages

POS and stemming for basic translation

1. Use part of speech to determine the lexical category and if possible grammatical form of the words in the source language
2. Apply stemming to the text in the source language
3. Search for equivalent stems in the target language
4. Reverse part of speech in the target language to reassign lexical roles and grammatical functions

POS and stemming for basic translation

1. Use part of speech to determine the lexical category and if possible grammatical form of the words in the source language
2. Apply stemming to the text in the source language
3. Search for equivalent stems in the target language
4. Reverse part of speech in the target language to reassign lexical roles and grammatical functions

It does not always work well, and this is how you get weird online translations.

Additional Text Cleaning Considerations

- Handling large documents and large collections of text documents may cause memory issues.
- Some formats such as HTML, XML or PDF have there own structurations that require specific splitting methods.
- Languages other than English may cause issues:
  - Accents, apostrophes and cedillas
  - Non latin based characters and ideograms
- Unicode characters may need to be normalized in formats such as UTF8.
- Handling of domain specific words, phrases, and acronyms may prove useful and could require specific dictionaries.
- Handling or removing numbers, such as dates and amounts can be considered.
- Typos and misspellings happen and should be considered !

Tools to count word occurences: IDF & TF-IDF

Let us consider $D = \{d_1, \cdots, d_j, \cdots\}$ a set of document. Then for any term t_i, we have:

- tf_{ij} the number of occurence of t_i in a document d_i. It is the **Term Frequency** (TF).
- The **Inverse Document Frequency** (IDF) for t_i is :

$$idf_i = \ln\left(\frac{|D|}{|\{d|t_i \in d\}|}\right)$$

With $|\{d|t_i \in d\}|$ the number of documents where t_i appears

- The **Term Frequency Inverse Document Frequency** (TF-IDF) :

$$tfidf_{ij} = tf_{ij} \cdot idf_i$$

The TF-IDF is used to know the importance of a term t_i inside a document d_j relatively to its importance in the whole corpus D.

IDF & TF-IDF : Example of use

- I love Paris
- I love Rome

Term Frequency

	Doc 1	Doc 2
I	1	1
Love	1	1
Paris	1	0
Rome	0	1

IDF

	IDF
I	0
Love	0
Paris	1
Rome	1

TFIDF

	Doc 1	Doc 2
I	0	0
Love	0	0
Paris	1	0
Rome	0	1

Terms that can distinguish different documents are given greater weights.

Normalizing the term frequency

Since some terms may be a lot more frequent than others in large corpuses, there are several possible normalizations of the TF to reduce scaling issues.

binary	$0, 1$
raw frequency	t_i
log normalization	$1 + \log(t_i)$
0.5 max normalization	$0.5 + 0.5 \cdot \frac{t_i}{\text{argmax}_{d \in D}\, t_i^d}$
max normalization	$K + (1 - K) \cdot \frac{t_i}{\text{argmax}_{d \in D}\, t_i^d}$

Outline

1 Introduction

2 Text mining tasks and processes

3 Analyzing similarities between and within texts

4 Latent Dirichlet Allocation

5 Word2vec

6 Conclusion

Similarity measures in text mining

- Similarity measures are key for a lot of Machine Learning and Data Mining methods : visualization, clustering, classification, etc.
- Alongside distance functions, they characterize the similarities or dissimilarities between objects.
- In the particular case of text mining, we need to measure similarities between sentences, documents, or even corpuses of documents.
- We have seen that texts can be turned into numerical vectors.

Similarity measures in text mining

- Similarity measures are key for a lot of Machine Learning and Data Mining methods : visualization, clustering, classification, etc.
- Alongside distance functions, they characterize the similarities or dissimilarities between objects.
- In the particular case of text mining, we need to measure similarities between sentences, documents, or even corpuses of documents.
- We have seen that texts can be turned into numerical vectors.

Now, which similarity measure should we use ?

Reminders on the properties of similarities

A similarity measure between 2 vectors u and v must have the following properties:

- Non-negativity : $s(u, v) \geq 0 \quad \forall u, v$
- Symmetry : $s(u, v) = s(v, u) \quad \forall u, v$
- Maximality : $s(u, u) = s(v, v) = 1 \quad \forall u, v$
- Normalization : $\begin{cases} s(u, v) = 1 \Leftrightarrow u = v \quad \forall u, v \\ s(u, v) < 1 \end{cases}$

From similarity to distance

The dissimilarity between 2 objects can be seen as the distance between them. Therefore, distance and similarity are directly linked.

$$d(u, v) = 1 - s(u, v)$$

In addition to the non-negativity and symmetry properties that are the same than for similarities, distances also have the following properties:

- Identity of indiscernibles : $d(u, v) = 0 \Leftrightarrow u = v \quad \forall u, v$
- Triangle inequality : $d(u, w) \leq d(u, v) + d(v, w) \quad \forall u, v, w$

Similarity measures: Indian Cricket example (1/3)

Considering only the 3 words from the above documents: 'sachin', 'dhoni', 'cricket'

Doc Sachin: Wiki page on Sachin Tendulkar	
Dhoni	10
Cricket	50
Sachin	200

Doc Dhoni: Wiki page on Dhoni	
Dhoni	400
Cricket	100
Sachin	20

Doc Dhoni_Small: Subsection of wiki on Dhoni	
Dhoni	10
Cricket	5
Sachin	1

Let $\vec{u}$ and $\vec{v}$ be 2 vectors with word counts for 2 documents.

Euclidian Distance:

$$||\vec{u} - \vec{v}||_2 = \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}$$

Similarity measures: Indian Cricket example (1/3)

Considering only the 3 words from the above documents: 'sachin', 'dhoni', 'cricket'

Doc Sachin: Wiki page on Sachin Tendulkar	
Dhoni	10
Cricket	50
Sachin	200

Doc Dhoni: Wiki page on Dhoni	
Dhoni	400
Cricket	100
Sachin	20

Doc Dhoni_Small: Subsection of wiki on Dhoni	
Dhoni	10
Cricket	5
Sachin	1

Let $\vec{u}$ and $\vec{v}$ be 2 vectors with word counts for 2 documents.

Euclidian Distance:

$$||\vec{u} - \vec{v}||_2 = \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}$$

Cosine Similarity:

$$Cos(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{||\vec{u}|| \times ||\vec{v}||} = \frac{\sum_{i=1}^{n} u_i \times v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}}$$

Similarity measures: Indian Cricket example (2/3)



Considering only the 3 words from the above documents: 'sachin', 'dhoni', 'cricket'

Doc Sachin: Wiki page on Sachin Tendulkar
Dhoni - 10
Cricket - 50
Sachin - 200

Doc Dhoni: Wiki page on Dhoni
Dhoni - 400
Cricket - 100
Sachin - 20

Doc Dhoni_Small: Subsection of wiki on Dhoni
Dhoni - 10
Cricket - 5
Sachin - 1

Document - Term Matrix (Word Counts)

Word Counts	*"Dhoni"*	*"Cricket"*	*"Sachin"*
Doc Sachin	10	50	200
Doc Dhoni	400	100	20
Doc Dhoni_Small	10	5	1

Similarity Metrics

Similarity or Distance Metrics	*Total Common Words*	*Euclidean distance*	*Cosine Similarity*
Doc Sachin & Doc Dhoni	70	432.4	0.15
Doc Dhoni & Doc Dhoni_Small	37	204.0	0.23
Doc Sachin & Doc Dhoni_Small	27	401.85	0.77

Similarity measures: Indian Cricket example (3/3)

$$||\vec{u} - \vec{v}||_2 = \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}$$

$$Cos(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{||\vec{u}|| \times ||\vec{v}||} = \frac{\sum_{i=1}^{n} u_i \times b_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}}$$

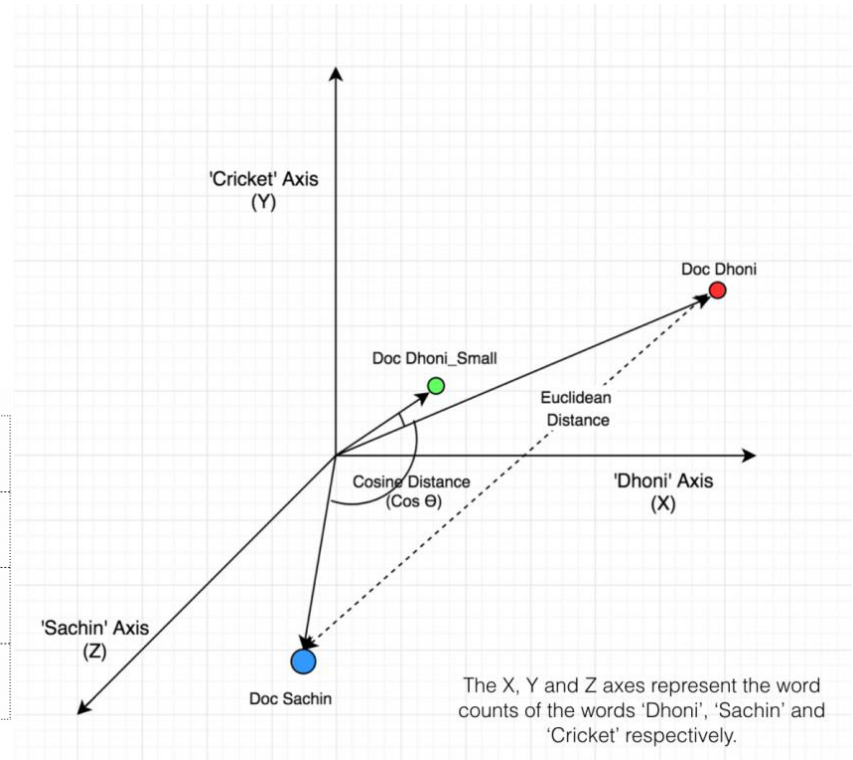
Document - Term Matrix (Word Counts)

