

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



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



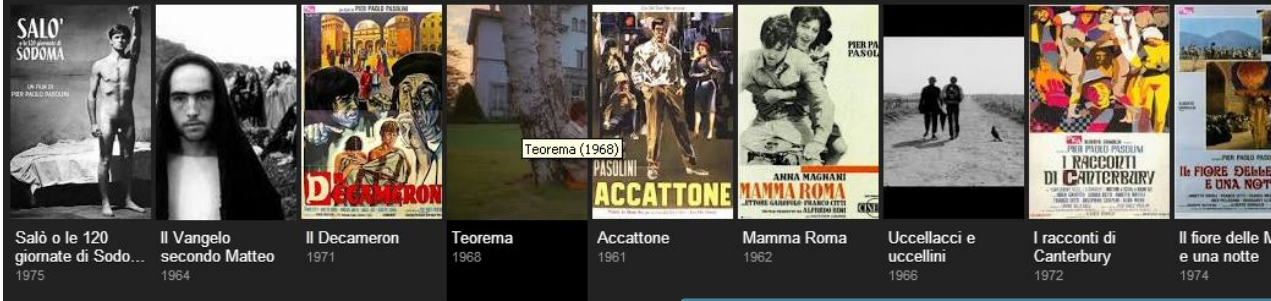
NLP Applications: Search

film del regista pier paolo pasolini



Web Notizie Immagini Shopping Video Altro Strumenti di ricerca

Pier Paolo Pasolini > Film



I cookie ci aiutano a fornire i nostri servizi. Utilizzando tali servizi, accetti l'utilizzo dei cookie da parte nostra.

Informazioni

OK

Opere cinematografiche di Pier Paolo Pasolini - Wikipedia

it.wikipedia.org/wiki/Opere_cinematografiche_di_Pier_Paolo_Pasolini

Passa a **Pasolini regista** - Pasolini inizia la sua attività di regista nel 1961 con il film **Accattone** che ambienta nelle borgate ... Nel 1962 produce il lungometraggio **Mamma Roma** nel quale riprendeva personaggi e ambienti del film precedente con ... dal **Il Decameron** del 1971, da **I racconti di Canterbury** del 1972 e da ...

Pier Paolo Pasolini - Wikipedia

it.wikipedia.org/wiki/Pier_Paolo_Pasolini

Pier Paolo Pasolini

Poeta

Pier Paolo Pasolini è stato un poeta, scrittore, regista, sceneggiatore, drammaturgo ed editorialista italiano. È considerato uno dei maggiori artisti e intellettuali italiani del XX secolo. Wikipedia

Data di nascita: 5 marzo 1922, Bologna

Assassinato: 2 novembre 1975, Ostia

Libri: Ragazzi di vita, Affabulazione, Petrolio

Genitori: Carlo Alberto Pasolini, Susanna Pasolini



NLP Applications: Machine Translation

The screenshot shows the Google Translate web interface. At the top is the Google logo. Below it, the word "Traduttore" is displayed in red. To the right of "Traduttore" is a link that says "Disattiva traduzione istantanea". Below this, there are two rows of language selection buttons. The first row contains "Inglese", "Italiano", "Francese", and "Rileva lingua" with a dropdown arrow. The second row contains "Italiano", "Inglese", "Spagnolo", and a dropdown arrow. To the right of these buttons is a blue button labeled "Traduci". Below the language buttons, there are two large text input areas. The left area contains the text "Hello" and has a close button (x) in the top right corner. Below the text input area are icons for voice input, text input, and a dropdown arrow. The right area contains the text "Ciao" and has a close button (x) in the top right corner. Below the text input area are icons for voice input, text input, and a dropdown arrow. Below the text input areas, there are two sections. The left section is titled "Vedi anche" and contains the text "Hello!, hello". The right section is titled "Traduzioni di Hello!" and contains a list of translations: "Ciao!", "Salve!", "Pronto?", "Pronto!", and "Piacere!". Each translation is preceded by a small grey square icon. To the right of each translation is a list of equivalent phrases in Italian. At the bottom of the page, there is a footer with the text "Google Traduttore per il Business:" followed by three links: "Translator Toolkit", "Traduttore di siti web", and "Strumento a supporto dell'export".

