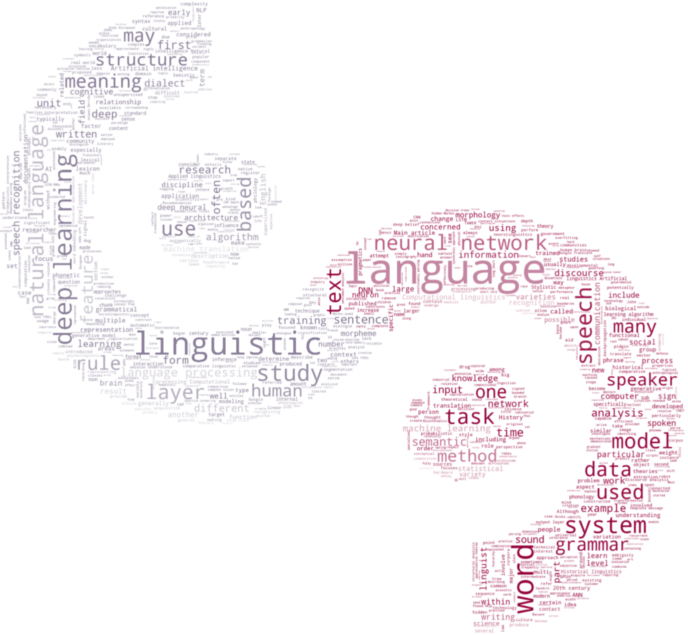

language
linguistic
deep learning
natural language
neural network
speech
system
word
grammar
data
model
used
study
human
layer
rule
task
one
method
speaker
structure
meaning
may
first
text
sentence
training
research
based
use
many
analysis
example
semantic
knowledge
input
network
time
feature
recognition
algorithm
architecture

Natural language processing

Joel Legrand
joe.legrand@centralsupelec.fr

26/02/2021

Table of Contents

General planning

- 9 lectures (1h)
- 9 practical sessions (2h)
- practical session supervised by Jeremy Fix and Joël Legrand

Evaluation

- Final exam (50%)
  - 50% theory
  - 50% practical session
- Project (50%)
  - Subject given in mid-march

Ressources

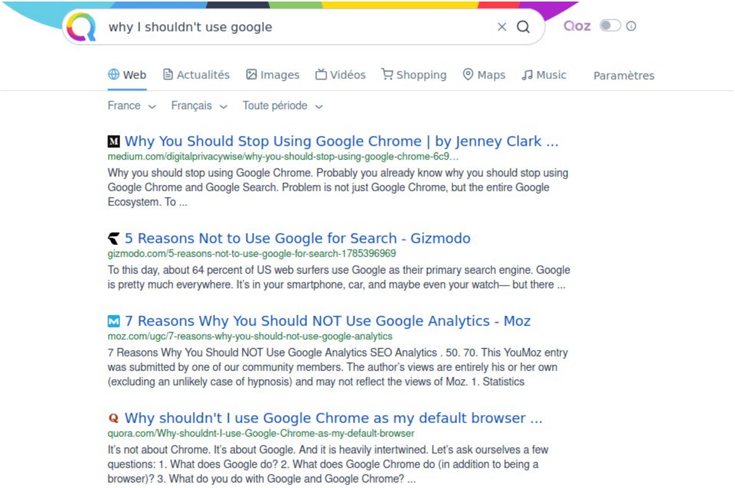
- Course Materials and practical session subjects on Edunao
- Partial correction after each practical session

Table of Contents

Table of Contents

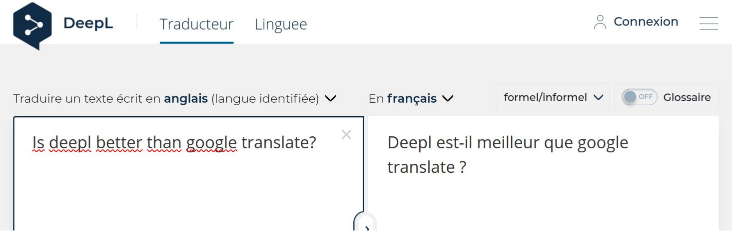
What is Natural Language Processing?