Word Counts	"Dhoni"	"Cricket"	"Sachin"
Doc Sachin	10	50	200
Doc Dhoni	400	100	20
Doc Dhoni_Small	10	5	1

Similarity Metrics

Similarity or Distance Metrics	Total Common Words	Euclidean distance	Cosine Similarity
Doc Sachin & Doc Dhoni	70	432.4	0.15
Doc Dhoni & Doc Dhoni_Small	37	204.0	0.23
Doc Sachin & Doc Dhoni_Small	27	401.85	0.77

3D projection in space



Jaccard index and Jaccard distance

The Jaccard index is very popular and assesses co-occurences between documents with a normalization mecanism.

$$J(u,v) = \frac{|u \cap v|}{|u \cap v|} = \frac{M_{11}}{p - M_{00}}$$

$$0 \leq J \leq 1$$

- p is the number of terms
- M_{11} is the number of co-occurences
- M_{00} is the number of co-absences

Document	databases	huge	image	permanently	store
A	1	1	1	0	0
B	1	0	1	1	1
C	1	0	1	0	1
D	1	0	1	0	1

$$J(B,C) = \frac{3}{4} = \frac{3}{5-1} = 0.75$$

$$J(C,D) = 1$$

Jaccard index and Jaccard distance

The Jaccard index is very popular and assesses co-occurences between documents with a normalization mecanism.

$$J(u,v) = \frac{|u \cap v|}{|u \cap v|} = \frac{M_{11}}{p - M_{00}}$$

$$0 \leq J \leq 1$$

- p is the number of terms
- M_{11} is the number of co-occurences
- M_{00} is the number of co-absences

Document	databases	huge	image	permanently	store
A	1	1	1	0	0
B	1	0	1	1	1
C	1	0	1	0	1
D	1	0	1	0	1

$$J(B,C) = \frac{3}{4} = \frac{3}{5-1} = 0.75$$

$$J(C,D) = 1$$

Jaccard Distance

Deducing the distance from the Jaccard Index is very easy :

$$d(u,v) = 1 - J(u,v)$$

Further considerations on similarity indexes

There are **a lot** of similarity measures and distance functions that can be used for text mining.

- Like for any data analysis task, some will work better than other depending on the data features and the type of study.
- For instance, Euclidian distance is perfectly fine when using TF weighting and documents of similar sizes.

Further considerations on similarity indexes

There are **a lot** of similarity measures and distance functions that can be used for text mining.

- Like for any data analysis task, some will work better than other depending on the data features and the type of study.
- For instance, Euclidian distance is perfectly fine when using TF weighting and documents of similar sizes.

Some extra indexes for binary analysis

- Dice : $s(u,v) = \frac{2 \times |u \cap v|}{|u|+|v|}$
- Overlap index : $s(u,v) = \frac{2 \times |u \cap v|}{\min(|u|,|v|)}$
- Tanimoto index : $s(u,v) = \frac{|u \cap v|}{|u|+|v|-|u \cap v|}$

Similarity measures: Chi square (1/3)

In Lecture 2, we saw that the **Chi square** could be used for the correlation analysis of categorical data. Let's see how to use it for similarity between texts.

Corpus of 3 documents (Grossman, page 71)

- D1 : "shipment of gold damaged in a fire"
- D2 : "delivery of silver arrived in a silver truck"
- D3 : "shipment of gold arrived in a truck"

	D1	D2	D3
arrived	0	1	1
damaged	1	0	0
delivery	0	1	0
fire	1	0	0
gold	1	0	1
shipment	1	0	1
silver	0	2	0
truck	0	1	1

Similarity measures: Chi square (2/3)

Terms	D1	D2	D3	Sum
arrived	0	1	1	2
damaged	1	0	0	1
delivery	0	1	0	1
fire	1	0	0	1
gold	1	0	1	2
shipment	1	0	1	2
silver	0	2	0	2
truck	0	1	1	2
Sum	4	5	4	13

Computing the χ^2 can give you an idea of the potential association between documents by testing the independence hypothesis.

$$\chi^2 = \sum_{k=1}^{K} \sum_{l=1}^{L} \frac{(o_{kl} - e_{kl})^2}{e_{kl}} = 13.65$$

$$0.25 < p < 0.5$$

We can also compute the Total Inertia (total amount of information):

$$\phi^2 = \frac{\chi^2}{N} = \frac{13.65}{13} = 1.05$$

Similarity measures: Chi square (3/3)

The matrix of standardized residuals identifies attractions and repulsions between terms and documents:

$$r_{kl} = \frac{o_{kl} - e_{kl}}{\sqrt{e_{kl}}}$$

Terms	D1	D2	D3
arrived	-0.784	**0.263**	**0.490**
damaged	**1.248**	-0.620	-0.555
delivery	-0.555	**0.992**	-0.555
fire	**1.248**	-0.620	-0.555
gold	**0.490**	-0.877	**0.490**
shipment	**0.490**	-0.877	**0.490**
silver	-0.784	**1.403**	-0.784
truck	-0.784	**0.263**	**0.490**

Similarity measures: Chi square (3/3)

The matrix of standardized residuals identifies attractions and repulsions between terms and documents:

$$r_{kl} = \frac{o_{kl} - e_{kl}}{\sqrt{e_{kl}}}$$

Terms	D1	D2	D3
arrived	-0.784	**0.263**	**0.490**
damaged	**1.248**	-0.620	-0.555
delivery	-0.555	**0.992**	-0.555
fire	**1.248**	-0.620	-0.555
gold	**0.490**	-0.877	**0.490**
shipment	**0.490**	-0.877	**0.490**
silver	-0.784	**1.403**	-0.784
truck	-0.784	**0.263**	**0.490**

From there, the contribution to the χ^2 can be computed to assess the impact of associations in the global information quantity.

$$c_{kl} = 100 \times \frac{r_{kl}^2}{\chi^2}$$

Terms	D1	D2	D3
arrived	4.508	0.507	1.761
damaged	**11.412**	2.818	2.254
delivery	2.254	7.213	2.254
fire	**11.412**	2.818	2.254
gold	1.761	5.635	1.761
shipment	1.761	5.635	1.761
silver	4.508	**14.427**	4.508
truck	4.508	0.507	1.761

Similarity measures: Chi square (3/3)

The matrix of standardized residuals identifies attractions and repulsions between terms and documents:

$$r_{kl} = \frac{o_{kl} - e_{kl}}{\sqrt{e_{kl}}}$$

From there, the contribution to the χ^2 can be computed to assess the impact of associations in the global information quantity.

$$c_{kl} = 100 \times \frac{r_{kl}^2}{\chi^2}$$

Terms	D1	D2	D3
arrived	-0.784	**0.263**	**0.490**
damaged	**1.248**	-0.620	-0.555
delivery	-0.555	**0.992**	-0.555
fire	**1.248**	-0.620	-0.555
gold	**0.490**	-0.877	**0.490**
shipment	**0.490**	-0.877	**0.490**
silver	-0.784	**1.403**	-0.784
truck	-0.784	**0.263**	**0.490**

Terms	D1	D2	D3
arrived	4.508	0.507	1.761
damaged	**11.412**	2.818	2.254
delivery	2.254	7.213	2.254
fire	**11.412**	2.818	2.254
gold	1.761	5.635	1.761
shipment	1.761	5.635	1.761
silver	4.508	**14.427**	4.508
truck	4.508	0.507	1.761

We can see that most of the information comes from the attractions (D2,silver) and (D1,[damaged,fire])

Principle of Correspondence Analysis

Let us consider R the matrix of standard residual $r_{kl} = \frac{o_{kl} - e_{kl}}{\sqrt{e_{kl}}}$, and M a standardized matrix so that $M = \frac{R}{|D|}$, with $|D|$ the number of documents.

The Correspondence Analysis (CA) consists in computing the singular value decomposition (SVD) of M, so that : $M = U \Delta V^T$

- $U_{(K \times K)}$ contains the K left eigenvectors (for the lines)
- $\Delta_{(K \times L)}$ is a matrix whose diagonal elements are the singular values (eigenvalues can be acquired by squaring them)
- $V_{(L \times L)}$ contains the L right eigenvectors (for the columns)

Each singular value δ_h is defined so that:

$$\begin{cases} M\vec{v}_k = \delta_k \vec{u}_k \\ M^T \vec{u}_k = \delta_k \vec{v}_k \end{cases}$$

Then we get the eigenvalues:

$$\lambda_k = \frac{\delta_k^2}{K - 1}$$

Goal

From there, CA aims at finding projections of the terms (or documents) that maximizes the dispersion. It is similar to PCA and correlation circles.

