Semantics in Natural Language Processing

PhD Programme in Computer Science and Mathematics
Courses 2021-2022

Pierpaolo Basile, pierpaolo.basile@uniba.it
Marco Polignano, marco.polignano@uniba.it

Natural Language

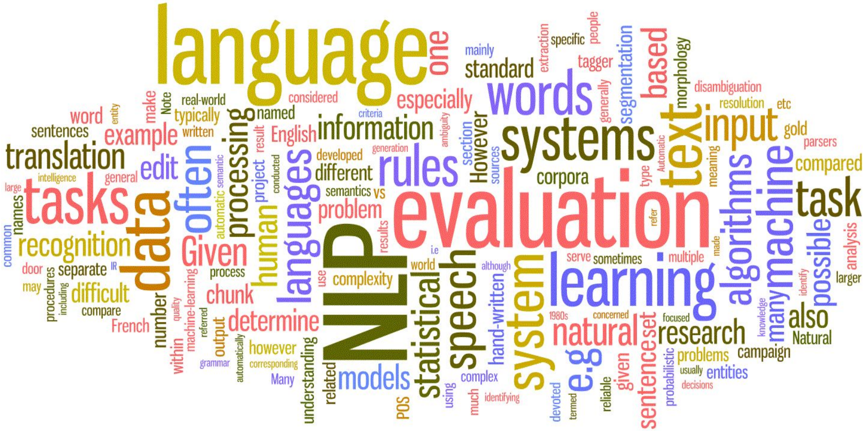
Refers to the language spoken by people, e.g. English, Japanese, Swahili, Italian, as opposed to artificial languages, like C++, Java, etc.

…Processing

Applications that deal with natural language in a way or another

NLP Applications

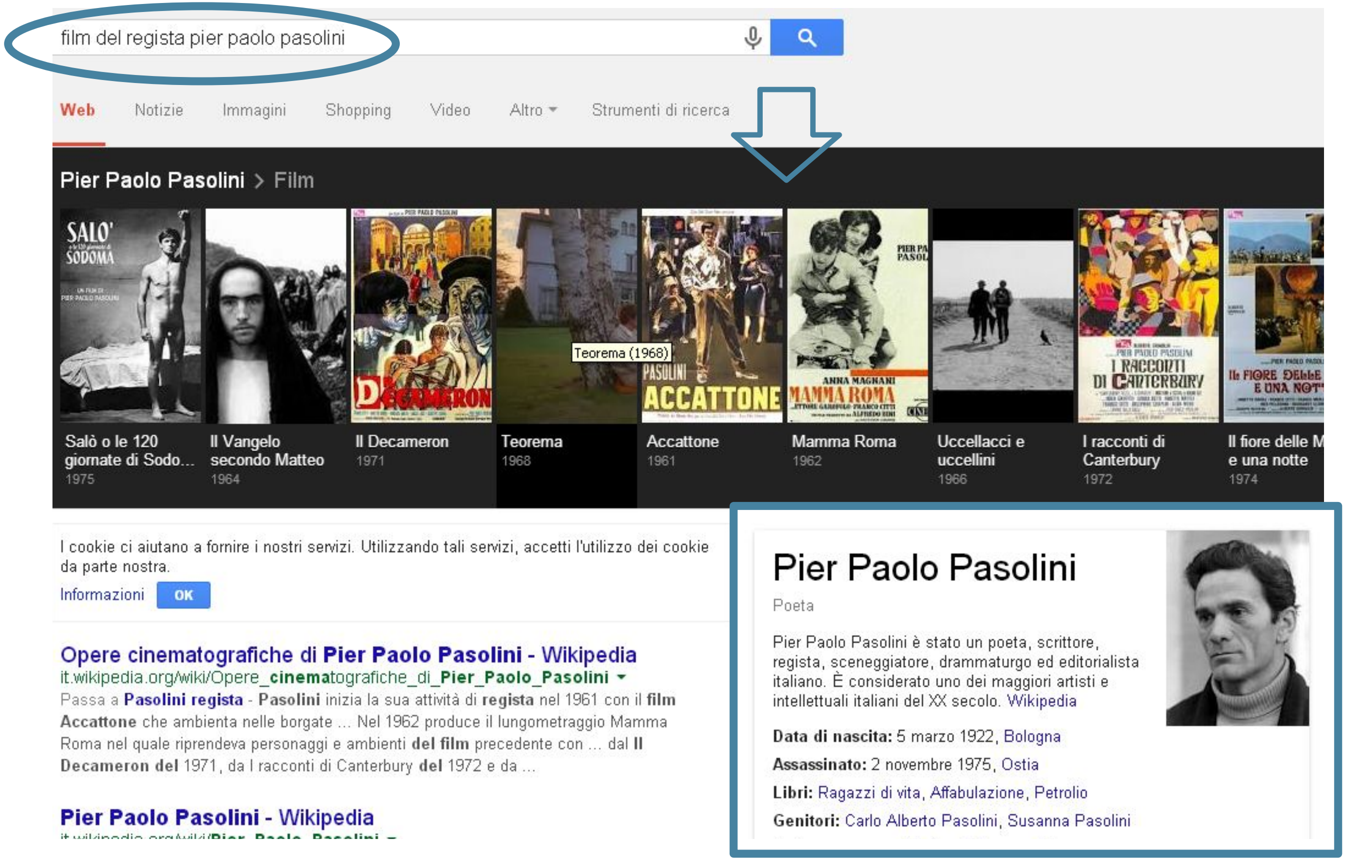
- Classify text into categories
- Index and search large texts
- Automatic translation
- Speech understanding
- Information extraction
- Automatic summarization
- Question answering
- Knowledge acquisition
- Text generations / dialogues