Information Retrieval



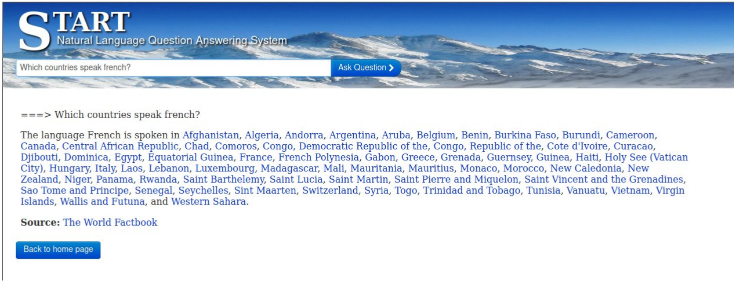
What is Natural Language Processing?

Translation



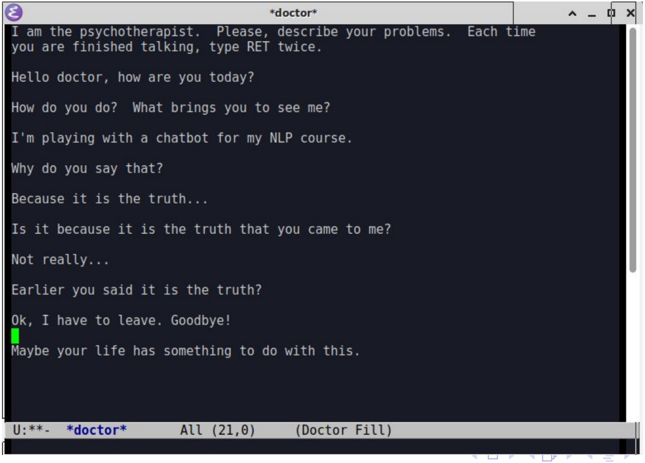
What is Natural Language Processing?

Question Answering



What is Natural Language Processing?

Chatbot



What is Natural Language Processing?

Applications

- Machine Translation
- Information Retrieval
- Question Answering
- Dialogue Systems
- Information Extraction
- Summarization
- Sentiment Analysis
- ...

Core technologies

- Language modelling
- Part-of-speech tagging
- Syntactic parsing
- Named-entity recognition
- Coreference resolution
- Word sense disambiguation
- Semantic Role Labelling
- ...

NLP lies at the intersection of **computational linguistics** and **artificial intelligence**.

Component of NLP

- Natural language understanding
  - Mapping the given input in natural language into a usefull computer-friendly representation.

    Different level of analysis can be done:
    - ★ Morphological analysis
    - ★ Syntactic analysis
    - ★ Semantic analysis
    - ★ Discourse analysis, ...

Component of NLP

- Natural language understanding
  - Mapping the given input in natural language into a usefull computer-friendly representation.

    Different level of analysis can be done:
    - Morphological analysis
    - Syntactic analysis
    - Semantic analysis
    - Discourse analysis, ...
- Natural language generation
  - Producing output in natural language from some internal representation.

    Different level of synthesis required
    - Deep planning (what to say)
    - syntactic generation, ...

Words

Words	This	is	a	simple	sentence

Example by Nathan Schneider

Morphology

Words	This	is	a	simple	sentence
Morphology		be 3sg present			

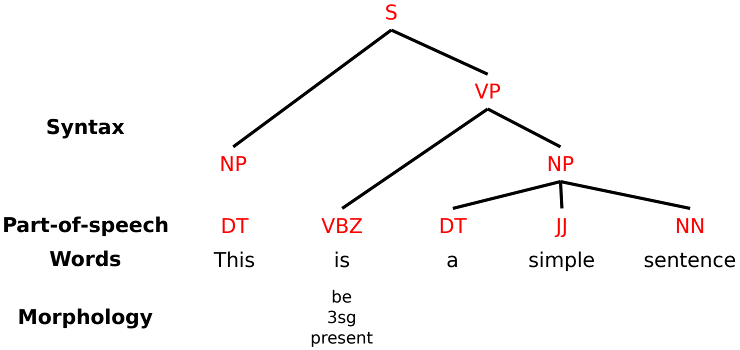
Example by Nathan Schneider

Part-of-speech

Part-of-speech	DT	VBZ	DT	JJ	NN
Words	This	is	a	simple	sentence
Morphology		be 3sg present			