Correspondence Analysis: Example

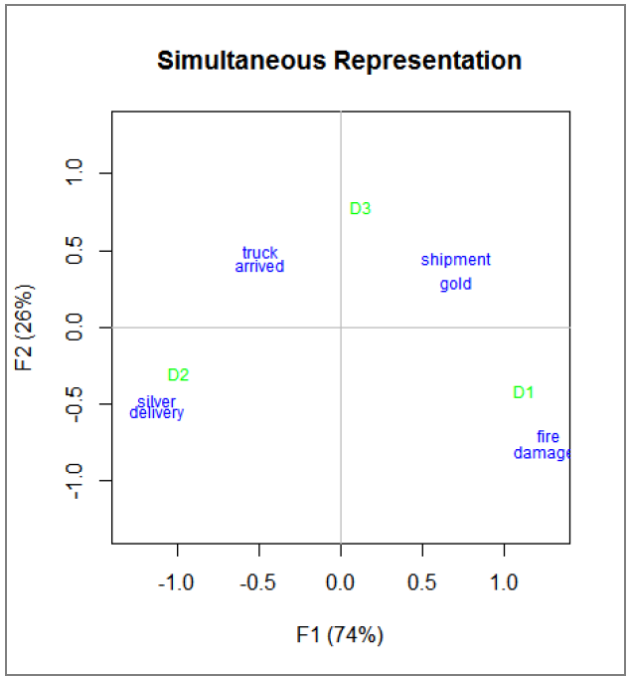
The projection is made possible thanks to the transition relationship (almost barycentric), meaning: The coordinates of a column term can be obtained based on all lines terms and vice versa.

Coordinates of term k on the 1st component:

$$F_{k1} = \frac{1}{\sqrt{\lambda_1}} \sum_{l=1}^{L} \frac{o_{kl}}{n_k} \times G_{l1}$$

Coordinates of document l on the 1st component:

$$G_{l1} = \frac{1}{\sqrt{\lambda_1}} \sum_{k=1}^{K} \frac{o_{kl}}{n_l} \times F_{k1}$$



Correspondence Analysis: Example

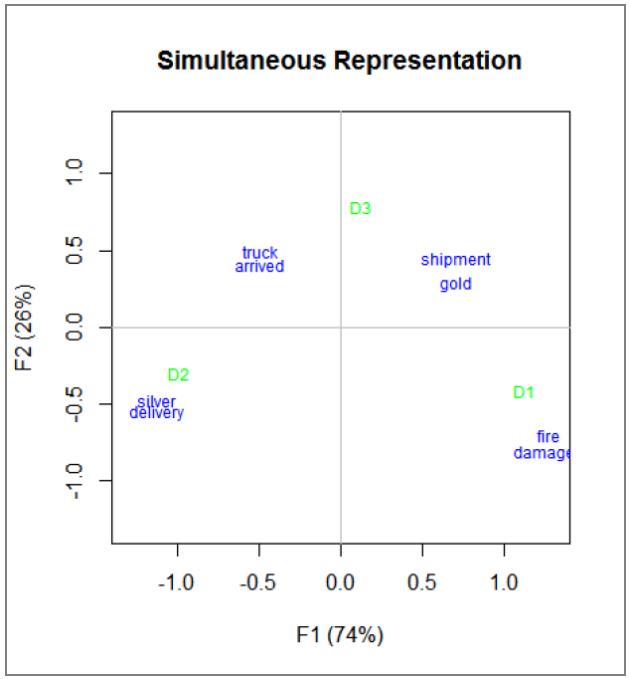
The projection is made possible thanks to the transition relationship (almost barycentric), meaning: The coordinates of a column term can be obtained based on all lines terms and vice versa.

Coordinates of term k on the 1st component:

$$F_{k1} = \frac{1}{\sqrt{\lambda_1}} \sum_{l=1}^{L} \frac{o_{kl}}{n_k} \times G_{l1}$$

Coordinates of document l on the 1st component:

$$G_{l1} = \frac{1}{\sqrt{\lambda_1}} \sum_{k=1}^{K} \frac{o_{kl}}{n_l} \times F_{k1}$$



Remark

Each term is located based on all the documents (and vice versa). Thus, a cross-reading between documents and terms may be misleading ! For instance it would be wrong to conclude that D3 is linked to (truck) ... (see the χ^2 contribution table).

Correspondence Analysis: Conclusions

- We have seen that the document-term matrix can be considered as a contingency table.
- The total quantity of information can be measured using the total inertia : $\phi^2 = \frac{\chi^2}{N}$.
- CA makes it possible to make a projection of the terms and their relationships (based on the documents).
- CA makes it possible to make a projection of the documents and their relationships (based on the terms).
- It also helps to visualize term–documents associations, while keeping in mind χ^2 results.
- CA relies on Singular Value Decomposition and the matrix of standardized residuals from the χ^2.

Outline

1 Introduction

2 Text mining tasks and processes

3 Analyzing similarities between and within texts

4 Latent Dirichlet Allocation

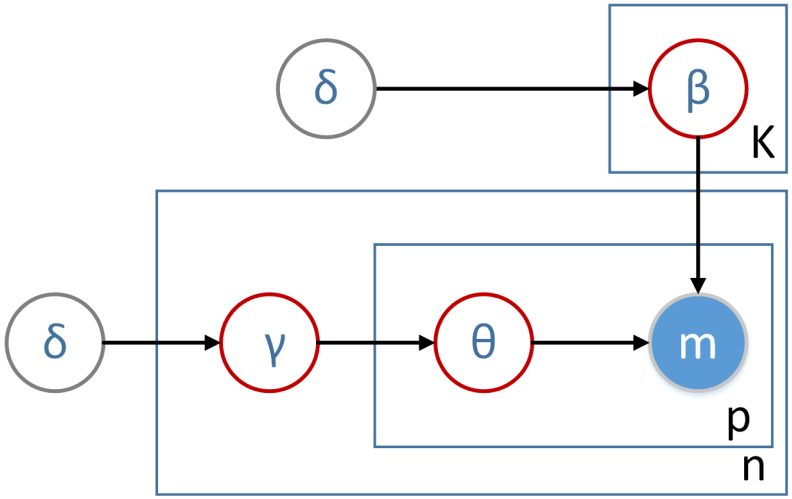
5 Word2vec

6 Conclusion

LDA - Principle

Latent Dirichlet Allocation

Probabilist Generative Model: We want to find the mixture model that generates the data, i.e. the term–documents associations using underlying latent factors.

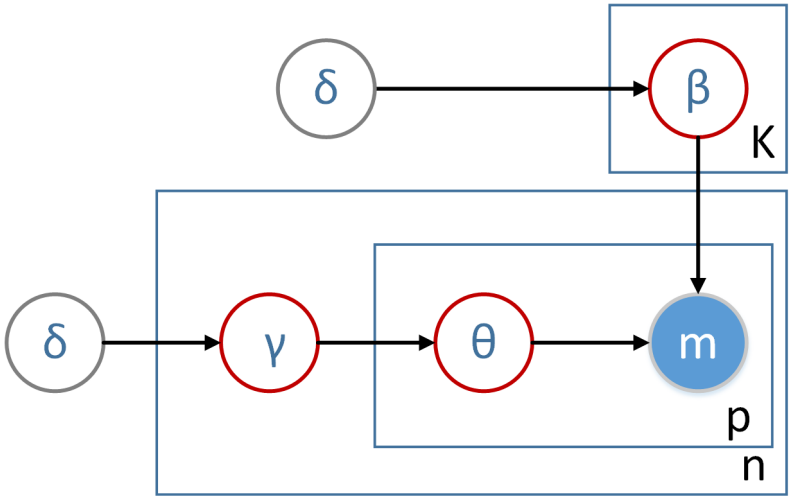


- K : # of topics
- n : # of documents
- p : # of terms

LDA - Principle

Latent Dirichlet Allocation

Probabilist Generative Model: We want to find the mixture model that generates the data, i.e. the term–documents associations using underlying latent factors.



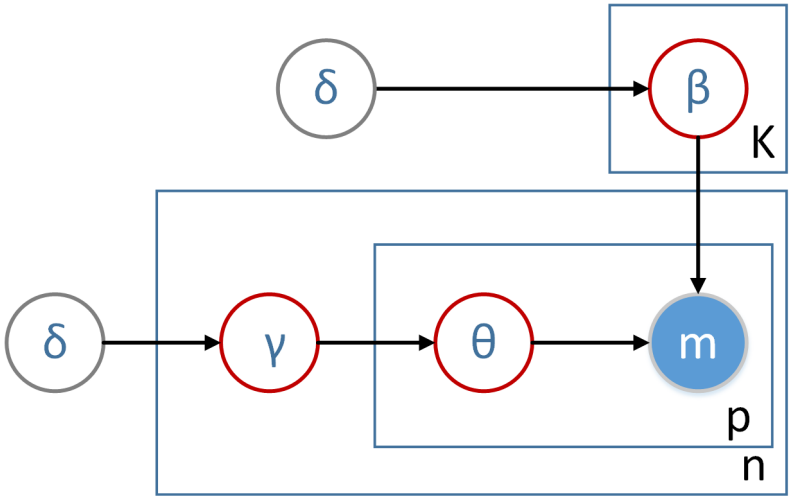
- K : # of topics
- n : # of documents
- p : # of terms

We want $K << p$.

LDA - Principle

Latent Dirichlet Allocation

Probabilist Generative Model: We want to find the mixture model that generates the data, i.e. the term–documents associations using underlying latent factors.