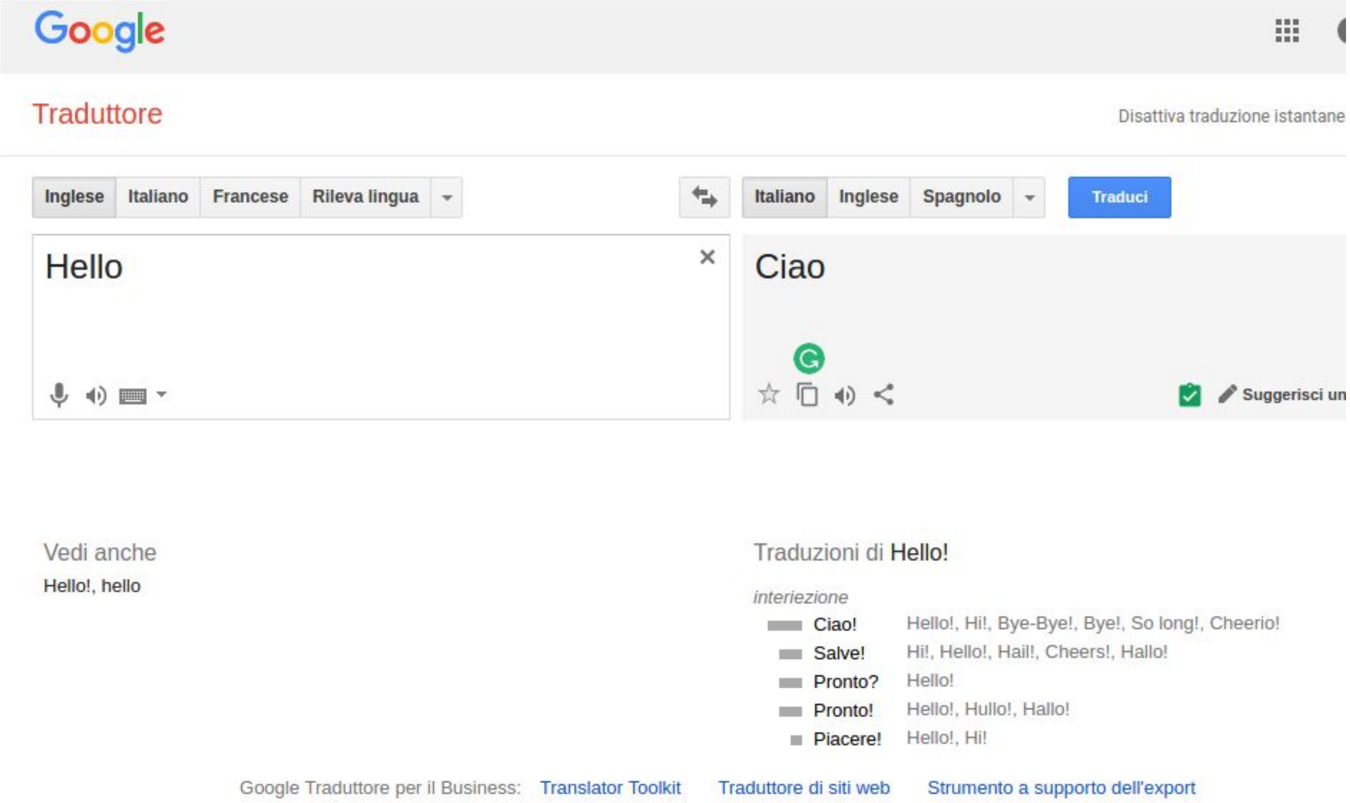
Why NLP?