Example by Nathan Schneider

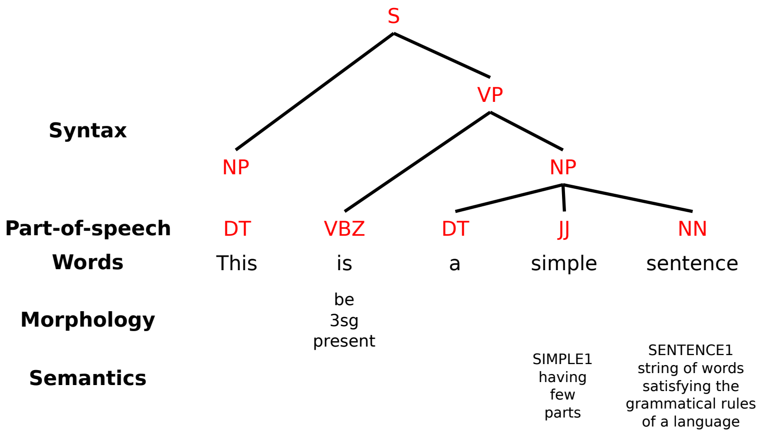
Syntax



Example by Nathan Schneider

+ Constituency parsing
+ Dependency parsing

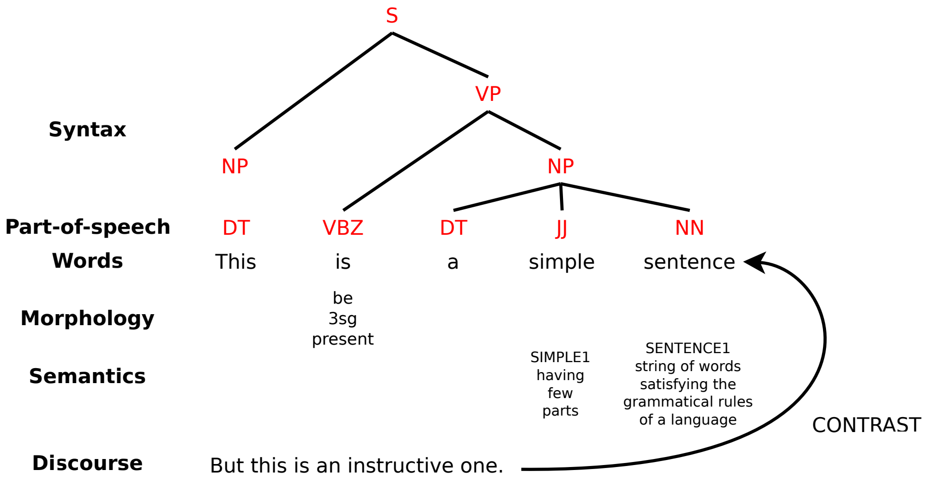
Semantics



Example by Nathan Schneider

+ Named entity recognition
+ Semantic role labelling
+ Word sense disambiguation

Discourse



Example by Nathan Schneider

+ Coreference resolution

Table of Contents

Why is NLP hard ? – Ambiguity

Ambiguity in natural language occurs at many levels:

- Word senses: bank (finance or river?)

Why is NLP hard ? – Ambiguity

Ambiguity in natural language occurs at many levels:

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)

Why is NLP hard ? – Ambiguity

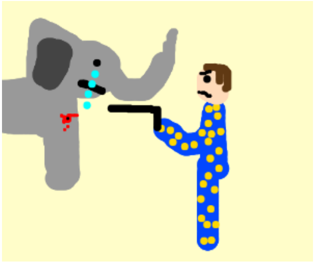
Ambiguity in natural language occurs at many levels:

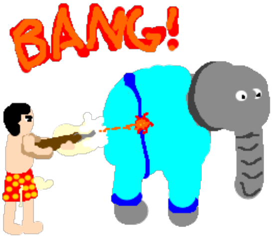
- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope

Why is NLP hard ? – Ambiguity

Ambiguity in natural language occurs at many levels:

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope
- Syntactic structure (again): One morning, I shot an elephant in my pajamas.