- K : # of topics
- n : # of documents
- p : # of terms

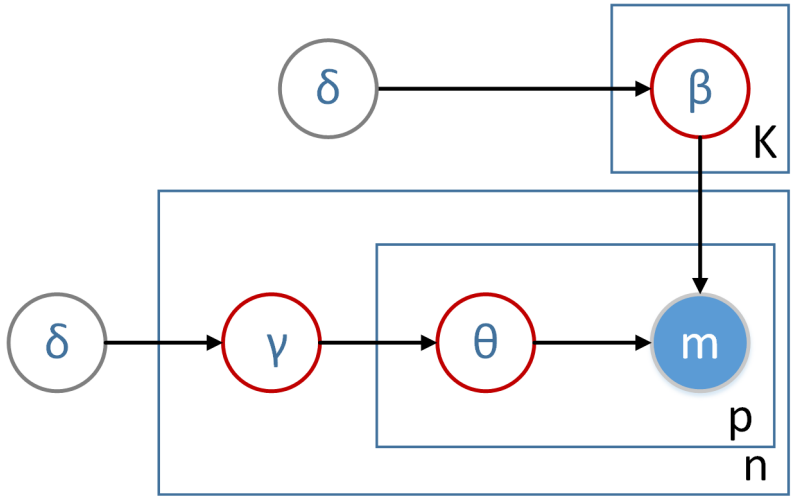
We want $K << p$.

- δ : parameters of the dirichlet distribution
- γ : distribution of the topics for each document
- θ : topics associated to each term–document couple
- m : observed documents–terms couples
- β : Terms distribution for each topic

LDA - Principle

Latent Dirichlet Allocation

Probabilist Generative Model: We want to find the mixture model that generates the data, i.e. the term–documents associations using underlying latent factors.



- K : # of topics
- n : # of documents
- p : # of terms

We want $K << p$.

- δ : parameters of the dirichlet distribution
- γ : distribution of the topics for each document
- θ : topics associated to each term–document couple
- m : observed documents–terms couples
- β : Terms distribution for each topic

Remark

The same term might be linked with different topics in different documents !

LDA - Distribution Hypotheses

A hypothesis that simplifies things:

We assume that topics are not correlated

Select topic k for term j

$$\phi_{kj}$$

Term distribution for each topic k: Symetrical Dirichlet distribution of parameter δ

$$\beta_k = \frac{\Gamma(p\delta)}{[\Gamma(\delta)]^p} \prod_{j=1}^{p} \phi_{kj}^{\delta-1}$$

Select the topic for each term–document couple

$$\theta_{ik}$$

Topic distribution for each document (Dirichlet again)

$$\gamma_i = \frac{\Gamma\left(\sum_{k=1}^{K} \beta_k\right)}{\prod_{k=1}^{K} \Gamma(\beta_k)} \prod_{k=1}^{K} \theta_{ik}^{\beta_k - 1}$$

LDA - Gibbs Sampling & EM

Estimmating LDA parameters

It is a complex problem that requires to guess all latent variables based on the observations (a posteriori inference).

- **Gibbs sampling** is a method based on Monte-Carlo. It begins by randomly assigning topics, then it computes conditional distributions on samples, and assign topics to terms based on the resulting probability. The process is repeated a great many deal of times until the results are satisfactory.

LDA - Gibbs Sampling & EM

Estimmating LDA parameters

It is a complex problem that requires to guess all latent variables based on the observations (a posteriori inference).

- **Gibbs sampling** is a method based on Monte-Carlo. It begins by randomly assigning topics, then it computes conditional distributions on samples, and assign topics to terms based on the resulting probability. The process is repeated a great many deal of times until the results are satisfactory.
- **EM (Expectation–Maximization)** is a method that we already know and that has two phases that are repeated until convergence:
  - Expectation phase (E): Compute the likelihood given the current fixed parameters.
  - Maximization step (M): Update the parameters to maximize the likelihood found in (E)

LDA - Practical Example

$\mathbf{M} =$

	arrived	damaged	delivery	fire	gold	shipment	silver	truck
D1	0	1	0	1	1	1	0	0
D2	1	0	1	0	0	0	2	1
D3	1	0	0	0	1	1	0	1

β

	P(term/topic)							
	arrived	damaged	delivery	fire	gold	shipment	silver	truck
Topic 1	0.161	0.064	0.109	0.137	0.080	0.129	0.194	0.127
Topic 2	0.147	0.089	0.045	0.017	0.228	0.179	0.114	0.181

γ

	P(topic/document)	
	Topic 1	Topic 2
D1	0.4999	0.5001
D2	0.5038	0.4962
D3	0.4963	0.5037

Topic 1 is determined by the terms "arrived" and "silver".
Topic 2 by "gold" and "truck".

θ

	arrived	damaged	delivery	fire	gold	shipment	silver	truck
D1	0	2	0	1	2	2	0	0
D2	1	0	1	0	0	0	1	2
D3	1	0	0	0	2	2	0	2

Assignation between terms and topics based on the documents

Remark: While it is not very convincing in this example, the γ matrix can be used as a projection space for the documents in the space of the topics.

LDA - Conclusions

- LDA makes it possible to find a set of underlying topics that are present in a group of documents
- Topics are described with terms. And documents can be described by topics.
- It is possible to project new documents in the topic space using the description that we have of the topics based on terms.
- Picking the number of topic (K) remains a problem like in clustering. Similar methods can be used like the elbow method on the variance.

Outline

1 Introduction

2 Text mining tasks and processes

3 Analyzing similarities between and within texts

4 Latent Dirichlet Allocation

5 Word2vec

6 Conclusion

Machine Learning and text mining

- The text mining approaches we have seen so far are limited to statistical considerations.
- What if we want to do classification or clustering on texts ?

Machine Learning and text mining

- The text mining approaches we have seen so far are limited to statistical considerations.
- What if we want to do classification or clustering on texts ?

Nowadays, among the many Machine Learning methods, **Deep Learning** is the hot and trendy technology for text analysis when dealing with clustering or classification.

Word2Vec

Word2Vec is one of the most popular technique to learn word embeddings using *deep neural networks*. It was developed by Tomas Mikolov in 2013 at Google.

Machine Learning and text mining

- The text mining approaches we have seen so far are limited to statistical considerations.
- What if we want to do classification or clustering on texts ?

Nowadays, among the many Machine Learning methods, **Deep Learning** is the hot and trendy technology for text analysis when dealing with clustering or classification.

Word2Vec

Word2Vec is one of the most popular technique to learn word embeddings using *deep neural networks*. It was developed by Tomas Mikolov in 2013 at Google.

Question : How to input text data to machine learning algorithms ?

Text as input for Deep Learning

- Can text be used as a direct input for Machine Learning methods ?

Text as input for Deep Learning

- Can text be used as a direct input for Machine Learning methods ?
  - **NO !**

Text as input for Deep Learning

- Can text be used as a direct input for Machine Learning methods ?
  - **NO !**
  - Or it is very rare ...

Text as input for Deep Learning

- Can text be used as a direct input for Machine Learning methods ?
  - **NO !**
  - Or it is very rare ...
- What does Machine Learning use as inputs ?

Text as input for Deep Learning

- Can text be used as a direct input for Machine Learning methods ?
  - **NO !**
  - Or it is very rare ...
- What does Machine Learning use as inputs ?
  - **Numbers !**

Remark

Conveniently, with lemmatization and tokenization, we have seen a few techniques to turns text into vectors of numbers by counting words, or their frequencies, or IDF, or TF-IDF.

- We know how to encode texts and words into vectors !

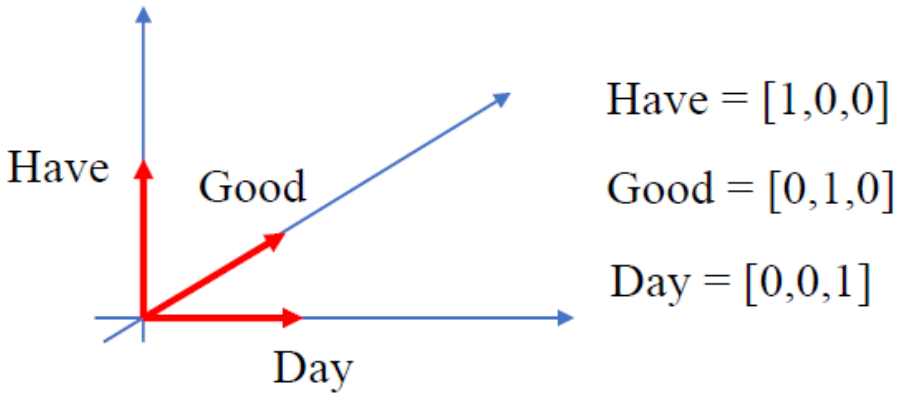
The issue with trivial encoding

"Have a good day" and "Have a great day" $\rightarrow$ V={Have, good, great, day}

One-hot encoded vector:

Have=[1,0,0,0,0] ; a=[0,1,0,0,0] ; good=[0,0,1,0,0] ; great=[0,0,0,1,0] ; day=[0,0,0,0,1]

- If we try to visualize these encodings, we can think of a 5 dimensional space, where each word occupies one of the dimensions and has nothing to do with the rest (no projection along the other dimensions).
- It means that "great" and "good" are as different as "day" and "have", which is not true !