- Google, Yahoo!, Bing (3,37%), Baidu (0,79%) -> **Information Retrieval**
- LinkedIn -> **Information Extraction + Information Retrieval**
- Google Translate, Babelfish, Systran -> **Machine Translation**
- Ask, IBM Watson -> **Question Answering**
- Myspace, Facebook, Twitter -> **Social Networks, Processing of User-Generated Content**
- All "Big Guys" have (several) strong NLP research labs: IBM, Microsoft, AT&T, Xerox, ORACLE-Sun Microsystems, etc.
- Academia: research in a university environment

NLP Applications: Search



NLP Applications: Machine Translation



Distributional Semantic Models

What’s Tezguno?

A bottle of Tezguno is on the table.

Everyone likes Tezguno.

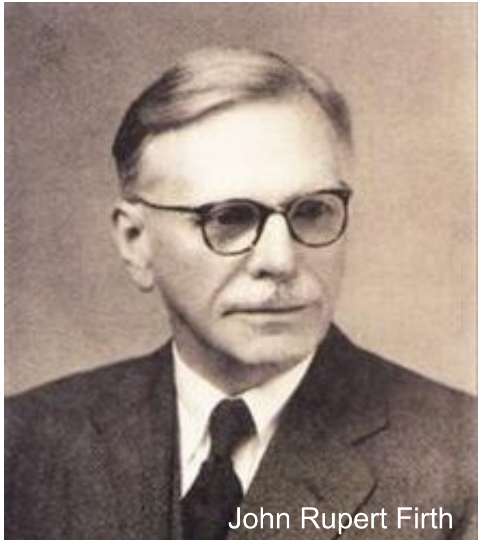
Tezguno makes you drunk.

We make Tezguno out of corn.

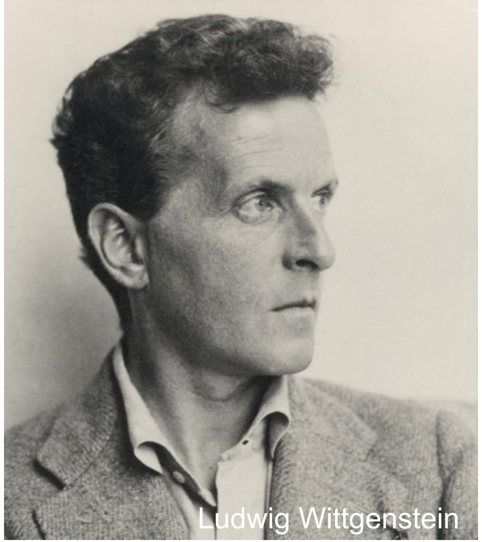
What's Tezguno?

Tezguno

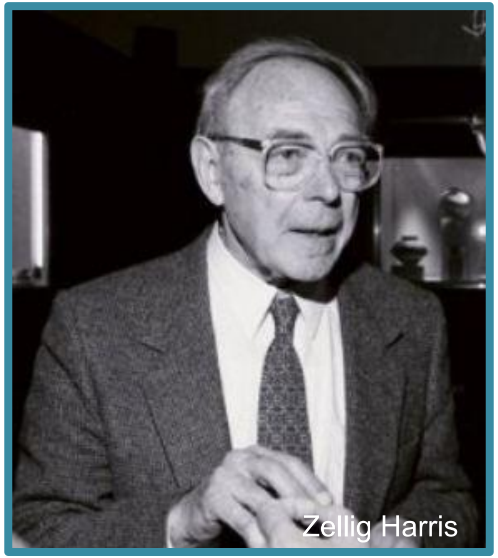
Distributional
Semantic Models (DSM)



You shall know a word by the company it keeps!



