Natural Language Processing - 2nd Semester (2024-2025)
1038141

1.11 - Vector Semantics (sparse)

SAPIENZA
UNIVERSITÀ DI ROMA

Prof. Stefano Faralli
faralli@di.uniroma1.it

Prof. Iacopo Masi
masi@di.uniroma1.it

**credits are reported in the last slide

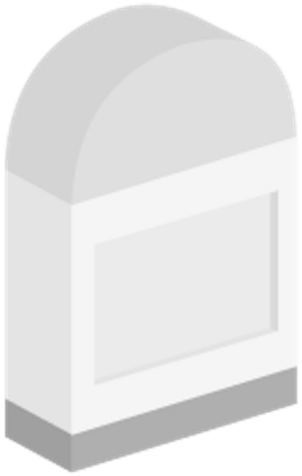
SAPIENZA
UNIVERSITÀ DI ROMA

1.11 - Vector Semantics (sparse)

- word meaning, lexical semantics relations
- similarity vs. relatedness, the distributional hypothesis
- Semantic space models, vector space models, word embeddings, bag-of-words
- vector similarity measures
- term-frequency inverse-document frequency, pointwise mutual information, positive pointwise mutual information
- problems with VSMs
- Q&A

Milestones in NLP

today topic is fundamental for the two areas below.



rule-based systems

statistical classical machine learning models

deep learning models

Previously …

https://www.youtube.com/watch?v=eX-BCKutDh0

Word meaning (a.k.a. lexical semantics)

What is the **meaning** of a word?

Word meaning (a.k.a. lexical semantics)

First answer: think of a dictionary!

bank[1] | baŋk |

noun

1 the land alongside or sloping down to a river or lake: *willows lined the bank of the stream.*

2 a long, high mass or mound of a particular substance: *a grassy bank* | *a bank of snow.*
- an elevation in the seabed or a riverbed; a mudbank or sandbank.
- a transverse slope given to a road, railway, or sports track to enable vehicles or runners to maintain speed round a curve.
- *[mass noun]* the sideways tilt of an aircraft when turning in flight: *a rather steep angle of bank.*

3 a set of similar things, especially electrical or electronic devices, grouped together in rows: *the DJ had big banks of lights and speakers on either side of his console.*
- a tier of oars: *the early ships had only twenty-five oars in each bank.*

4 the cushion of a pool table: *[as modifier] : a bank shot.*

verb *[with object]*

1 heap (a substance) into a mass or mound: *the rain* ***banked*** *the soil* ***up*** *behind the gate* | *snow was banked in humps at the roadside.*
- *[no object]* form into a mass or mound: *purple clouds* ***banked up*** *over the hills.*
- heap up (a fire) with tightly packed fuel so that it burns slowly: *she* ***banked up*** *the fire.*
- edge or surround with a ridge or row of something: *steps banked with pots of chrysanthemums.*

2 (with reference to an aircraft or vehicle) tilt or cause to tilt sideways in making a turn: *[no object] : the plane banked as if to return to the airport* | *[with object] : I banked the aircraft steeply and turned.*
- *[with object]* build (a road, railway, or sports track) higher at the outer edge of a bend to facilitate fast cornering: *the track was banked to allow a train to take curves faster while maintaining passenger comfort* | (as adjective **banked**) *: a banked racetrack.*

3 (often as noun **banking**) *British* (of a locomotive) provide additional power for (a train) in ascending an incline.

4 (of an angler) succeed in landing (a fish): *it was the biggest rainbow trout that had ever been banked.*

5 *North American* (in pool) play (a ball) so that it rebounds off a surface such as a cushion: *I banked the eight ball off two cushions.*

Word meaning (a.k.a. lexical semantics)

First answer: think of a dictionary!

- Main elements:
  - Lemmas
  - Senses
  - Definitions

bank[1] | baŋk |

noun

1 the land alongside or sloping down to a river or lake: *willows lined the bank of the stream.*

2 a long, high mass or mound of a particular substance: *a grassy bank* | *a bank of snow.*
- an elevation in the seabed or a riverbed; a mudbank or sandbank.
- a transverse slope given to a road, railway, or sports track to enable vehicles or runners to maintain speed round a curve.
- *[mass noun]* the sideways tilt of an aircraft when turning in flight: *a rather steep angle of bank.*

3 a set of similar things, especially electrical or electronic devices, grouped together in rows: *the DJ had big banks of lights and speakers on either side of his console.*
- a tier of oars: *the early ships had only twenty-five oars in each bank.*

4 the cushion of a pool table: *[as modifier] : a bank shot.*

verb *[with object]*

1 heap (a substance) into a mass or mound: *the rain* ***banked*** *the soil* ***up*** *behind the gate* | *snow was banked in humps at the roadside.*
- *[no object]* form into a mass or mound: *purple clouds* ***banked up*** *over the hills.*
- heap up (a fire) with tightly packed fuel so that it burns slowly: *she* ***banked up*** *the fire.*
- edge or surround with a ridge or row of something: *steps banked with pots of chrysanthemums.*

2 (with reference to an aircraft or vehicle) tilt or cause to tilt sideways in making a turn: *[no object] : the plane banked as if to return to the airport* | *[with object] : I banked the aircraft steeply and turned.*
- *[with object]* build (a road, railway, or sports track) higher at the outer edge of a bend to facilitate fast cornering: *the track was banked to allow a train to take curves faster while maintaining passenger comfort* | (as adjective **banked**) *: a banked racetrack.*

3 (often as noun **banking**) *British* (of a locomotive) provide additional power for (a train) in ascending an incline.