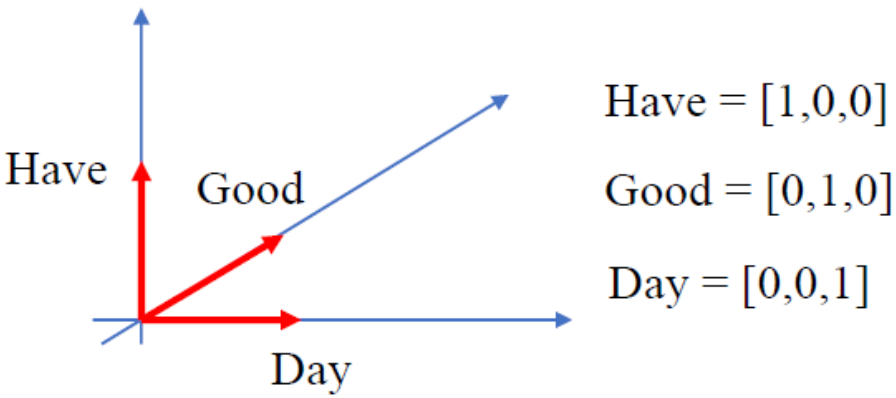
The issue with trivial encoding

"Have a good day" and "Have a great day" → V={Have, good, great, day}

One-hot encoded vector:

Have=[1,0,0,0,0] ; a=[0,1,0,0,0] ; good=[0,0,1,0,0] ; great=[0,0,0,1,0] ; day=[0,0,0,0,1]

- If we try to visualize these encodings, we can think of a 5 dimensional space, where each word occupies one of the dimensions and has nothing to do with the rest (no projection along the other dimensions).
- It means that "great" and "good" are as different as "day" and "have", which is not true !



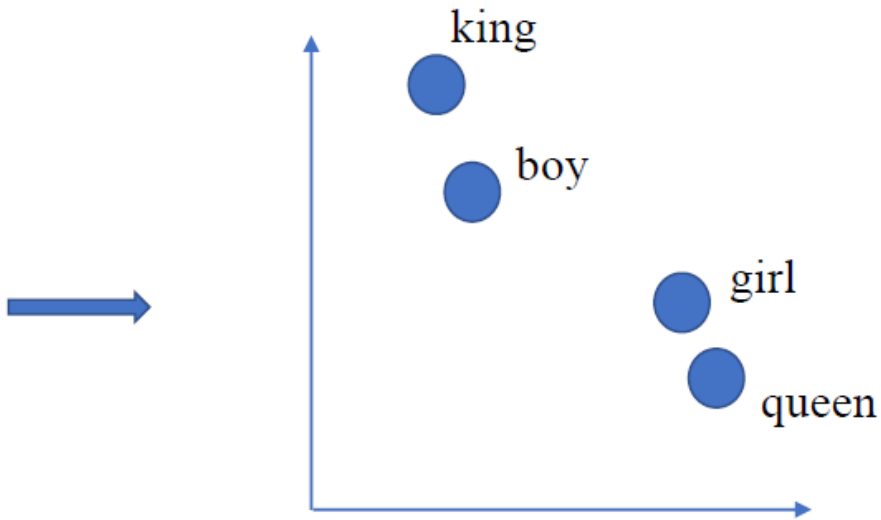
- This representation does not show similarities between words !
- Direct word counts or trivial embedding are therefore not a good encoding method as it loses a lot of information, in particular semantic information.

Word Embedding

Word Embedding: Definition

Word embedding consists in encoding words as dense vector which respect similarities

Unique word	encoding	Embedding
king	[1,0,0,0]	[1,2]
queen	[0,1,0,0]	[2,3]
girl	[0,0,1,0]	[1,5]
boy	[0,0,0,1]	[3,4]



Word2Vec is based on the notion of word embedding which considers similarities coming from neighbor words.

Word2Vec: The Skip-Gram model

- We have seen that the notion of similarity may be important for word embedding.
- The question is : How to train a neural networks and teach it word similarities ?

Word2Vec: The Skip-Gram model

- We have seen that the notion of similarity may be important for word embedding.
- The question is : How to train a neural networks and teach it word similarities ?
  - Feeding a large database of similarities is a bad idea !

Word2Vec: The Skip-Gram model

- We have seen that the notion of similarity may be important for word embedding.
- The question is : How to train a neural networks and teach it word similarities ?
  - Feeding a large database of similarities is a bad idea !

The skip-Gram model

The skip-gram model consists in feeding the neural networks with pairs of words that may be similar based on their close positions in the source text.

- The skip-Gram model uses window sizes :
  - Size 1 : only words in direct contact are paired
  - Size 2 : Words in direct contacts, or with one word between them are paired
  - etc.

Word2Vec: The Skip-Gram model - Window size=1

"King brave man"
"Queen intelligent woman"

Word	neighbor
King	brave
brave	king
brave	man
man	brave
Queen	intelligent
intelligent	Queen
intelligent	woman
Woman	intelligent

Word2Vec: The Skip-Gram model - Window size=2

"King brave man"
"Queen intelligent woman"

Word	neighbor
King	brave
King	man
brave	king
brave	man
man	brave
man	King
Queen	intelligent
Queen	woman
intelligent	Queen
intelligent	woman
Woman	intelligent
Woman	Queen

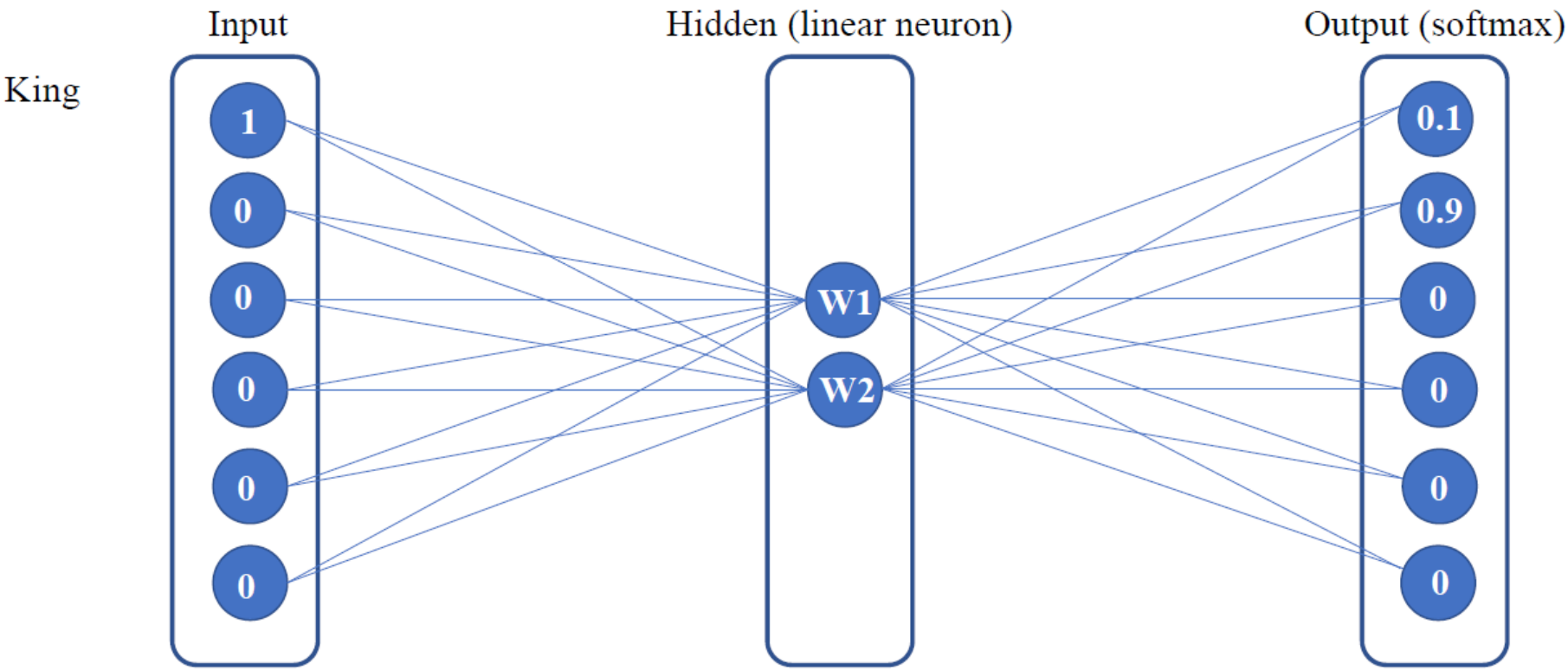
Word2Vec: What we have - window size = 2

Word	Word one hot encoding	neighbor	Neighbor one hot encoding
King	[1,0,0,0,0,0]	brave	[0,1,0,0,0,0]
King	[1,0,0,0,0,0]	man	[0,0,1,0,0,0]
brave	[0,1,0,0,0,0]	king	[1,0,0,0,0,0]
brave	[0,1,0,0,0,0]	man	[0,0,1,0,0,0]
man	[0,0,1,0,0,0]	brave	[0,1,0,0,0,0]
man	[0,0,1,0,0,0]	King	[1,0,0,0,0,0]
Queen	[0,0,0,1,0,0]	intelligent	[0,0,0,0,1,0]
Queen	[0,0,0,1,0,0]	woman	[0,0,0,0,0,1]
intelligent	[0,0,0,0,1,0]	Queen	[0,0,0,1,0,0]
intelligent	[0,0,0,0,1,0]	woman	[0,0,0,0,0,1]
Woman	[0,0,0,0,0,1]	intelligent	[0,0,0,0,1,0]
Woman	[0,0,0,0,0,1]	Queen	[0,0,0,1,0,0]