The meaning of a word is its use in the language

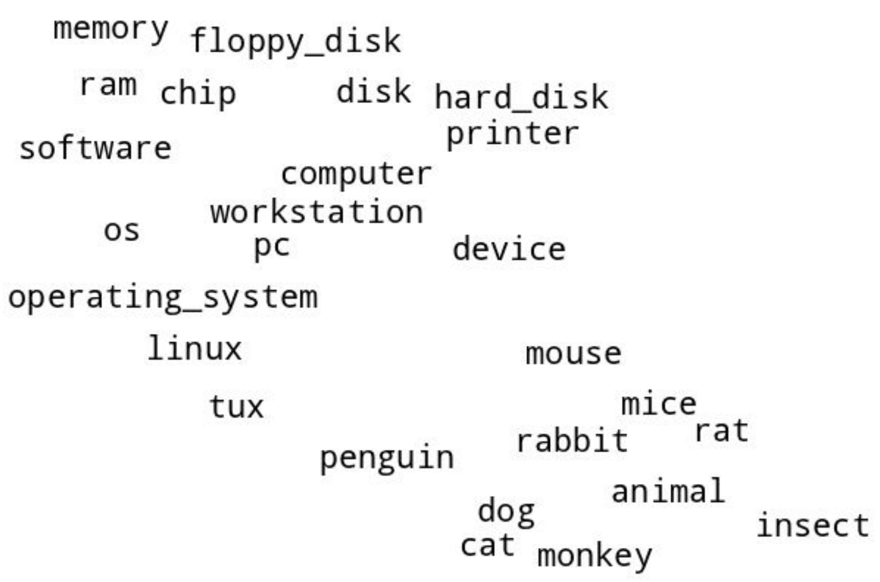


Distributional structure

Mathematical structures of language

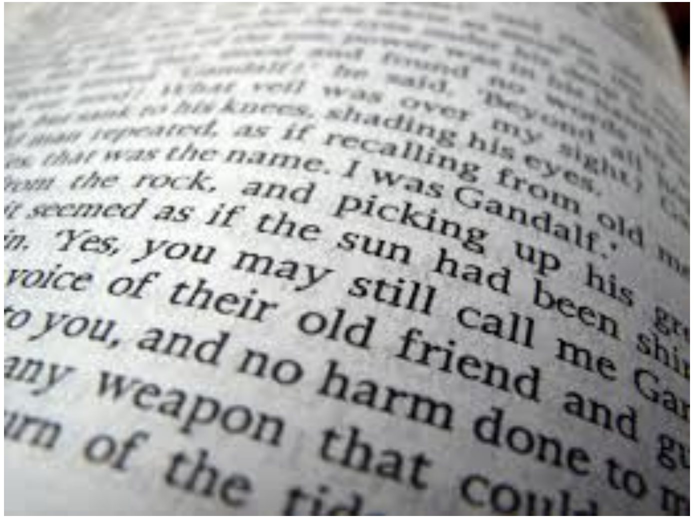
Distributional Semantic Models

- Analysis of word-usage statistics over huge corpora
- Geometric space of concepts
- Similar words are represented close in the space



Distributional Semantics

Extract co-occurrences



Text extraction

Yes, you may still call me…

co-occurrences

```
Yes -> [you, may]
you -> [Yes, may, still]
may -> [Yes, you, still, call]
still -> [you, may, call, me]
call -> [may, still, me, …]
me -> [still, call, …]
```

Distributional Semantics

Count co-occurrences

	dog	cat	bread	pasta	meat	mouse
dog	40	27	1	0	1	5
cat	27	32	0	1	0	8
bread	1	0	22	15	8	0
pasta	0	1	15	24	10	1
meat	1	0	8	10	30	2
mouse	5	8	0	1	2	31

Distributional Semantics

Word similarity

	dog	cat	bread	pasta	meat	mouse
dog	40	27	1	0	1	5
cat	27	32	0	1	0	8
bread	1	0	22	15	8	0
pasta	0	1	15	24	10	1
meat	1	0	8	10	30	2
mouse	5	8	0	1	2	31