4 (of an angler) succeed in landing (a fish): *it was the biggest rainbow trout that had ever been banked.*

5 *North American* (in pool) play (a ball) so that it rebounds off a surface such as a cushion: *I banked the eight ball off two cushions.*

Lexical semantic relations: Synonymity and Antonymity

- Synonyms: words that have **the same meaning** in some or all contexts.
- Examples
  - couch / sofa
  - big / large
  - automobile / car
  - love / enjoy
  - water / H20
- Senses that have opposite meanings are called instead **antonyms**, e.g., (up/down, love/hate, small/big)

Semantic similarity vs. relatedness

- Words with similar meanings are connected by the relation of ***semantic similarity***.
- Such words are not synonyms, but rather share some element of meaning
- Examples: „car“ and „bicycle“ / „lion“ and „bear“
- We refer to generic word association as ***semantic relatedness*** instead
- IMPORTANT: <u>semantic similarity ≠ semantic relatedness</u>
  - car, bicycle: similar
  - car, gasoline: related, not similar

Human perception of word similarity

word1	word2	similarity
vanish	disappear	9.8
behave	obey	7.3
belief	impression	5.95
muscle	bone	3.65
modest	flexible	0.98
hole	agreement	0.3

SimLex-999 dataset [Hill et al., 2015]

Lexical semantic relations: Hypernymy and hyponymy

- A ***hypernym*** of a sense / word meaning is a *generalization* of it (i.e., less specific, denotes a superclass)
  - "vehicle" is a hypernym of "bicycle"
  - "fruit" is a hypernym of "apple"

- Conversely, a ***hyponym*** is a *specialization* of a sense, hence denoting a subclass
  - "car" is a hyponym of "vehicle"
  - "banana" is a hyponym of "fruit"

Word senses: the enumerative approach

A fixed sense inventory enumerates the range of all possible meanings of a word

- Knife (as a noun)
  1. edge tool used as a cutting instrument; has a pointed blade with a sharp edge and a handle
  2. a weapon with a handle and blade with a sharp point
  3. any long thin projection that is transient

- Words in context are assumed to *select/activate* one of these senses in each context

  - She chopped the vegetables with a chef's knife.
    - ➠ Knife as a **CUTTING TOOL**
  - A man was beaten and cut with a knife.
    - ➠ Knife as a **WEAPON**

The enumerative approach limitations

- Knife (as a noun)
  1. edge tool used as a cutting instrument; has a pointed blade with a sharp edge and a handle
  2. a weapon with a handle and blade with a sharp point
  3. any long thin projection that is transient

? should we *add a further sense* to the inventory for "a cutting blade forming part of a machine"

? are word senses *application-independent*

? can we *enumerate all* possible senses anyway

A solution: word meaning from word usage

- KEY IDEA: **define word (senses) by their usages**

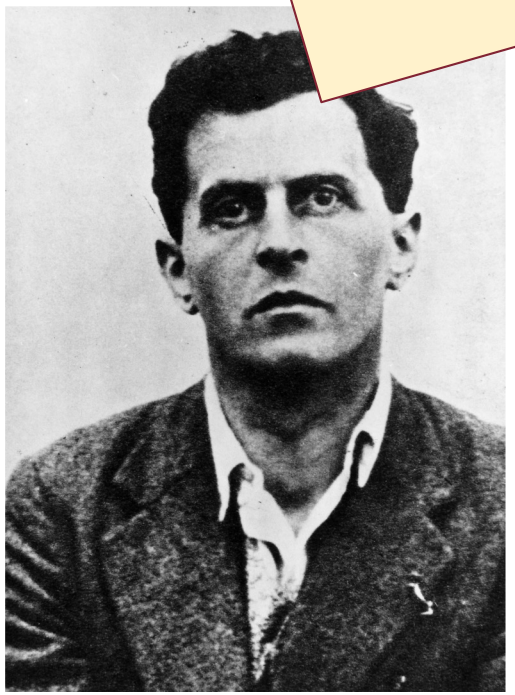
that is, words are defined by their *environments*, namely the words around them

A solution: word meaning from word usage

let's dig more!!

A solution: word meaning from word usage

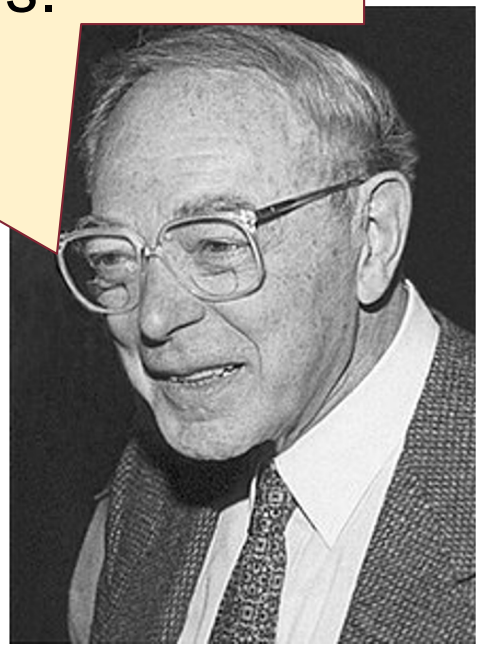
- Wittgenstein (1945): The meaning of a word is its use in the language.



https://en.wikipedia.org/wiki/Ludwig_Wittgenstein

A solution: word meaning from word usage

- Wittgenstein (1945): The meaning of a word is its use in the language.
- Zellig Harris (1954): If A and B have almost identical environments we say that they are synonyms.



https://en.wikipedia.org/wiki/Zellig_Harris

A solution: word meaning from word usage

- Wittgenstein (1945): The meaning of a word is its use in the language.
- Zellig Harris (1954): If A and B have almost identical environments we say that they are synonyms.
- Firth (1957): You shall know a word by the company it keeps.