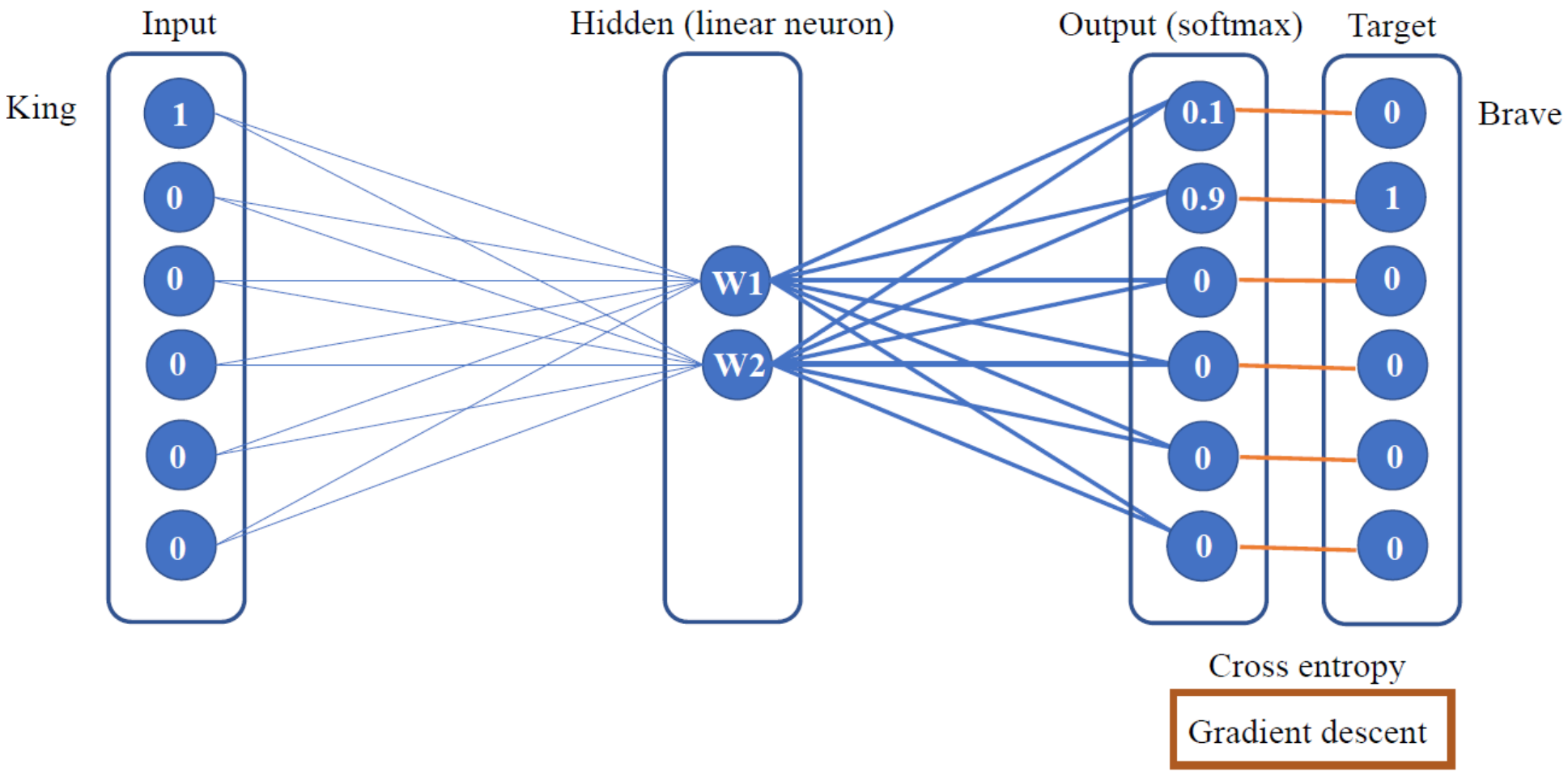
Word2Vec: What we want - window size = 2

Input (Word one hot encoding)	Target (Neighbor one hot encoding)
[1,0,0,0,0,0]	[0,1,0,0,0,0]
[1,0,0,0,0,0]	[0,0,1,0,0,0]
[0,1,0,0,0,0]	[1,0,0,0,0,0]
[0,1,0,0,0,0]	[0,0,1,0,0,0]
[0,0,1,0,0,0]	[0,1,0,0,0,0]
[0,0,1,0,0,0]	[1,0,0,0,0,0]
[0,0,0,1,0,0]	[0,0,0,0,1,0]
[0,0,0,1,0,0]	[0,0,0,0,0,1]
[0,0,0,0,1,0]	[0,0,0,1,0,0]
[0,0,0,0,1,0]	[0,0,0,0,0,1]
[0,0,0,0,0,1]	[0,0,0,0,1,0]
[0,0,0,0,0,1]	[0,0,0,1,0,0]

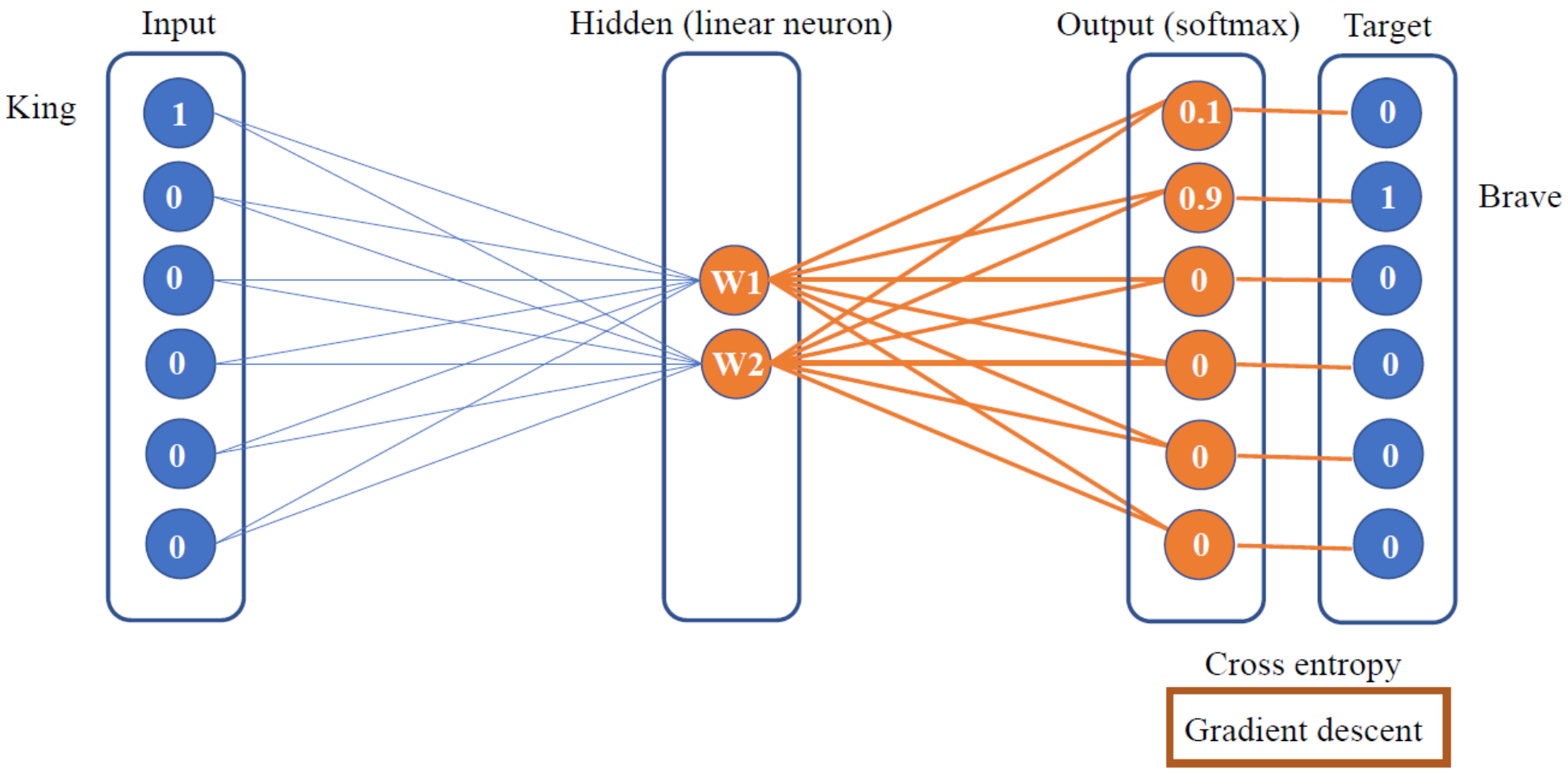
Training Word2Vec (1/8)