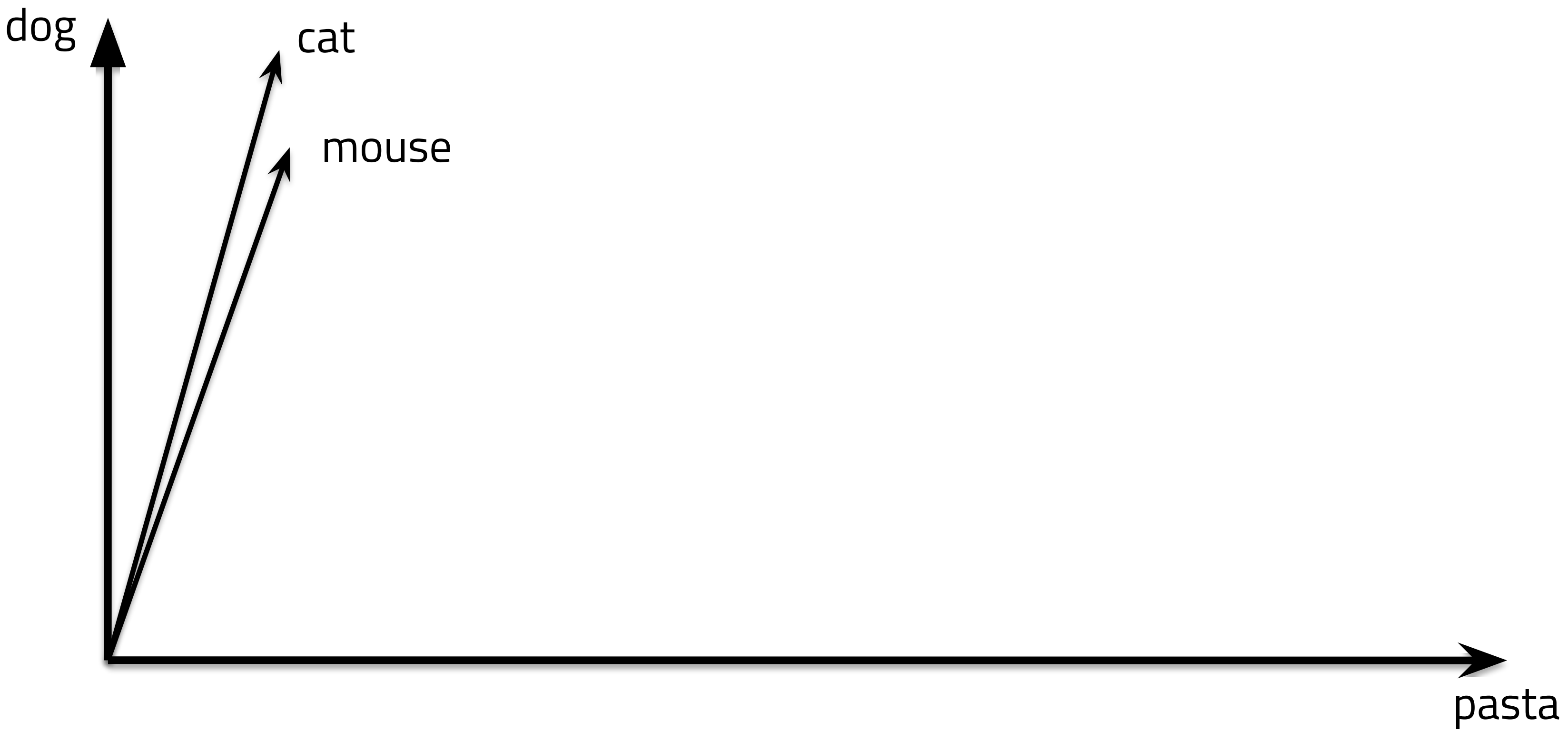
Distributional Semantics
Word Similarity

	dog	cat	bread	pasta	meat	mouse
dog	40	27	1	0	1	5
cat	27	32	0	1	0	8
bread	1	0	22	15	8	0
pasta	0	1	15	24	10	1
meat	1	0	8	10	30	2
mouse	5	8	0	1	2	31

Distributional Semantics
Word Similarity

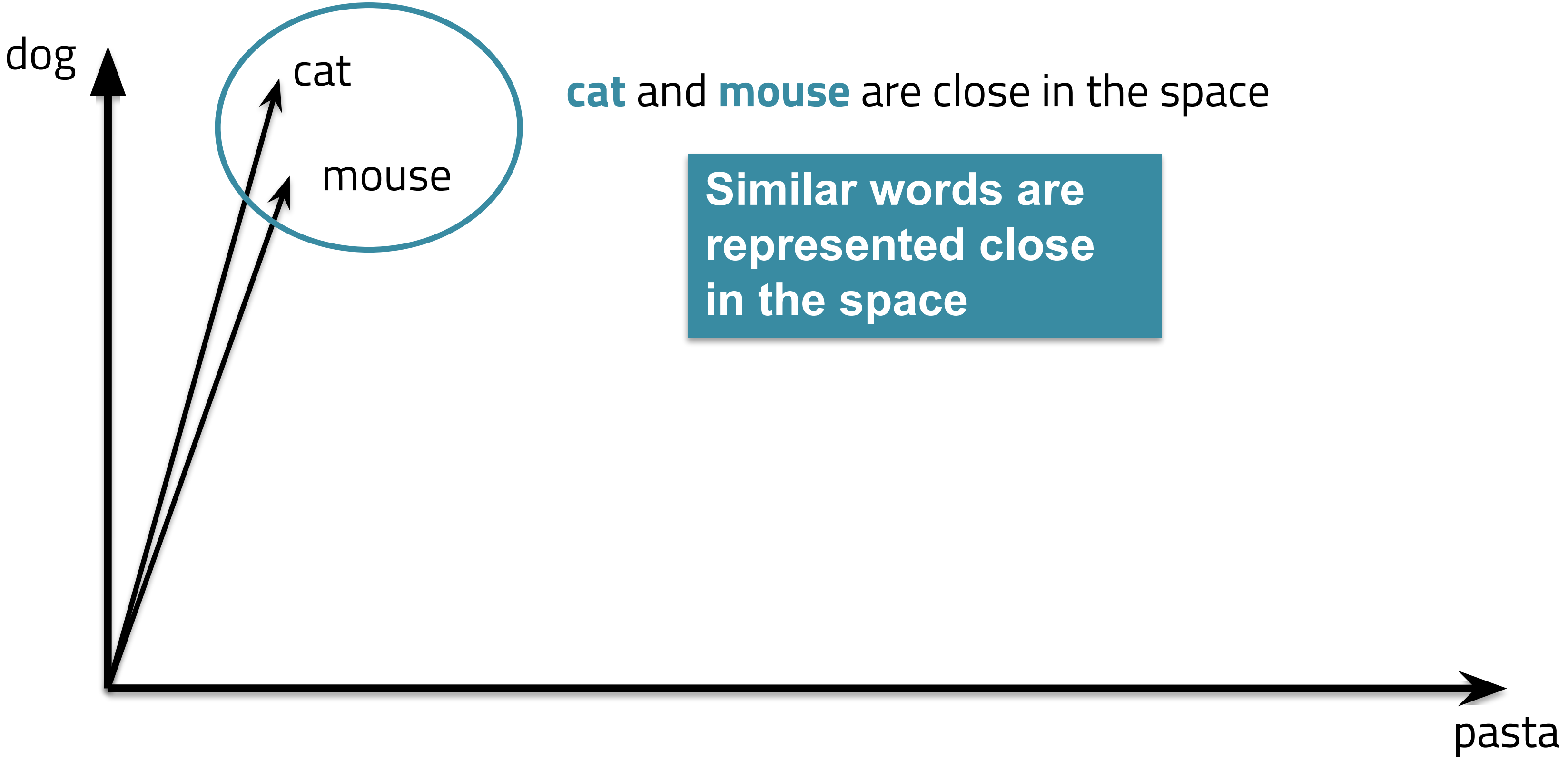
	dog	cat	bread	pasta	meat	mouse
dog	40	27	1	0	1	5
cat	27	32	0	1	0	8
bread	1	0	22	15	8	0
pasta	0	1	15	24	10	1
meat	1	0	8	10	30	2
mouse	5	8	0	1	2	31