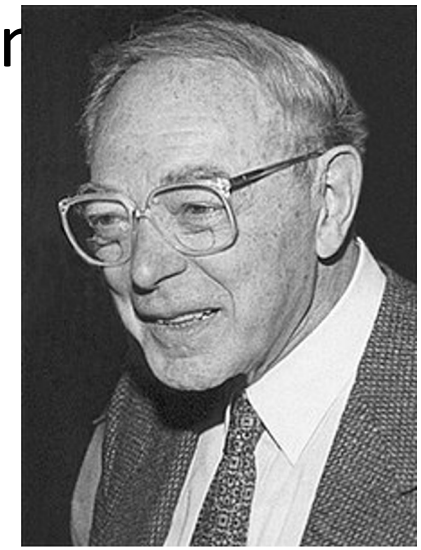
https://en.wikipedia.org/wiki/John_Rupert_Firth

The DISTRIBUTIONAL HYPOTHESIS

- **The distributional hypothesis**
  - similar words tend to appear in similar contexts

[Harris, 1954]

  - word co-occurrences reveal meaning ar relations



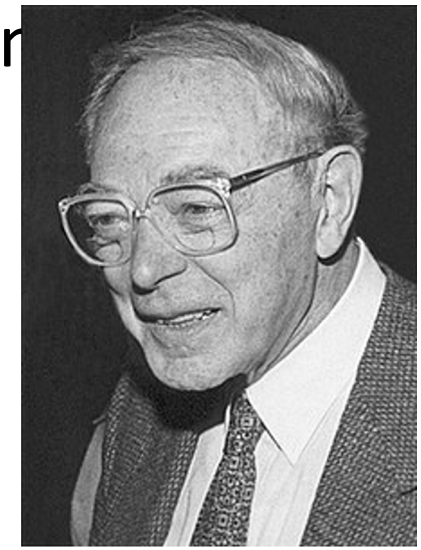
The DISTRIBUTIONAL HYPOTHESIS

that's how today we refer to "environments"

- **The distributional hypothesis**
  - similar words tend to appear in similar **contexts**

[Harris, 1954]

  - word co-occurrences reveal meaning ar relations



Example: what is ong choi?!

- Suppose you see these sentences:
  - *Ong choi is delicious sautéed with garlic*.
  - *Ong choi is superb over rice*
  - *Ong choi leaves with salty sauces*
- And you've also seen these:
  - *... spinach sautéed with garlic over rice*
  - *Chard stems and leaves are delicious*
  - *Collard greens and other salty leafy greens*

 You can infer that *ong choi* ***is a*** *leafy green like spinach, chard, or collard greens*

Example: what is ong choi?!

https://en.wikipedia.org/wiki/Ipomoea_aquatica

Lexical Semantic Vector Representations and Models

- VSMs are models of word meaning focused on similarity
- VSMs define the meaning of a word as a **vector**, a list of numbers, a **point** in N-dimensional space
- **Similar words are "nearby in space"**

Lexical Semantic Vector Representations and Models

- VSMs are models of word meaning focused on similarity
- VSMs define the meaning of a word as a **vector**, a list of numbers, a **point** in N-dimensional space
- **Similar words** are **"nearby in space"**

[Jurafsky&Martin, 2022]

Words in Space → Word Embeddings

- Representing words in a vector space is a standard process in NLP, called **embedding**
- It is called "**embedding**" because the objects are embedded into a vector space
- In our case, we embed words, so we obtain ***word embeddings***

Two well-known word embedding methods

- **Sparse** (today)
  - A common baseline model
  - Sparse vectors
  - Words are represented by a simple function of the counts of nearby words (**word co-occurrence**)
- **Dense** (later, with Prof. Masi)
  - dense vectors with latent dimensions -> latent space
  - e.g.: Word2Vec: where (among other) a classifier is trained to distinguish nearby and far-away words

Sparse VSMs in practice: Term-Document Matrix

given a corpus:




As you like It Twelfth Night Julius Caesar Henry V

given a vocabulary:

V={"battle", "good", "fool", "wit"}

we count how many times a word w in V occurs in each book:

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

- This matrix represents how often each word in the vocabulary is occurred in each document in the corpus
- Each document is represented by a **vector of words** (i.e., a column vector of the matrix)

Sparse VSMs in practice: Term-Document Matrix

given a corpus:




As you like It Twelfth Night Julius Caesar Henry V

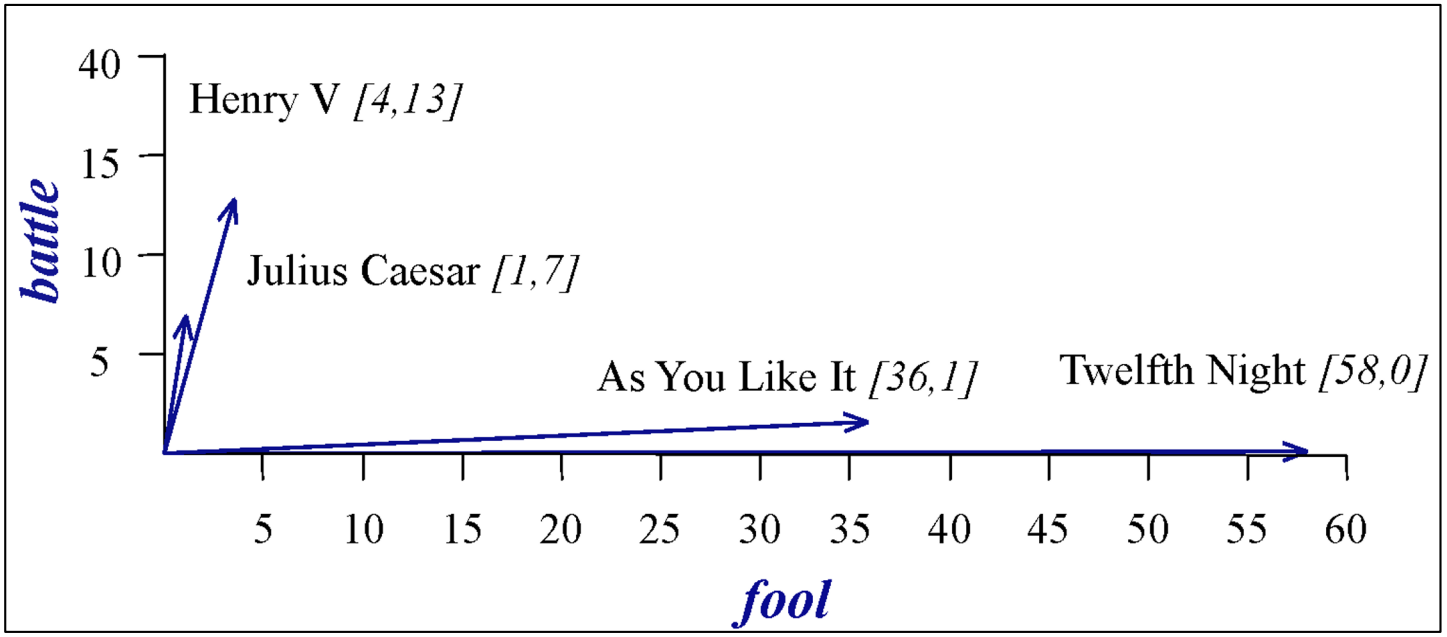
given a vocabulary:

V={"battle", "good", "fool", "wit"}

we count how many times a word w in V occurs in each book:

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

- This matrix represents how often each word in the vocabulary is occurred in each document in the corpus
- Each document is represented by a **vector of words** (i.e., a column vector of the matrix)



	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

- This matrix represents how often each word in the vocabulary is occurred in each document in the corpus
- Each document is represented by a **vector of words** (i.e., a column vector of the matrix)

Sparse VSMs in practice: Term-Document Matrix

given a corpus:




As you like It Twelfth Night Julius Caesar Henry V

given a vocabulary:

V={"battle", "good", "fool", "wit"}

we count how many times a word w in V occurs in each book:

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Sparse VSMs in practice: Term-Document Matrix

given a corpus:




As you like It Twelfth Night Julius Caesar Henry V

given a vocabulary:

V={"battle", "good", "fool", "wit"}

we count how many times a word w in V occurs in each book:

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

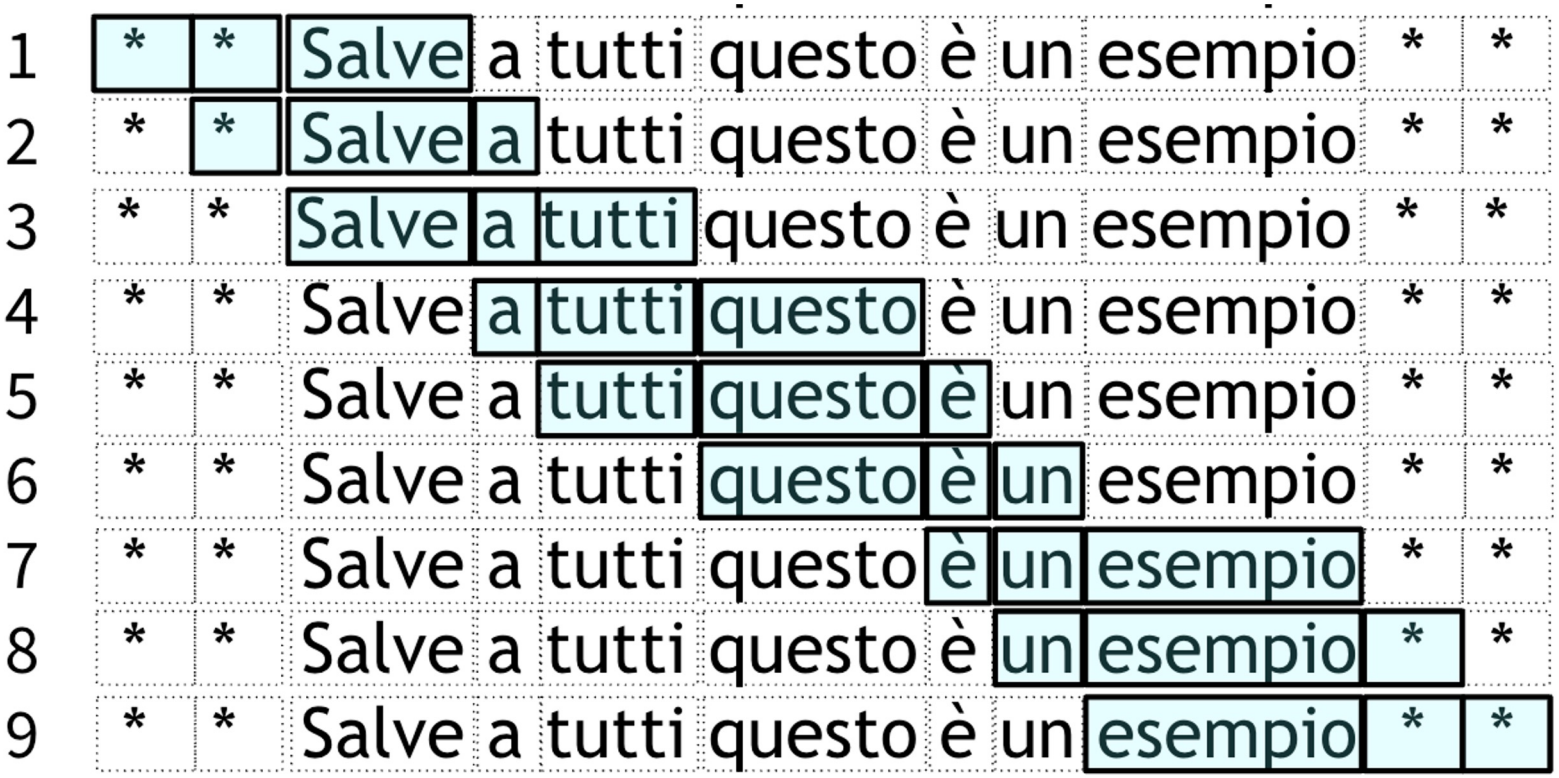
- Each word can also be represented by a vector of documents
- Note that some words are genre-specific (*battle*, *fool*), while some are not (*good*)