Google

Traduttore

Disattiva traduzione istantanea

Inglese Italiano Francese Rileva lingua

Italiano Inglese Spagnolo

Traduci

Hello

Ciao

Vedi anche

Hello!, hello

Traduzioni di Hello!

interiezione

- Ciao! Hello!, Hi!, Bye-Bye!, Bye!, So long!, Cheerio!
- Salve! Hi!, Hello!, Hail!, Cheers!, Hallo!
- Pronto? Hello!
- Pronto! Hello!, Hullo!, Hallo!
- Piacere! Hello!, Hi!

Google Traduttore per il Business: [Translator Toolkit](#) [Traduttore di siti web](#) [Strumento a supporto dell'export](#)

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?

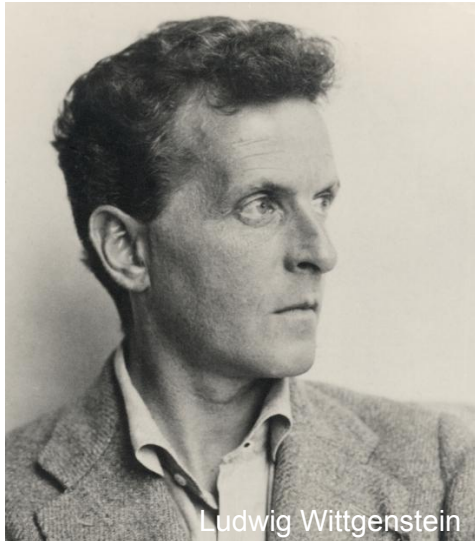
Tezguno



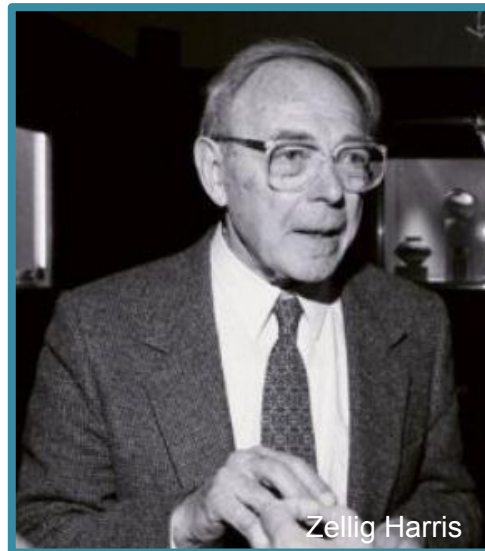
Distributional Semantic Models (DSM)



*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

A 2D word embedding visualization showing hierarchical relationships between computer hardware, software, and animals. The words are arranged in a way that reflects their semantic similarity and relationships. For example, 'memory' and 'floppy_disk' are close together, as are 'ram' and 'chip'. 'software' is a general term, with 'os' and 'operating_system' as specific instances. 'computer' is a general term, with 'workstation' and 'pc' as specific instances. 'device' is a general term, with 'mouse' and 'printer' as specific instances. 'animal' is a general term, with 'dog', 'cat', 'monkey', 'rabbit', 'mice', 'rat', and 'insect' as specific instances. The words are arranged in a way that reflects their semantic similarity and relationships.

memory floppy_disk
ram chip disk hard_disk
software computer printer
os workstation
pc device
operating_system
linux mouse
tux
penguin rabbit mice rat
dog animal
cat monkey insect

Distributional Semantics

Extract co-occurrences

Text
extraction

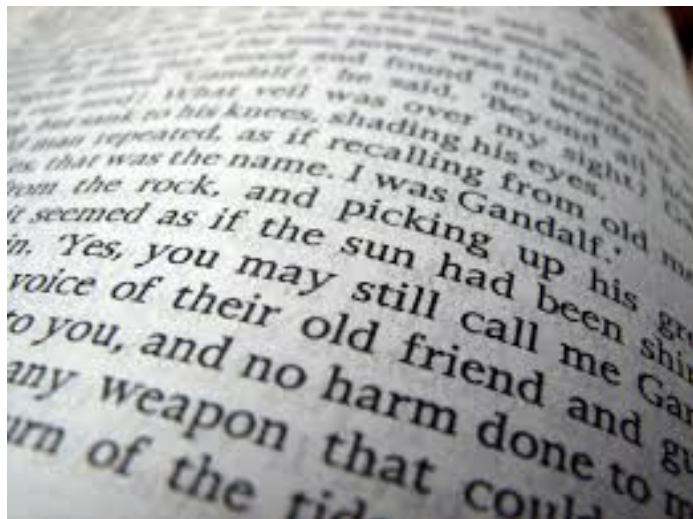


Yes, you may still call me...