Geometric space
WordSpace



Geometric space

WordSpace



cat and **mouse** are close in the space

Similar words are represented close in the space

DSM generalization

- A DSM is defined as <T, C, R, W, M, d, S>
  - T: target elements (words)
  - C: contexts
  - R: the relation between T and C
  - W: weighting schema
  - M: geometric space T*x*C
  - d: matrix reduction M -> M'
  - S: similarity function in M'

Build a DSM

1. Corpus pre-processing
2. Identify words and contexts
3. Count co-occurrences (words in contexts)
4. Weight (optional)
5. Space reduction (optional)

Parameters

- **The definition of context**
  - surrounding words, phrase, sentence, paragraph, document, syntactic context
- Weighting schema
- Similarity function

1. Pre-processing

- **Tokenization is necessary!**
  - PoS-tagging
  - Lemmatization
  - Parsing
- **A deep analysis**
  - Introduces errors
  - Requires other parameters
  - Language dependent

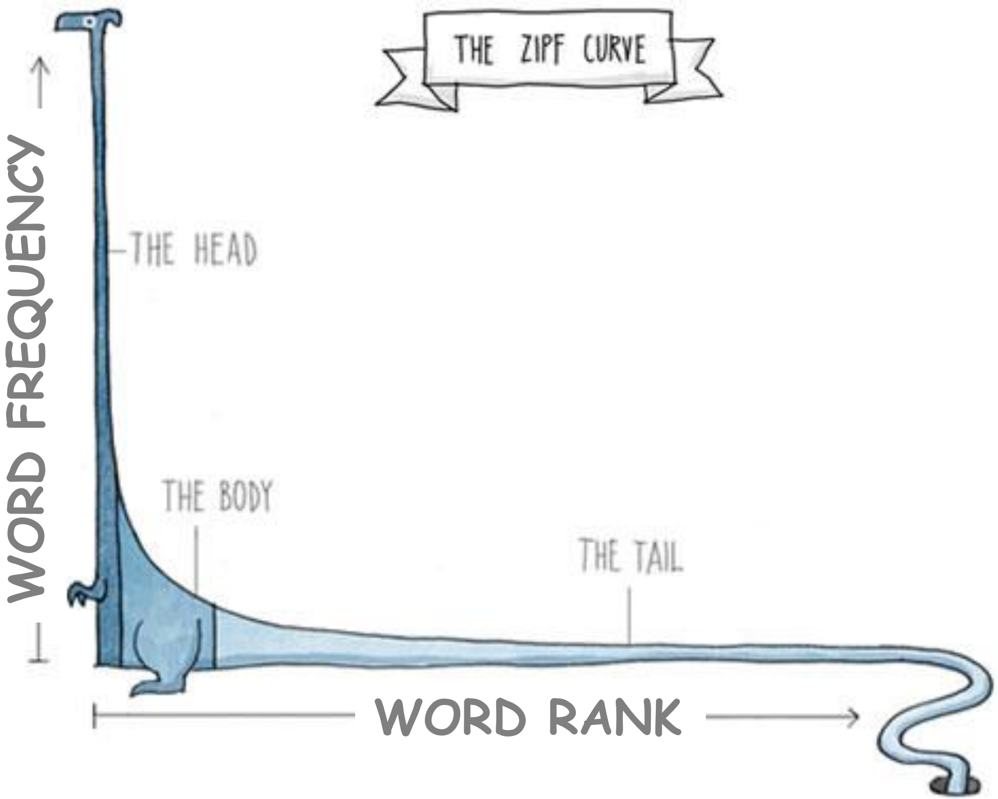
2. The context

- Document
  - the whole document
  - paragraph, sentence, passage
- Word
  - Most *n* frequent words
  - Where?
    - surrounding words (window)
    - pattern
    - syntactic dependency

3. Weighting schema

- Occurrences
- log(occurrences): relax most frequent words
- Mutual Information, Log-Likelihood Ratio
- Tf-Idf, word-entropy, …

Why sparse?



THE ZIPF CURVE
WORD FREQUENCY
THE HEAD
THE BODY
THE TAIL
WORD RANK

5. Matrix reduction

- DSM is high dimensional and very sparse:
  1. matrix reduction: LSI/LSA, PCA
  2. Random Indexing
  3. ...

Latent Semantic Analysis (LSA)

$$
\begin{array}{ccccccccc}
 & X & & & U & & \Sigma & & V^T \\