Sparse VSMs in practice: Word-Word Matrix (co-occurrences)

Given a context **C** with **m** words and a sliding window of length **n** we can obtain **m+n-1** sub-contexts (each of **n** words) by sliding the window on **C**.

C= “Salve a tutti questo è un esempio“

m=7, n=3 ( )



Sparse VSMs in practice: Word-Word Matrix (co-occurrences)

Given a context **C** with **m** words and a sliding window of length **n** we can obtain **m+n-1** sub-contexts (each of **n** words) by sliding the window on **C**.

C= “Salve a tutti questo è un esempio“

m=7, n=3 ( )

n-1 head and tail jolly characters are added to **C**

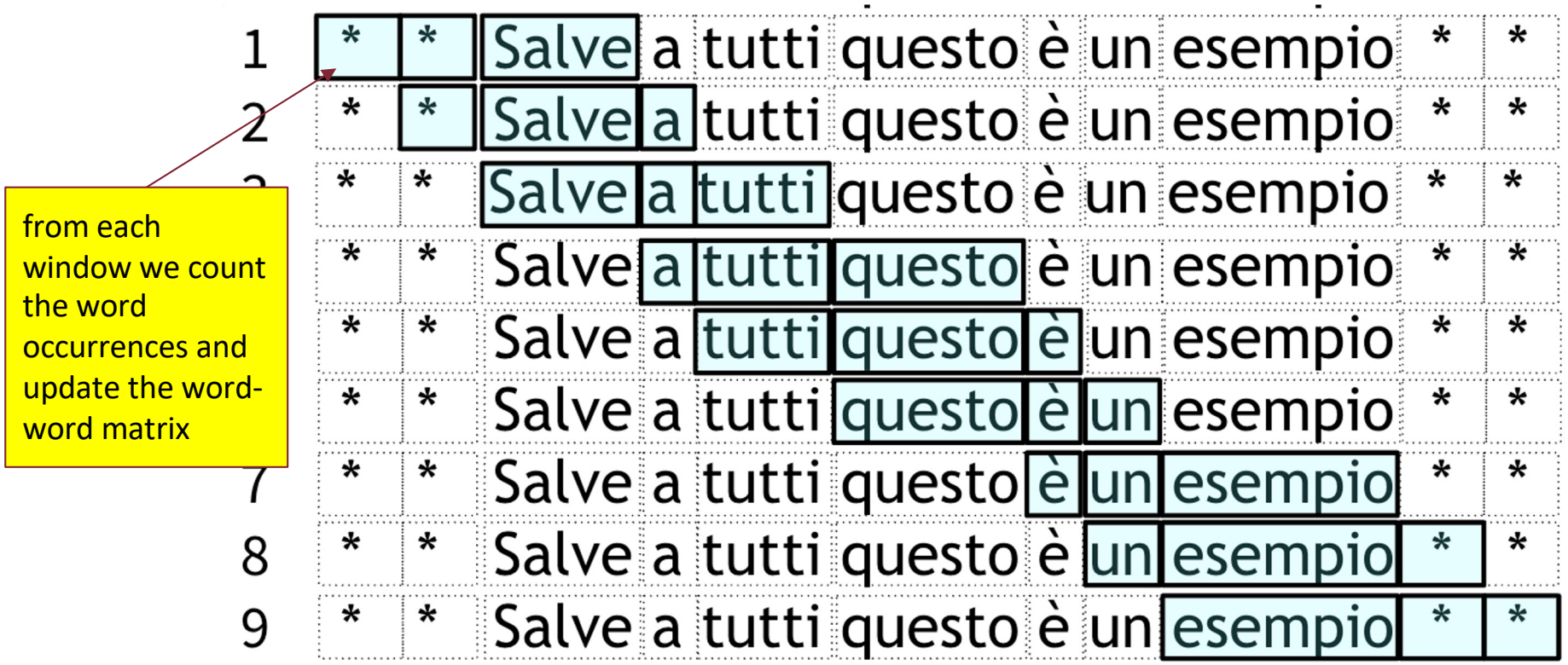
1	*	*	Salve	a	tutti	questo	è	un	esempio	*	*
2	*	*	Salve	a	tutti	questo	è	un	esempio	*	*
3	*	*	Salve	a	tutti	questo	è	un	esempio	*	*
4	*	*	Salve	a	tutti	questo	è	un	esempio	*	*
5	*	*	Salve	a	tutti	questo	è	un	esempio	*	*
6	*	*	Salve	a	tutti	questo	è	un	esempio	*	*
7	*	*	Salve	a	tutti	questo	è	un	esempio	*	*
8	*	*	Salve	a	tutti	questo	è	un	esempio	*	*
9	*	*	Salve	a	tutti	questo	è	un	esempio	*	*

Sparse VSMs in practice: Word-Word Matrix (co-occurrences)

Given a context **C** with **m** words and a sliding window of length **n** we can obtain **m+n-1** sub-contexts (each of **n** words) by sliding the window on **C**.