Why is NLP hard ? – Ambiguity

Ambiguity in natural language occurs at many levels:

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope
- Syntactic structure (again): One morning, I shot an elephant in my pajamas.
- Multiple: I made her duck

- I cooked a duck for her

Why is NLP hard ? – Ambiguity

Ambiguity in natural language occurs at many levels:

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope
- Syntactic structure (again): One morning, I shot an elephant in my pajamas.
- Multiple: I made her duck

- I cooked a duck for her
- I coocked a duck belonging to her

Why is NLP hard ? – Ambiguity

Ambiguity in natural language occurs at many levels:

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope
- Syntactic structure (again): One morning, I shot an elephant in my pajamas.
- Multiple: I made her duck

- I cooked a duck for her
- I coocked a duck belonging to her
- I crafted a rubber duck which she owns

Why is NLP hard ? – Ambiguity

Ambiguity in natural language occurs at many levels:

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope
- Syntactic structure (again): One morning, I shot an elephant in my pajamas.
- Multiple: I made her duck

- I cooked a duck for her
- I coocked a duck belonging to her
- I crafted a rubber duck which she owns
- I caused her to quickly lower her head or body

Why is NLP hard ? – Ambiguity

Ambiguity in natural language occurs at many levels:

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope
- Syntactic structure (again): One morning, I shot an elephant in my pajamas.
- Multiple: I made her duck

- I cooked a duck for her
- I coocked a duck belonging to her
- I crafted a rubber duck which she owns
- I caused her to quickly lower her head or body
- I used magic and turned her into a duck

Why is NLP hard ? – Ambiguity

Ambiguity in natural language occurs at many levels:

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope
- Syntactic structure (again): One morning, I shot an elephant in my pajamas.
- Multiple: I made her duck

- I cooked a duck for her
- I coocked a duck belonging to her
- I crafted a rubber duck which she owns
- I caused her to quickly lower her head or body
- I used magic and turned her into a duck
- ...

Why is NLP hard ? – Ambiguity

Ambiguity in natural language occurs at many levels:

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope
- Syntactic structure (again): One morning, I shot an elephant in my pajamas.
- Multiple: I made her duck
- Reference: John dropped the goblet onto the glass table and **it** broke.

Why is NLP hard ? – Ambiguity

Ambiguity in natural language occurs at many levels:

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope
- Syntactic structure (again): One morning, I shot an elephant in my pajamas.
- Multiple: I made her duck
- Reference: John dropped the goblet onto the glass table and **it** broke.
- Discourse: The meeting is cancelled. Nicholas isn't coming to the office today.

Why is NLP hard ? – Ambiguity

Ambiguity in natural language occurs at many levels:

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope
- Syntactic structure (again): One morning, I shot an elephant in my pajamas.
- Multiple: I made her duck
- Reference: John dropped the goblet onto the glass table and **it** broke.
- Discourse: The meeting is cancelled. Nicholas isn't coming to the office today.

How can we model ambiguity, and choose the correct analysis in context?

Why is NLP hard ? – Sparcity

Statistical NLP
Like most other parts of AI, NLP is dominated by statistical methods (mostly neural networks):

- Typically more robust than earlier rule-based methods.
- Relevant statistics/probabilities are **learned from data**.
- Normally requires **lots of data** about any particular phenomenon.

Why is NLP hard ? – Sparcity

- Term frequency decreases rapidly as a function of rank! (George Kingsley Zipf)
- Example: Most frequent words (word types) in the English Europarl corpus (out of 24m word tokens)

any words

Frequency	Type
1,698,599	the
849,256	of
793,731	to
640,257	and
508,560	in
407,638	that
400,467	is
394,778	a
263,040	I

nouns

Frequency	Type
124,598	European
104,325	Mr
92,195	Commission
66,781	President
62,867	Parliament
57,804	Union
53,683	report
53,547	Council
45,842	States

- But also, out of 93638 distinct word types, 36231 occur only once. Examples: cornflakes, mathematicians, fuzziness, jumbling