 & \begin{array}{c} \mathbf{t}_j \\ \downarrow \end{array} & & & & & & & \begin{array}{c} \mathbf{t'}_j \\ \downarrow \end{array} \\
(\mathbf{t}_i^T) \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,j} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \dots & x_{i,j} & \dots & x_{i,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,j} & \dots & x_{m,n} \end{bmatrix} & = & (\hat{\mathbf{t}}_i^T) \rightarrow & \begin{bmatrix} \begin{bmatrix} \\ \\ \mathbf{u}_1 \\ \\ \\ \end{bmatrix} \dots \begin{bmatrix} \\ \\ \mathbf{u}_l \\ \\ \\ \end{bmatrix} \end{bmatrix} & \cdot & \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{bmatrix} & \cdot & \begin{bmatrix} [& \mathbf{v}_1 &] \\ & \vdots & \\ [& \mathbf{v}_l &] \end{bmatrix}
\end{array}
$$

$X=U\Sigma V^T$, we can consider a lower-dimensional approximation of the higher-dimensional space by keeping only the first k singular values -> $X_k=U_k\Sigma_k V_k^T$

Word embedding

Word embedding

- Words or phrases from the vocabulary are mapped to vectors of real numbers
  - similar to DSM-based approaches
- Involves a mathematical embedding from a **space with many dimensions** per word to a **continuous vector space with a much lower dimension**
  - similar to matrix reduction
- **Dimensionality reduction on the word co-occurrence matrix** can be considered a word embedding!

Word embedding
issues

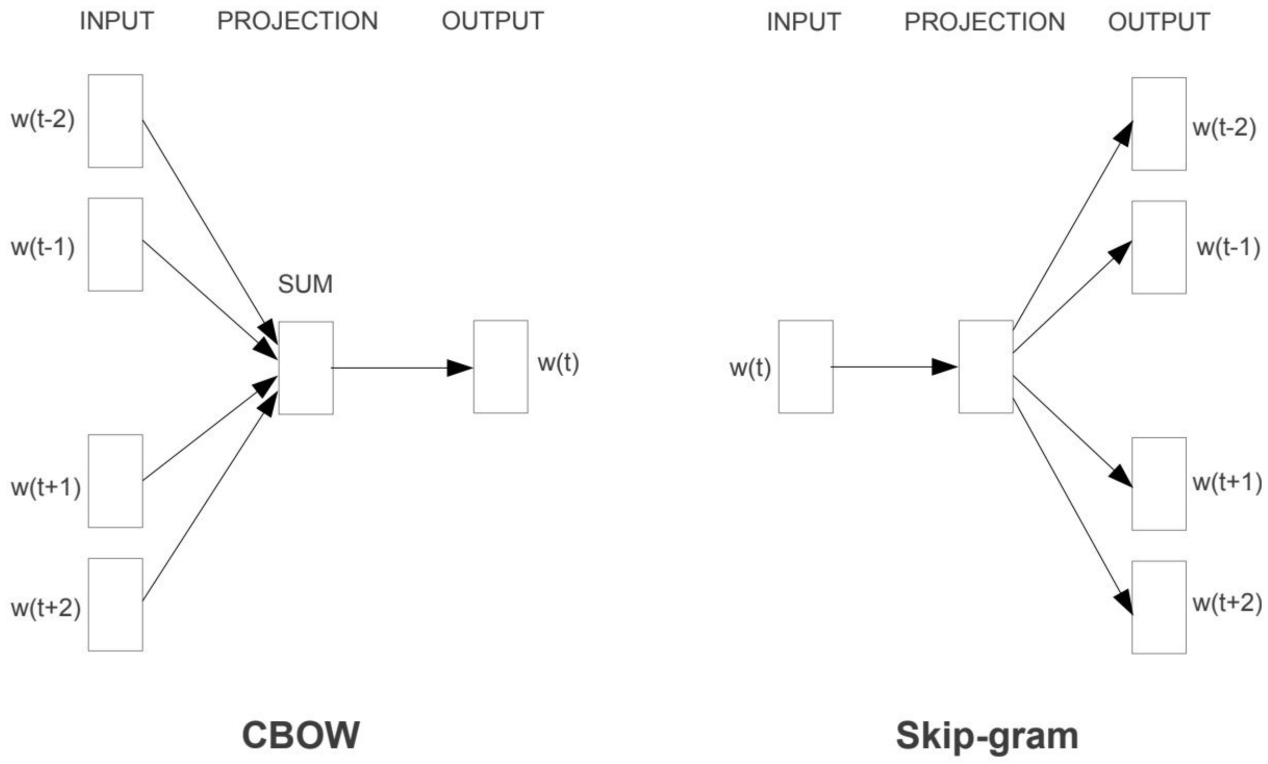
- **possible meanings** of a word **are conflated** into a single representation (a single vector in the semantic space)
  - sense-based vectors are a solution
- words are represented in isolation
  - composition of vectors is necessary for representing complex structures (sentences)
- word vector space models in general have the same issues