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

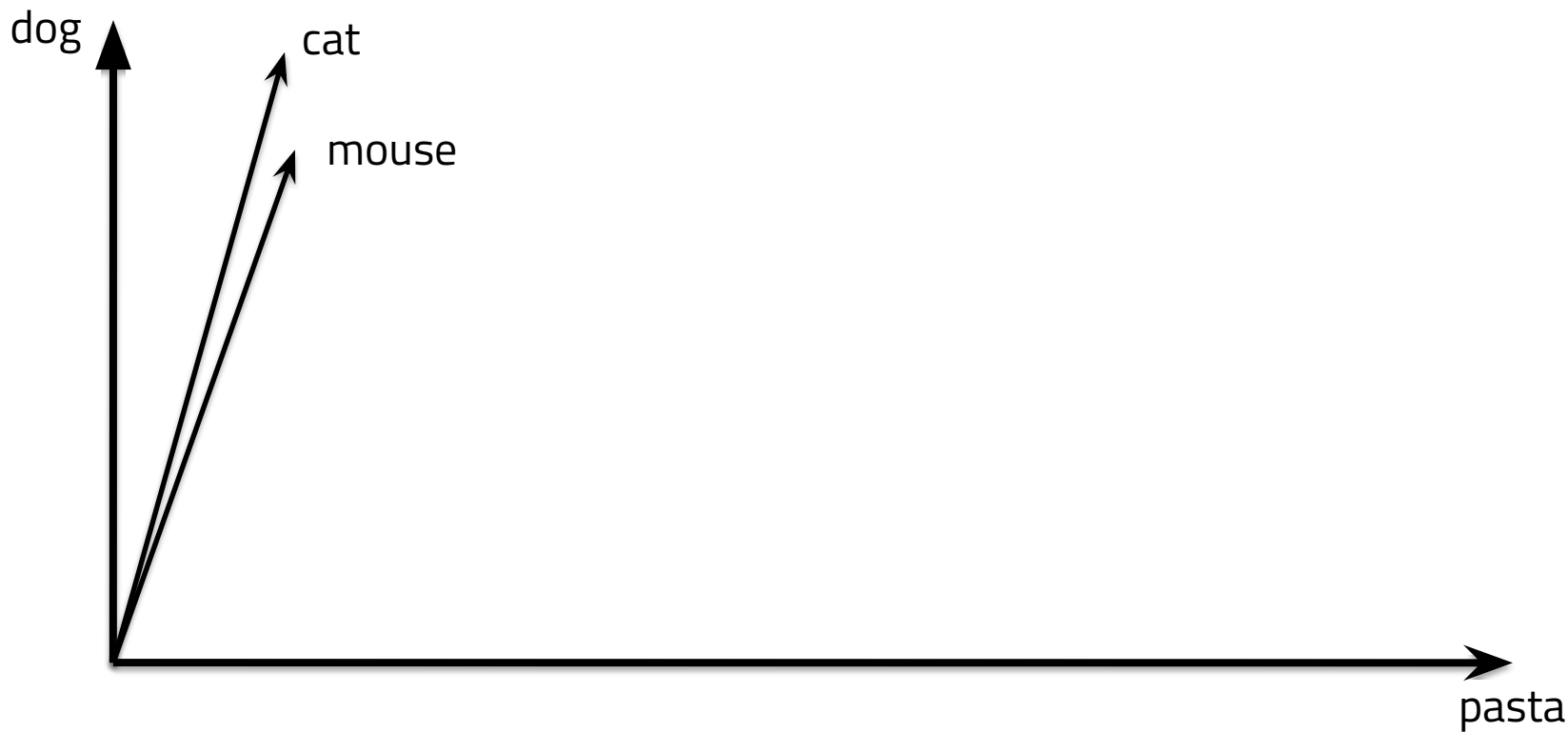
Word Similarity



| | 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 |

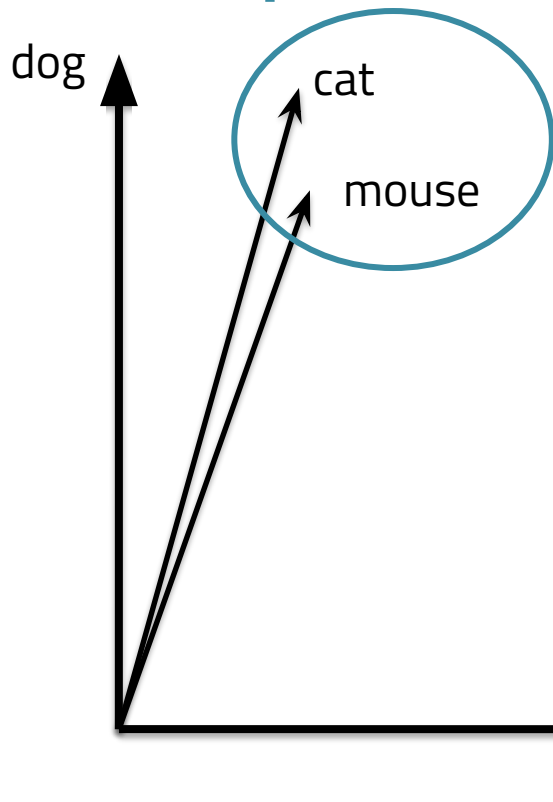
Geometric space

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 $\langle T, C, R, W, M, d, S \rangle$
 - T: target elements (words)
 - C: contexts
 - R: the relation between T and C
 - W: weighting schema
 - M: geometric space $T \times C$
 - d: matrix reduction $M \rightarrow M'$
 - 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