C= “Salve a tutti questo è un esempio“

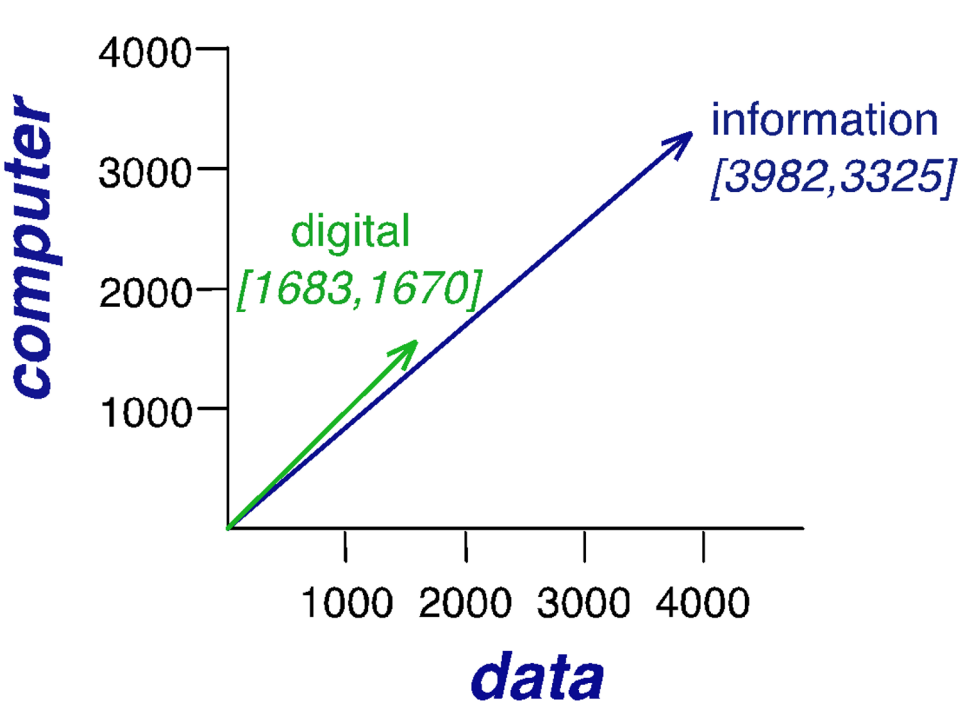
m=7, n=3 ( )



Sparse VSMs in practice: Word-Word Matrix (co-occurrences)

- each word is represented by a vector of word co-occurrences
- the counts depend from the length **n** of the sliding window

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	
strawberry	0	...	0	0	1	60	19	
digital	0	...	1670	1683	85	5	4	
information	0	...	3325	3982	378	5	13	



Bag of Words (BoW) representation

- It is convenient to represent vectors as **bags-of-words** (BoW) which are *real-valued multisets of words*
  - "the best of the best" → {the: 1, best: 1, of: 1} (**binary**)
  - "the best of the best" → {the: 2, best: 2, of: 1} (**counts**)
- Each vocabulary item is a dimension with an assigned value, so under VSM:
  - {the: 1, best: 1, of: 1} → (1, 1, 1)
  - {the: 2, best: 2, of: 1} → (2, 2, 1)

Vector similarity measures

How to compute a **similarity** between a pair of vectors?

Euclidean Distance

- Euclidean *distance* measures the length of the line segment connecting two points

$$d(\vec{a}, \vec{b}) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$

- The lower the distance, the higher is similarity

Dot product

- Also known as the **inner product**

$$\text{dot-product}(\vec{v},\vec{w}) = \vec{v}\cdot\vec{w} = \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + \ldots + v_N w_N$$

- Problem: dot product is higher if a vector is longer, with higher values in each dimension
- That is, dot product will be higher for frequent words
- Solution: normalize by vector length

$$|\vec{v}| = \sqrt{\sum_{i=1}^{N} v_i^2}$$

Cosine Similarity

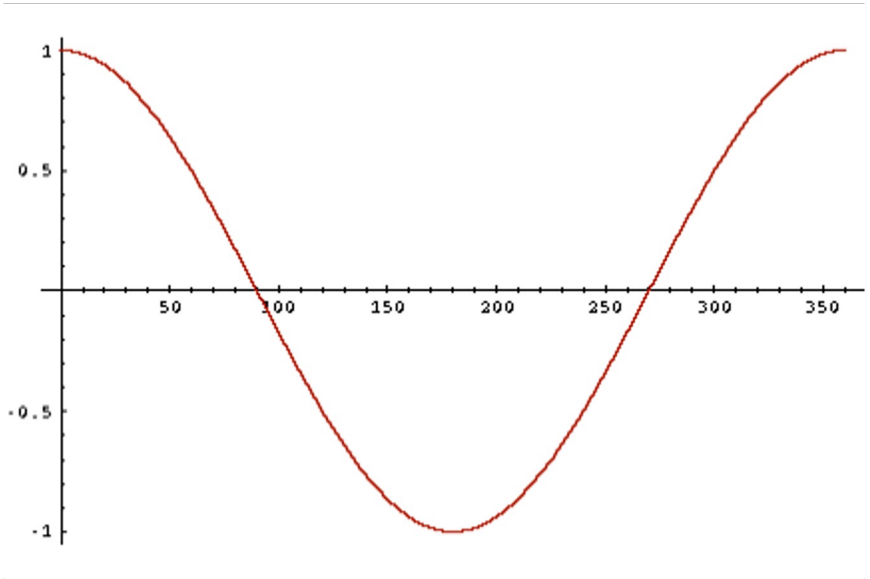
- Length-normalized dot product is the same as the *cosine of the angle between the two vectors*

$$\text{cosine}(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

- Cosine similarity measures the cosine of the angle between two vectors

Cosine Similarity

- -1: vectors point in opposite directions
- +1: vectors point in same directions
- 0: vectors are orthogonal