Why is NLP hard ? – Sparcity

Zipf's Law

- Term-frequency decreases rapidly as a function of rank
- Zipf's Law:

$$f_t = \frac{k}{r_t}$$

f_t = frequency (number of times term t occurs)

r_t = frequency-based rank of term t

k = constant (specific to the collection)

Why is NLP hard ? – Sparcity

Example for $k = 0.1$

$$f_t = \frac{0.1}{r_t}$$

- most frequent term accounts for 10% of the text

Why is NLP hard ? – Sparcity

Example for $k = 0.1$

$$f_t = \frac{0.1}{r_t}$$

- most frequent term accounts for 10% of the text
- The second most frequent term accounts for 5%

Why is NLP hard ? – Sparcity

Example for $k = 0.1$

$$f_t = \frac{0.1}{r_t}$$

- most frequent term accounts for 10% of the text
- The second most frequent term accounts for 5%
- The third most frequent term accounts for about 3%

Why is NLP hard ? – Sparcity

Example for $k = 0.1$

$$f_t = \frac{0.1}{r_t}$$

- most frequent term accounts for 10% of the text
- The second most frequent term accounts for 5%
- The third most frequent term accounts for about 3%
- Together, the top 10 account for about 30%

Why is NLP hard ? – Sparcity

Example for $k = 0.1$

$$f_t = \frac{0.1}{r_t}$$

- most frequent term accounts for 10% of the text
- The second most frequent term accounts for 5%
- The third most frequent term accounts for about 3%
- Together, the top 10 account for about 30%
- Together, the top 20 account for about 36%

Why is NLP hard ? – Sparcity

Example for $k = 0.1$

$$f_t = \frac{0.1}{r_t}$$

- most frequent term accounts for 10% of the text
- The second most frequent term accounts for 5%
- The third most frequent term accounts for about 3%
- Together, the top 10 account for about 30%
- Together, the top 20 account for about 36%
- Together, the top 50 account for about 45%

Why is NLP hard ? – Sparcity

Example for $k = 0.1$

$$f_t = \frac{0.1}{r_t}$$

- most frequent term accounts for 10% of the text
- The second most frequent term accounts for 5%
- The third most frequent term accounts for about 3%
- Together, the top 10 account for about 30%
- Together, the top 20 account for about 36%
- Together, the top 50 account for about 45%
- that's nearly half the text!

Why is NLP hard ? – Sparcity

Example for $k = 0.1$

$$f_t = \frac{0.1}{r_t}$$

- With some crafty manipulation, it also tells us that the faction of terms that occur n times is given by

$$\frac{1}{n(n+1)}$$

  - About half the terms occur only once!

Why is NLP hard ? – Sparcity

Example for $k = 0.1$

$$f_t = \frac{0.1}{r_t}$$

- With some crafty manipulation, it also tells us that the faction of terms that occur n times is given by

$$\frac{1}{n(n+1)}$$

  - About half the terms occur only once!
  - About 75% of the terms occur 3 times or less!

Why is NLP hard ? – Sparcity

Example for $k = 0.1$

$$f_t = \frac{0.1}{r_t}$$

- With some crafty manipulation, it also tells us that the faction of terms that occur n times is given by

$$\frac{1}{n(n+1)}$$

  - About half the terms occur only once!
  - About 75% of the terms occur 3 times or less!
  - About 83% of the terms occur 5 times or less!

Why is NLP hard ? – Sparcity

Example for $k = 0.1$

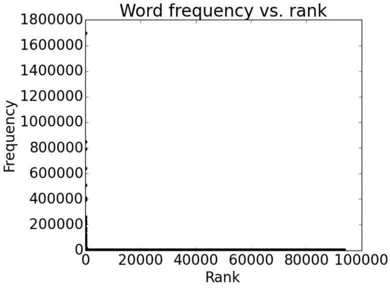
$$f_t = \frac{0.1}{r_t}$$

- With some crafty manipulation, it also tells us that the faction of terms that occur n times is given by