word2vec

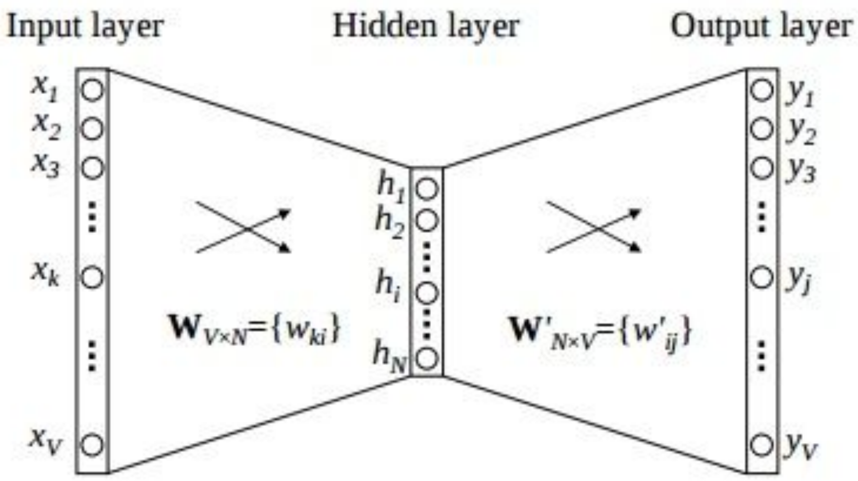
- it is used to produce word embeddings
- two-layer neural networks that are trained to reconstruct linguistic contexts of words
- word vectors are positioned in the vector space such that words that **share common contexts** in the corpus are **located in close proximity** to one another

Mikolov, Tomas; et al. (2013). *Efficient Estimation of Word Representations in Vector Space*

CBOW vs Skip-gram



CBOW



The input layer and the target, both are one-hot encoded of size [1 X V]

Input-Hidden layer matrix size =[V X N] , hidden-Output layer matrix size =[N X V] : where N is the number of dimensions. Also, N is the number of neurons in the hidden layer.

The input is multiplied by the input-hidden weights and called hidden activation.

The hidden input gets multiplied by hidden- output weights and output is calculated.

Error between output and target is calculated and propagated back to re-adjust the weights.

The weight between the hidden layer and the output layer is taken as the word vector representation of the word.

word2vec
objective function

$$-\log(p(w_O)|p(w_I))$$

context words

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^\top v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left({v'_w}^\top v_{w_I}\right)}$$

The objective function tries to increase $v'^{T}_{wO} v_{wI}$, this means that words that share more contexts will be similar to each other.

word2vec
parameters

- **vector dimension:** 100–1,000
- **training algorithm**
  - hierarchical softmax: works better for infrequent words
  - negative sampling: works better for frequent words and better with low dimensional vectors
- **sub-sampling:** words with frequency above a certain threshold may be subsampled
- **context window:** how many words before and after a given word would be included as context words, recommended values: 10 for skip-gram and 5 for CBOW

Thanks!!

Any questions?

You can find me at @basilepp
pierpaolo.basile@uniba.it