- **In our case:** since counts or frequencies are non-negative, cosine similarity is always between 0 and 1

Cosine Similarity: example

- Which pair of words is more similar?

	pie	data	computer
cherry	442	8	2
digital	5	1683	1670
information	5	3982	3325

$$\text{cosine}(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

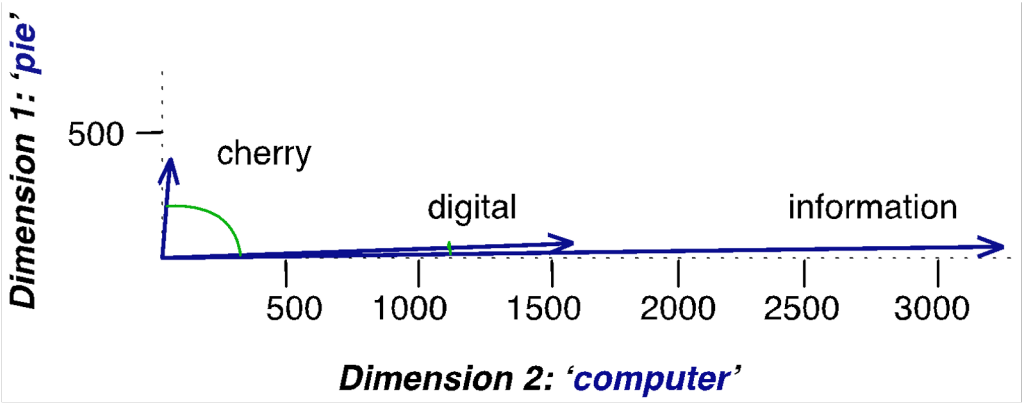
$$\text{cosine}(\text{cherry, information}) = \frac{442\times 5 + 8\times 3982 + 2\times 3325}{\sqrt{442^2+8^2+2^2}\sqrt{5^2+3982^2+3325^2}} = .017$$

$$\text{cosine}(\text{digital, information}) = \frac{5\times 5 + 1683\times 3982 + 1670\times 3325}{\sqrt{5^2+1683^2+1670^2}\sqrt{5^2+3982^2+3325^2}} = .996$$

Cosine Similarity: example

- Which pair of words is more similar?

	pie	data	computer
cherry	442	8	2
digital	5	1683	1670
information	5	3982	3325



$$\text{cosine}(\text{cherry, information}) = \frac{442 \times 5 + 8 \times 3982 + 2 \times 3325}{\sqrt{442^2 + 8^2 + 2^2}\sqrt{5^2 + 3982^2 + 3325^2}} = .017$$

$$\text{cosine}(\text{digital, information}) = \frac{5 \times 5 + 1683 \times 3982 + 1670 \times 3325}{\sqrt{5^2 + 1683^2 + 1670^2}\sqrt{5^2 + 3982^2 + 3325^2}} = .996$$

Frequencies Are Not Enough!

- Words can occur together frequently, but is it good?
  - **Yes**: "sugar" appears a lot near "apricot"
  - **No**: function words like "it", "the", etc.

- We need specialized *weighting* ***beyond simple counting***
  - tf-idf (term frequency, inverse document frequency)
  - PMI (pointwise mutual information)

tf-idf

- A simple, but extremely robust weighting method; proposed by Salton and Buckley [Salton&Buckley, 1988]
- **Idea:** promote document-specific words, penalize non-specific words
- **tf-idf** is a product of two values:
  - **tf**: term frequency (often *log*-transformed)
  - **idf**: inverse document frequency

tf-idf

- Term frequency (tf) measures how frequently the word *t* appears in the document *d*

$$\mathrm{tf}_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{count}(t,d) & \text{if } \mathrm{count}(t,d) > 0 \\ 0 & \text{otherwise} \end{cases}$$

- Inverse document frequency (idf) is inversely proportional to the number of documents in a corpus that contain *t*

$$\mathrm{idf}_t = \log_{10}\left(\frac{N}{\mathrm{df}_t}\right)$$

Total # of docs in collection

# of docs with word *t*

- So, the tf-idf value for word *t* in document *d* is

$$w_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t$$

tf-idf

- *tf* values are computed per each word-document pair, whereas *idf* values are computed per each word across the entire document collection
- The value of *idf* does not depend on a particular document
- Example: $N = 10^6$

term	df(term)	idf(term)
Frodo	10000	2
Sam	1000	3
stab	100	4
the	1000000	1

tf-idf: exercise

- **Corpus**:

d_1: "Frodo accidentally stabbed Sam and then some orcs"

d_2: "Frodo was stabbing regular orcs but never stabbed super orcs – Uruk-Hais"

d_3: "Sam was having a barbecue with some friendly orcs"

tf-idf: exercise

- **Corpus**:

d_1: "Frodo accidentally stabbed Sam and then some orcs"

d_2: "Frodo was stabbing regular orcs but never stabbed super orcs – Uruk-Hais"

d_3: "Sam was having a barbecue with some friendly orcs"

- **Compute tf-idf for each word in "Frodo stabs orc":**

tf-idf: exercise

- **Corpus**:

d_1: "Frodo accidentally stabbed Sam and then some orcs"

d_2: "Frodo was stabbing regular orcs but never stabbed super orcs – Uruk-Hais"

d_3: "Sam was having a barbecue with some friendly orcs"