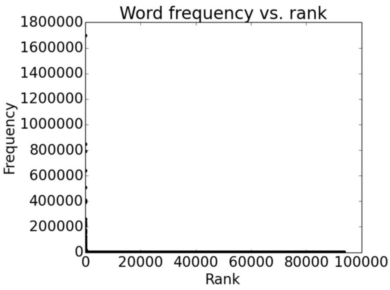
$$\frac{1}{n(n+1)}$$

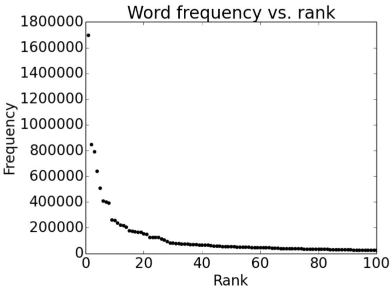
  - About half the terms occur only once!
  - About 75% of the terms occur 3 times or less!
  - About 83% of the terms occur 5 times or less!
  - About 90% of the terms occur 10 times or less!

Why is NLP hard ? – Sparcity

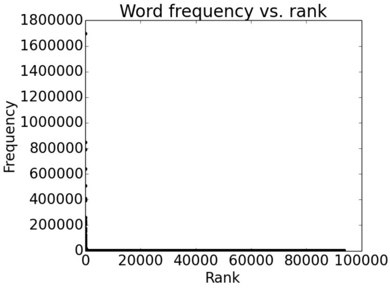


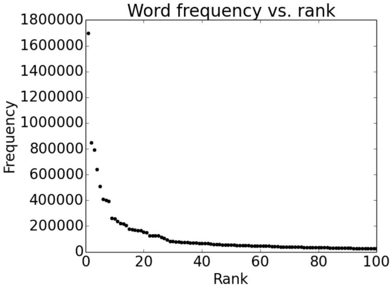
Why is NLP hard ? – Sparcity



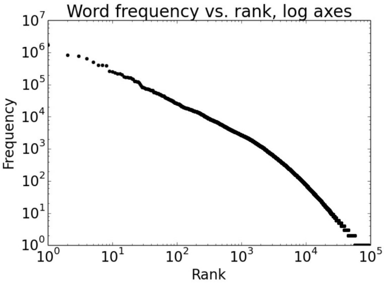


Why is NLP hard ? – Sparcity





If we use a logarithmic scale



Why is NLP hard ? – Sparcity

Implications of Zipf's Law

- The most descriptive words are those that do **not** appear in every document
  - Ignoring the most frequent terms greatly reduces the size of the index
  - The top 50 accounts for about 45% of the collection
  - **Warning**: these words can be important in combination with others (e.g., negation)
- Regardless of how large our corpus is, there will be a lot of infrequent (and zero-frequency!) words.

Why is NLP hard ? – Sparcity

Implications of Zipf's Law

- The most descriptive words are those that do **not** appear in every document
  - Ignoring the most frequent terms greatly reduces the size of the index
  - The top 50 accounts for about 45% of the collection
  - **Warning**: these words can be important in combination with others (e.g., negation)
- Regardless of how large our corpus is, there will be a lot of infrequent (and zero-frequency!) words.
  - In fact, the same holds for many other levels of linguistic structure (e.g.,syntactic rules in a CFG).

Why is NLP hard ? – Corpus variation

Suppose we train a part of speech tagger or a parser on the Wall Street Journal

ex : *Commonwealth Edison now faces an additional court-orderedrefund on its summer/winter rate differential collections that the Illinois Appellate Court has es-timated at $140 million.*

What will happen if we try to use this tagger/parser for social media?

Ex: *ikr smh he asked fir yo last name so he can add u on fb lololol*

Why is NLP hard ? – Expressivity

- Not only can one form have different meanings (ambiguity) but the same meaning can be expressed with different forms:

She gave the book to Tom	vs.	She gave Tom the book
Some kids popped by	vs.	A few children visited
Is that window still open?	vs.	Please close the window

Why is NLP hard ? – Common sense

- The correct interpretation of a sentence is often context-dependent and requires **world knowledge**.
- Very difficult to capture, since we don't even know how to represent general knowledge:
  - What is the "meaning" of a word or sentence?
  - How to model context?
  - how to model and include general knowledge?