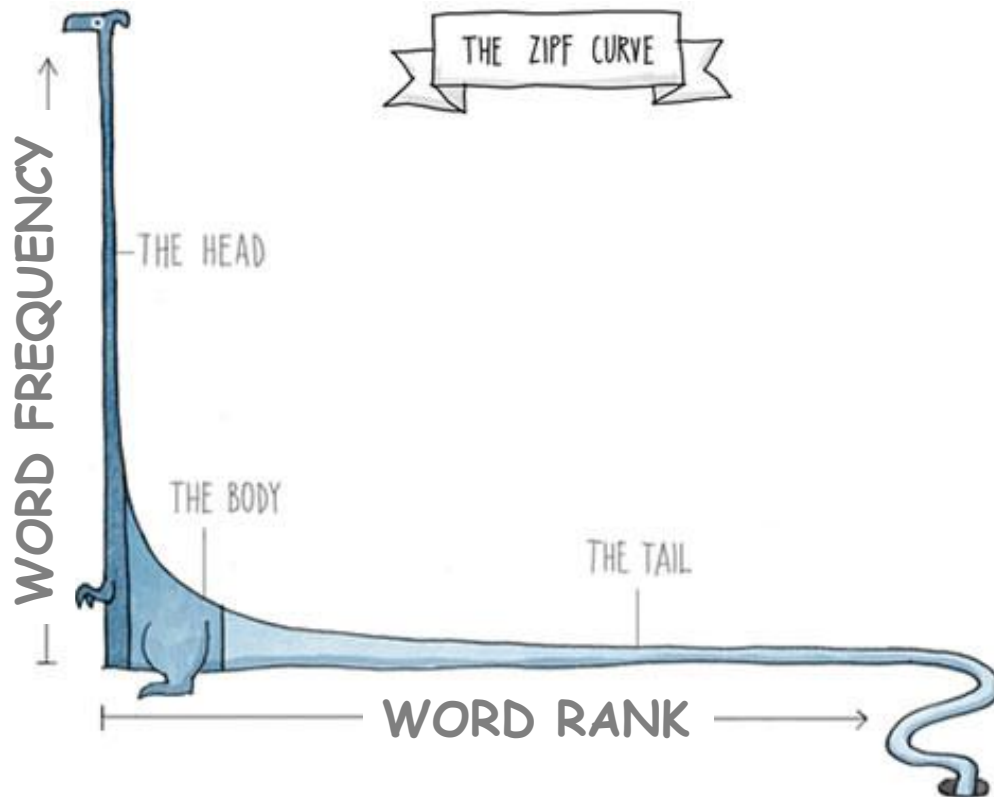
2. The context

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

3. Weighting schema

- Occurrences
- $\log(\text{occurrences})$: relax most frequent words
- Mutual Information, Log-Likelihood Ratio
- Tf-Idf, word-entropy, ...

Why sparse?



5. Matrix reduction

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

Latent Semantic Analysis (LSA)

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

$X=U\Sigma V^T$, 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!