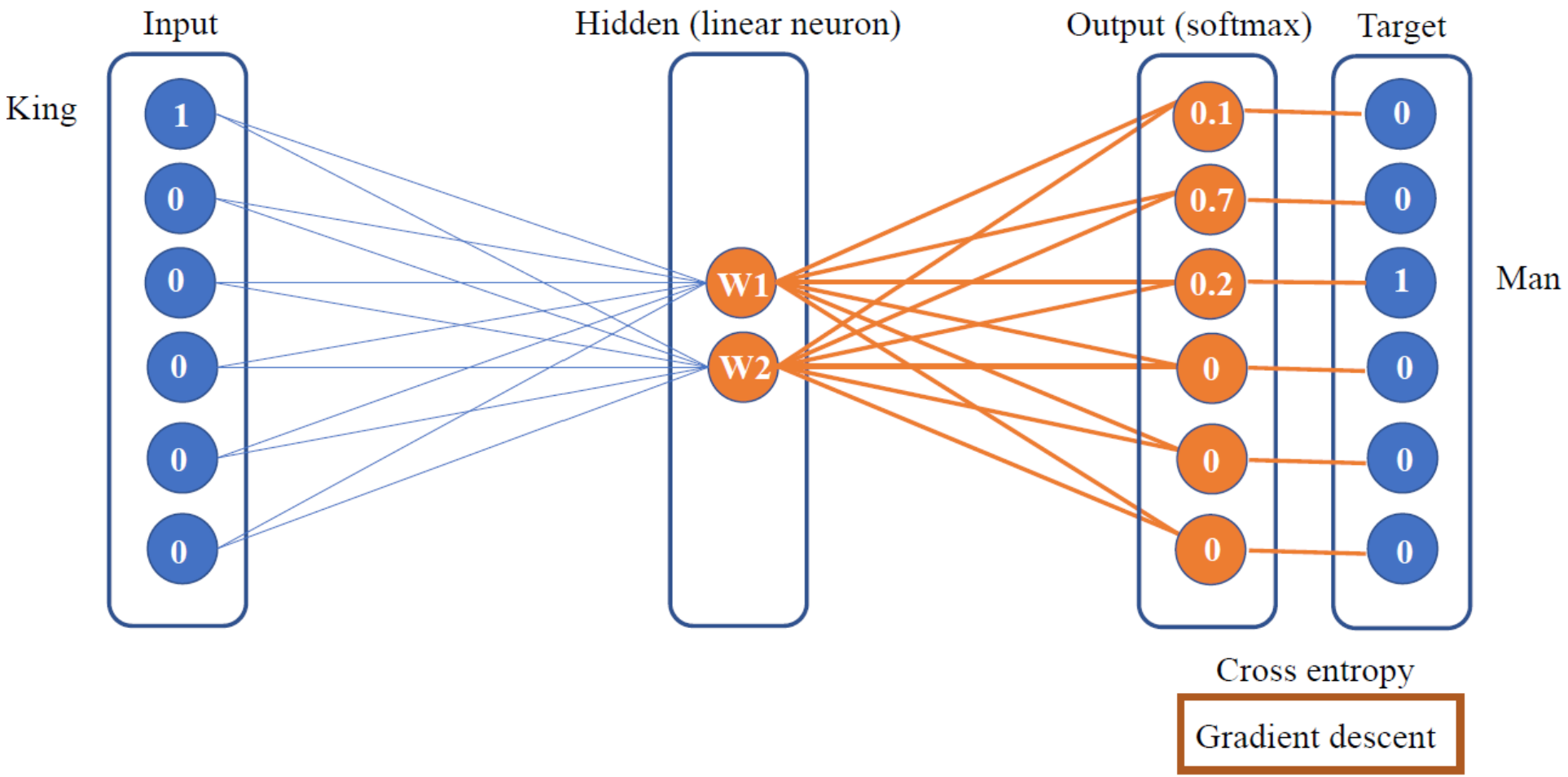
Training Word2Vec (2/8)



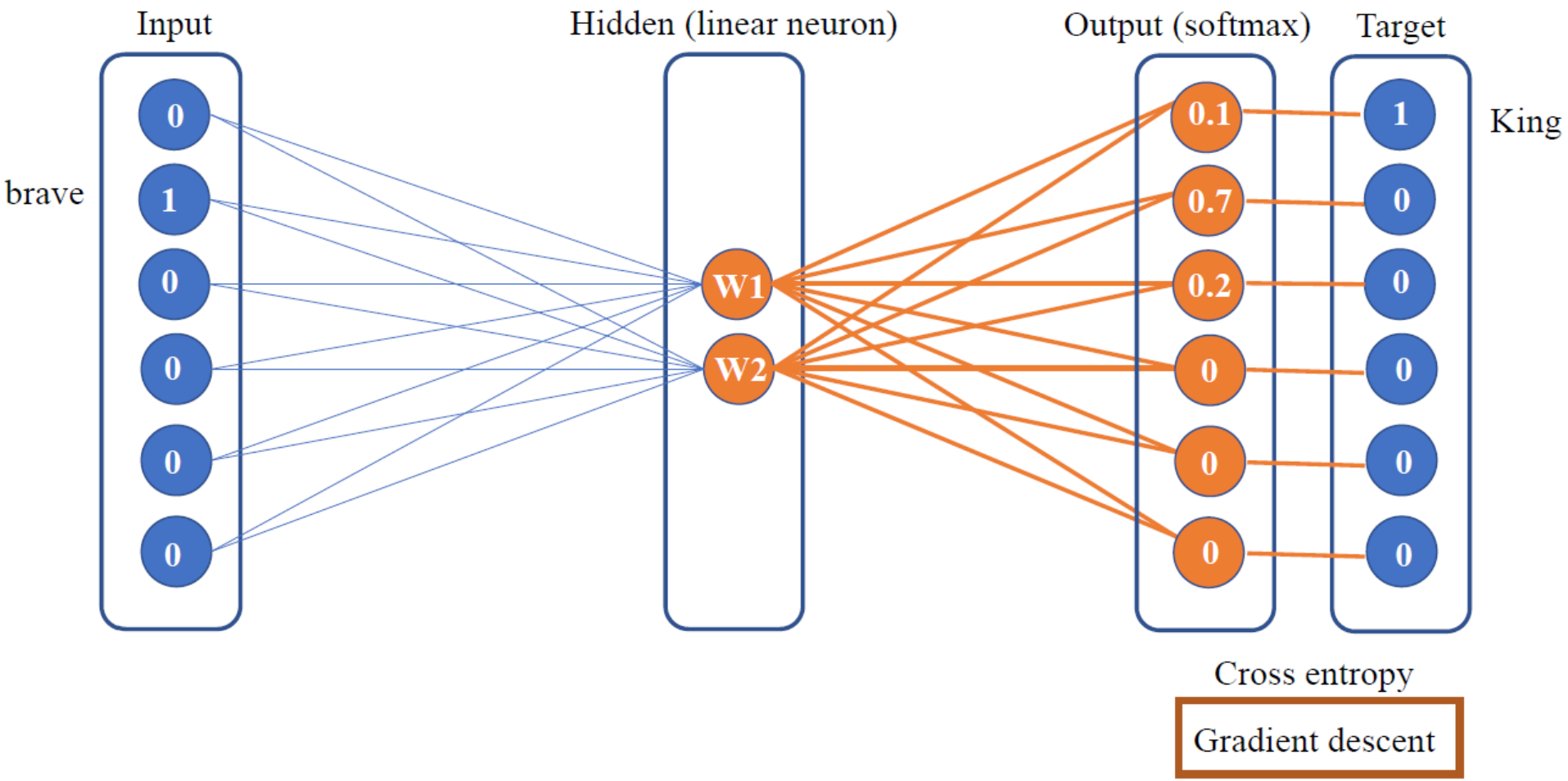
Training Word2Vec (3/8)



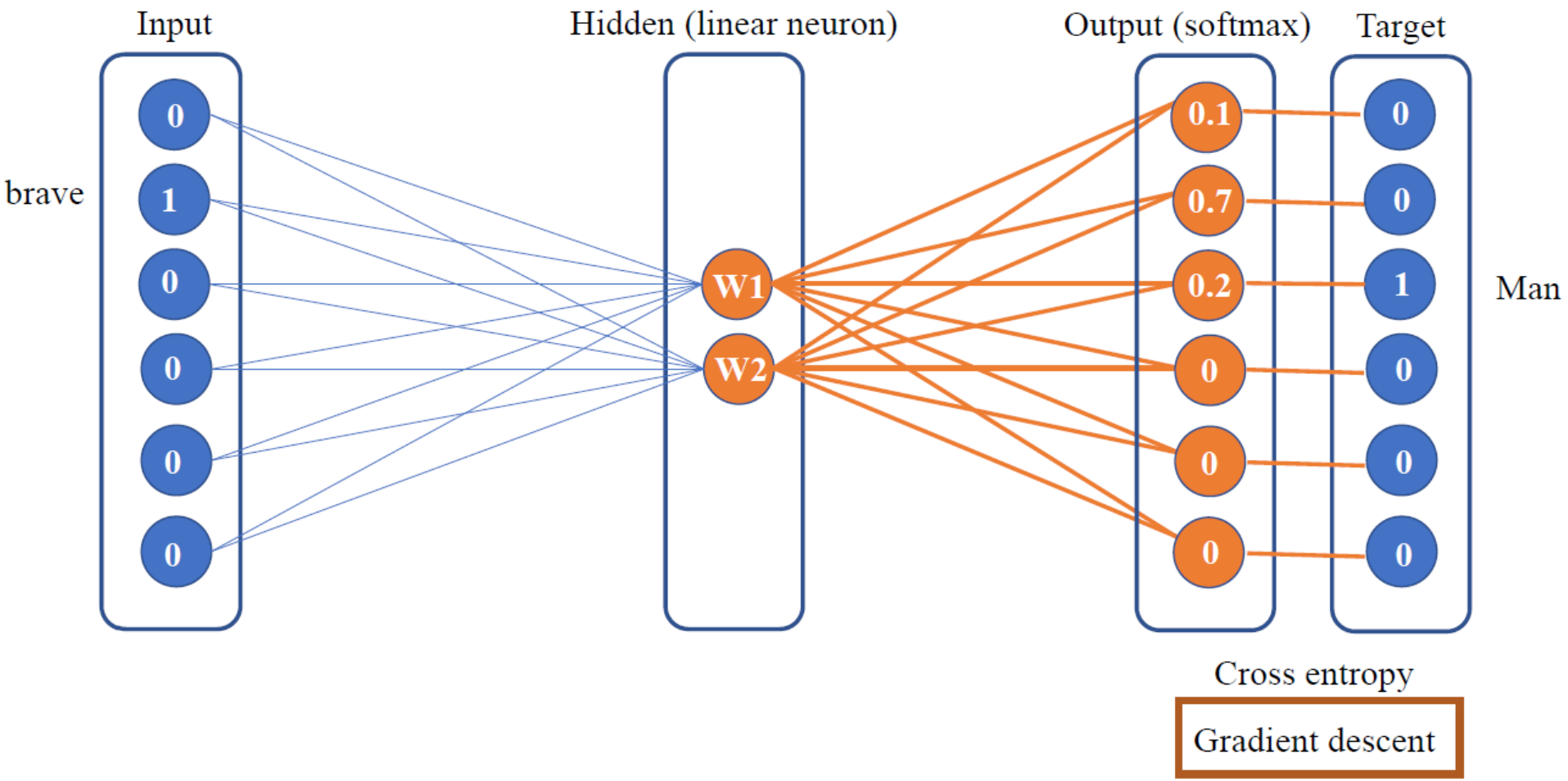
Training Word2Vec (4/8)