- I dropped the glass on the floor and it broke
- I dropped the hammer on the glass table and it broke

Table of Contents

Table of Contents

Introduction

- Words of the text represent discrete, categorical features.
- How do we encode such data in a way which is ready to be used by the algorithms?
- The mapping from textual data to real valued vectors is called feature extraction.
- Many methods exists, from the simplest to more sofisticated ones:
  - Bags-of-words
  - TF-IDF
  - ...
  - Continuous vectorial representations (word embeddings)

Bag of words representation (BoW)

Example

Sentence:

It is the best of the best

Bag of words:

it	is	the	best	of	a	an	...
1	1	2	2	1	0	0	...

- Idea: represent a document by the occurrence counts (or the frequency) of each word
- Drawback: ordering of words is lost
  - *John is quicker than Mary* and *Mary is quicker than John* have the same vectors

Bags of Ngrams

Example

Sentence:

It is the best of the best

Bag of words:

<start>	it is	is the	the best	best of	of the	best <end>	a best	...
1	1	1	2	1	1	1	0	...

- Idea: represent a document by the occurrence counts (or the frequency) of each Ngram
- Drawback:
  - The ordering of Ngrams is not considered
  - The more you increase the Ngram size, the more the data become sparce

Limitations of BoW

- **Vocabulary**

  The vocabulary requires careful design, most specifically in order to manage the size, which impacts the sparsity of the document representations.

Limitations of BoW

- **Vocabulary**

  The vocabulary requires careful design, most specifically in order to manage the size, which impacts the sparsity of the document representations.

- **Sparsity**

  Sparse representations are harder to model both for computational reasons (space and time complexity) and also for information reasons

  - the challenge is for the models to harness so little information in such a large representational space
  - the more you refine the features, the more data you need to train

Limitations of BoW

- **Vocabulary**

  The vocabulary requires careful design, most specifically in order to manage the size, which impacts the sparsity of the document representations.

- **Sparsity**

  Sparse representations are harder to model both for computational reasons (space and time complexity) and also for information reasons

  - the challenge is for the models to harness so little information in such a large representational space
  - the more you refine the features, the more data you need to train

- **Meaning**

  Discarding word order ignores the context, and in turn meaning of words in the document (semantics).

TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) tackle a main limitation of BoW:

- highly frequent words tend to dominate in the document
- but may not contain as much "informational content" to the model as rarer but perhaps domain specific words.

TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) tackle a main limitation of BoW:

- highly frequent words tend to dominate in the document
- but may not contain as much "informational content" to the model as rarer but perhaps domain specific words.

TF-IDF rescale the frequency of words by how often they appear in all documents

- **Term Frequency**: is a scoring of the frequency of the word in the current document.
- **Inverse Document Frequency**: is a scoring of how rare the word is across documents.

Intuition: The more frequent a term is within a corpus, the less discriminating it is. TF-IDF increases the relevance of a given term according to its rarity within the corpus.

TF-IDF

$$\text{TF-IDF} = TF * IDF$$

where

$$TF(term) = \frac{\text{Number of times the term appears in document}}{\text{total number of terms in the document}}$$

and

$$IDF(term) = log(\frac{\text{total number of documents}}{\text{Number of documents with term in it}})$$

TF-IDF

Example: TF-IDF for word “qui”

Document 1	Document 2	Document 3
Son nom est célébré par le bocage **qui** frémit, et par le ruisseau **qui** murmure, les vents l'emportent jusqu'à l'arc céleste, l'arc de grâce et de consolation que sa main tendit dans les nuages.	À peine distinguait-on deux buts à l'extrémité de la carrière : des chênes ombrageaient l'un, autour de l'autre des palmiers se dessinaient dans l'éclat du soir.	Ah ! le beau temps de mes travaux poétiques ! les beaux jours que j'ai passés près de toi ! Les premiers, inépuisables de joie, de paix et de liberté ; les derniers, empreints d'une mélancolie **qui** eut bien aussi ses charmes.

$$TF_{\text{qui, document 1}} = \frac{2}{38}$$

$$IDF_{\text{qui}} = log(\frac{2}{3})$$

$$\text{TF-IDF}_{\text{qui, document 1}} = \frac{2}{38} \cdot log(\frac{2}{3}) = 0.0092$$

BoW and TF-IDF for query mapping

The proximity between a query q and a document d can be obtained by computing the cosine similarity between their respective vectorial representations (v_q, v_d).

$$\text{cosine\_similarity}(v_q, v_d) = \frac{v_q . v_d}{||v_q||\;||v_d||}$$

The higher the cosine similarity will be, the better the document will correspond to the query (hopefully).