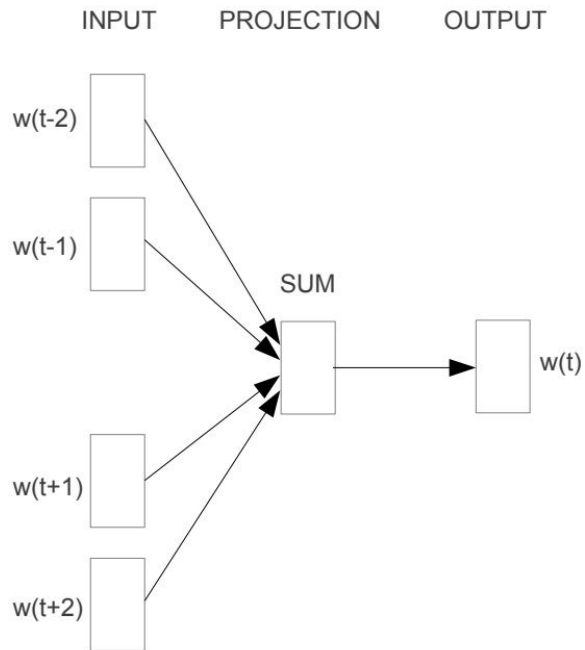
Word embedding issues

- **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

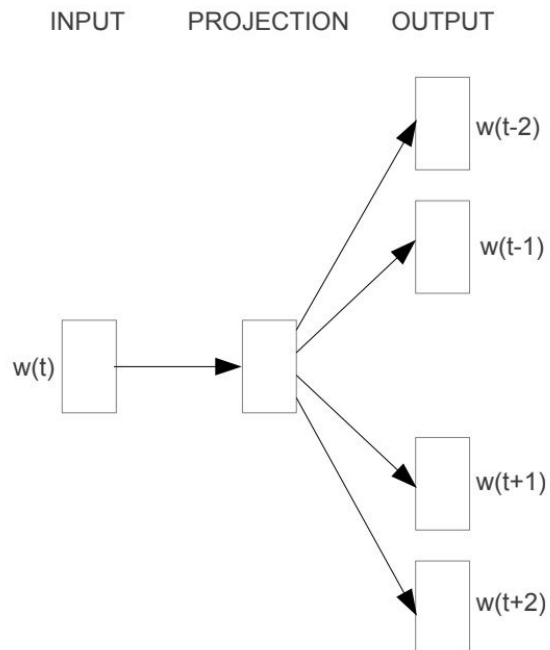
word2vec

- 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

CBOW vs Skip-gram

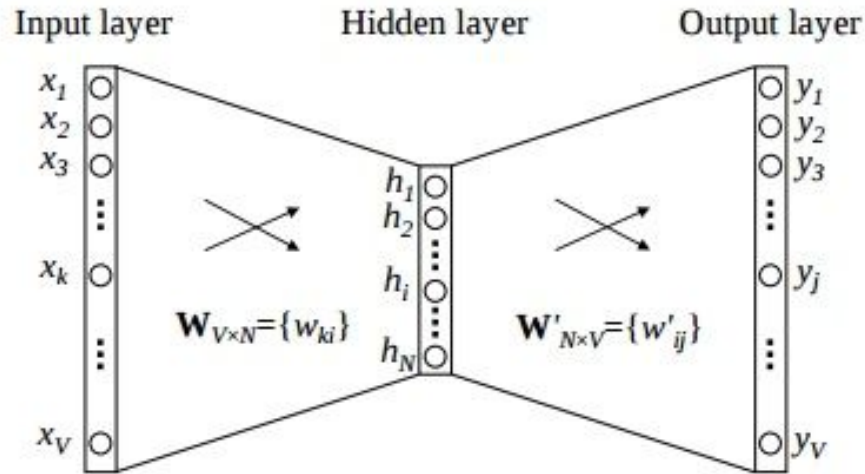


CBOW



Skip-gram

CBOW



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

Input-Hidden layer matrix size $= [V \times N]$, hidden-Output layer matrix size $= [N \times V]$: 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.

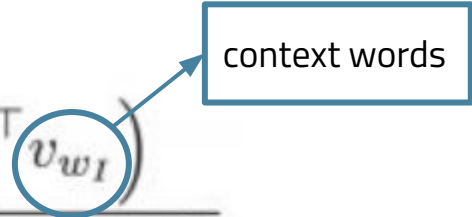
The hidden input gets multiplied by hidden-output weights and output is calculated.

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

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

$$p(w_o|w_I) = \frac{\exp(v'_{w_o} \cdot v_{w_I})}{\sum_{w=1}^W \exp(v'_w \cdot v_{w_I})}$$


context words

The objective function tries to increase $v'_{w_o} \cdot v_{w_I}$, 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