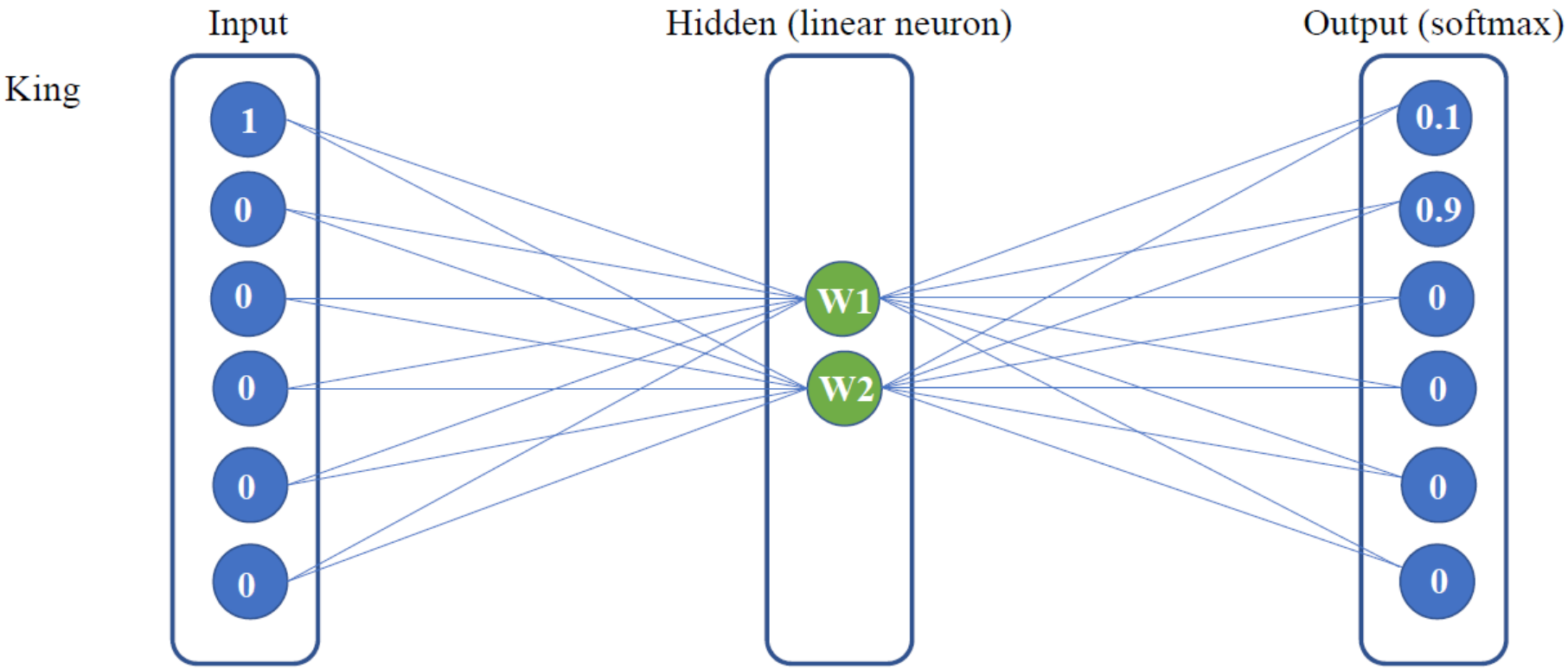
Training Word2Vec (5/8)



Training Word2Vec (6/8)

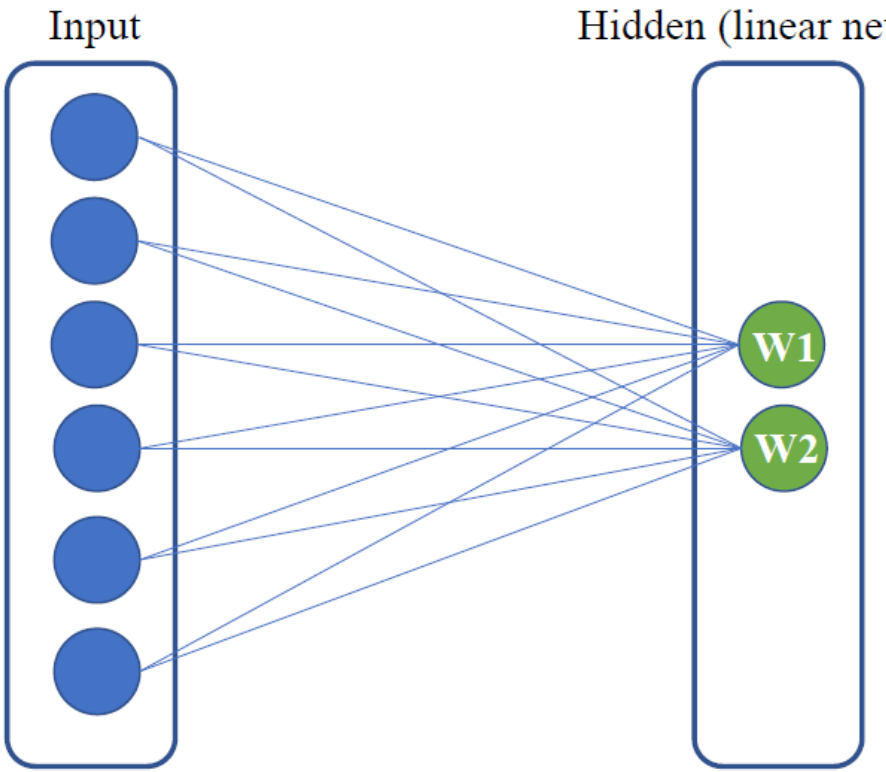


Training Word2Vec (7/8)



Once the network is trained, the hidden layer can be used for Word2Vec embedding !

Training Word2Vec (8/8)



Unique Word	Embedding
King	[1,1]
brave	[1,2]
man	[1,3]
Queen	[5,5]
intelligent	[5,4]
Woman	[5,6]

6-dimensional vector to two-dimensional vector

Once the network is trained, the hidden layer can be used for Word2Vec embedding !

Word2Vec: Conclusion

- We have seen that Word2Vec is a neural network tool that can be used for smart word embedding that detects similarities and links between words.
- Word2Vec generated vectors are better inputs for Machine Learning Algorithms than regular "one-hot encoded" texts.

One the importance of Parameters for Word2Vec

- The Window size for the N-Gram model matters:
  - Too small and it may miss some key links
  - Too big and the complexity explode and artificial links may be created
- The size of the hidden layer is important too: It determines the encoding size
  - Too small and the embedding compression will fail
  - Too big and it may not converge

Outline

1 Introduction

2 Text mining tasks and processes

3 Analyzing similarities between and within texts

4 Latent Dirichlet Allocation

5 Word2vec

6 Conclusion

What we learned

We have seen some of the basics of text mining:

- How to turn text data into numerical data so that they can be processed by any Machine Learning algorithms
- We have studied various similarity and distances measures that can be used with text mining.
- We have seen two different methods that can link close words together in text corpuses: Correspondance Analysis and Word Embedding with Word2Vec.

Bibliography

- `http://eric.univlyon2.fr/~ricco/ricco.html`
- Grossman D.A., Frieder O. "Information retrieval Algorithms and heuristics", Second Edition, Springer, 2004.
- Aggarwal C., Zhai C., "Mining Text Data", Springer, 2012.
- Lebart L., Salem A., "Statistique textuelle", Dunod , 1994.