Table of Contents

Word representation - One-hot encoding

One-hot encoding:

- Vector size: the size of the dictionnary
- Assign an integer index to each word
- Vector of "0" except a single "1" at the position corresponding to the word's index
- Ex: the $\rightarrow$ (1, 0, 0, 0, ...)
  cat $\rightarrow$ (0, 1, 0, 0, ...)

Word representation - One-hot encoding

One-hot encoding:

- Vector size: the size of the dictionnary
- Assign an integer index to each word
- Vector of "0" except a single "1" at the position corresponding to the word's index
- Ex: the $\rightarrow$ (1, 0, 0, 0, ...)
  cat $\rightarrow$ (0, 1, 0, 0, ...)

Problem with one-hot vectors:

- Similarity issue: ideally we would want similar words like "cat" and "tiger" to have similar features.
- Vocabulary size (the dictionnary can be large)
- Sparcity: many machine learning models won't work well with very high dimensional and sparse features.

Word embeddings

Word embeddings:

- Mapping words to vectors of real numbers.
- Dense, semantically-meaningful representation

Word embeddings

Word embeddings:

- Mapping words to vectors of real numbers.
- Dense, semantically-meaningful representation

Intuitions:

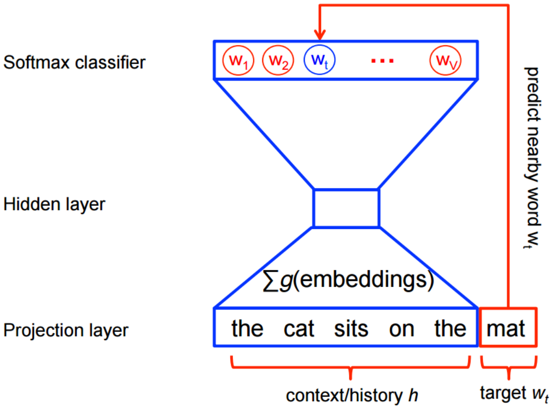
- Words appearing in similar contexts tend to have similar meaning (Harris,1954)
- "You shall know a word by the company it keep", (Firth, 1957)

Matrix factorization methods

LSA: Define co-occurence matrix of words and apply SVD to reduce dimensionality

See labwork n°1

NN based word embeddings - language modeling



Idea: Given a window of text (context), predict the next word. This task is called **language modeling** (see course 2).

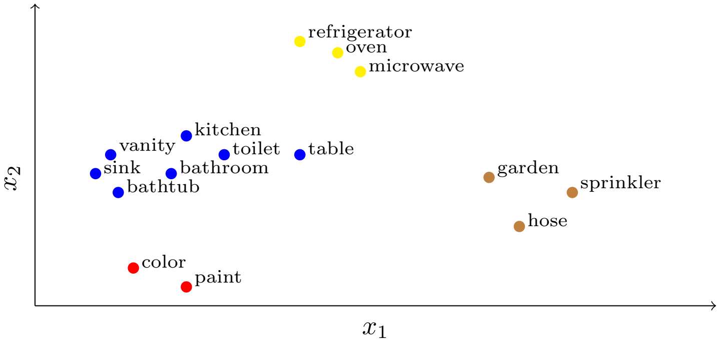
Advantage of NN-based word embeddings

- Continuous vector representations
- Low dimension (compared to dictionary size)
- Pre-trained on large unlabeled data

FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

Figure: Closest neighbors in term of Euclidean distance (from Collobert et al., 2011)

Visualization of word embeddings



Word embeddings generalities

- We can get embeddings implicitly from any task that involves words
- However, good generic embeddings are good for other tasks whish may have less training data (transfert learning)
- Embeddings can be fine-tuned to the final task