- **Compute tf-idf for each word in "Frodo stabs orc":** idf("Frodo") = $\log_{10}(3/2) = .176$

tf("Frodo", d_1) = 1

tf("Frodo", d_2) = 1

tf("Frodo", d_3) = 0

tf-idf("F.", d_1) = tf("Frodo",d_1) x idf("Frodo")=.176

…

tf-idf: exercise

- **Corpus**:

d_1: "Frodo accidentally stabbed Sam and then some orcs"

d_2: "Frodo was stabbing regular orcs but never stabbed super orcs – Uruk-Hais"

d_3: "Sam was having a barbecue with some friendly orcs"

- **Compute tf-idf for each word in "Frodo stabs orc":** idf("Frodo") = $\log_{10}(3/2) = .176$

tf("Frodo", d_1) = 1

tf("Frodo", d_2) = 1

tf("Frodo", d_3) = 0

tf-idf("F.", d_1) = tf("Frodo",d_1) x idf("Frodo")=.176

…

Pointwise Mutual Information (PMI)

- Alternative weighting to tf-idf
- **Idea:** ask whether a context word is particularly *informative* about the target word
  - How "informative" in terms of Information Theory
- Do the words in a pair co-occur more than if they were independent?

Pointwise Mutual Information (PMI)

- Do events X and Y co-occur more than if they were independent?

$$\mathrm{PMI}(X,Y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

  - So, PMI is defined from $-\infty$ to $+\infty$

- PMI can be applied to words [Church&Hanks, 1989]

$$\mathrm{PMI}(w_1,w_2) = \log_2 \frac{P(w_1,w_2)}{P(w_1)P(w_2)}$$

- Probabilities can be estimated using bigram and unigram language models

Positive Pointwise Mutual Information (PPMI)

- Negative values are problematic, i.e., things are co-occurring *less* than we expect by chance
- Unreliable without *enormous* corpora
  - Imagine w_1 and w_2 whose probability is each 10^{-6}
  - Hard to be sure $P(w_1,w_2)$ is significantly different than 10^{-12}
- So we just replace negative PMI values by 0
- Thus, **positive PMI (PPMI)** between w_1 and w_2 is:

$$\text{PPMI}(w_1, w_2) = \max\left(\log_2 \frac{P(w_1, w_2)}{P(w_1)P(w_2)}, 0\right)$$

PPMI: exercise

Consider a **term-context matrix** F with W rows and C columns; f_{ij} is number of times the word w_i occurs in context c_j

or co-occurs with the word c_j

Count(w,context)

	computer	data	pinch	result	sugar
apricot	0	0	1	0	1
pineapple	0	0	1	0	1
digital	2	1	0	1	0
information	1	6	0	4	0

PPMI: exercise

Calculate a P matrix with W rows and C columns; $p_{ij} = p(w_i, c_j)$

Count(w,context)

	computer	data	pinch	result	sugar
apricot	0	0	1	0	1
pineapple	0	0	1	0	1
digital	2	1	0	1	0
information	1	6	0	4	0

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

$$p(w_i) = \frac{\sum_{j=1}^{C} f_{ij}}{N}$$

$$N = \sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}$$

$$p(c_j) = \frac{\sum_{i=1}^{W} f_{ij}}{N}$$

p(w,context) **p(w)**

	computer	data	pinch	result	sugar	
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

PPMI: exercise

Calculate the PPMI matrix with W rows and C columns; $ppmi_{ij}$ =PPMI(w_i, c_j)

	p(w,context)					p(w)
	computer	data	pinch	result	sugar	
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

$$\text{PPMI}(w_1, w_2) = \max\left(\log_2 \frac{P(w_1, w_2)}{P(w_1)P(w_2)}, 0\right)$$

PPMI(w,context)

	computer	data	pinch	result	sugar
apricot	-	-	2.25	-	2.25
pineapple	-	-	2.25	-	2.25
digital	1.66	0.00	-	0.00	-
information	0.00	0.57	-	0.47	-

PPMI: exercise

Calculate the PPMI matrix with W rows and C columns; $ppmi_{ij}$ =PPMI(w_i, c_j)

p(w,context)

	computer	data	pinch	result	sugar	**p(w)**
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

$$\text{PPMI}(w_1, w_2) = \max\left(\log_2 \frac{P(w_1, w_2)}{P(w_1)P(w_2)}, 0\right)$$

PPMI(w,context)

	computer	data	pinch	result	sugar
apricot	-	-	2.25	-	2.25
pineapple	-	-	2.25	-	2.25
digital	1.66	0.00	-	0.00	-
information	0.00	0.57	-	0.47	-

Problems with PMI and PPMI

- PMI overestimates infrequent events
  - Very rare words have high PMI values
- Solutions:
  - Adjust the probabilities of rare words
  - Use Laplace smoothing

PPMI: Probability Adjustment

- Raise the *context* probabilities to α=.75 (Levy et al., 2015)

$$\text{PPMI}_\alpha(w,c) = \max(\log_2 \frac{P(w,c)}{P(w)P_\alpha(c)}, 0)$$

$$P_\alpha(c) = \frac{count(c)^\alpha}{\sum_c count(c)^\alpha}$$

- For a rare context, $P_\alpha(c) > P(c)$
- Consider two events: c(a)=99, c(b)=1 with N=100

$$P(a) = 0.99; P(b) = 0.01$$

$$P_\alpha(a) = \frac{99^{.75}}{99^{.75}+1^{.75}} = .97 \qquad P_\alpha(b) = \frac{1^{.75}}{99^{.75}+1^{.75}} = .03$$

Problems with Sparse Vector Space Model

- Most values are zeros (that is, *sparse* model)
  - Not every programming environment includes an efficient sparse matrix representation
- Difficult to deal with *out of vocabulary* (OOV) words
  - "This bar serves fresh **jabuticaba** juice."
- Have very large dimensionality of vectors in large text corpora
- In practice, *dense* vectors show better results
  - …and they will be covered later in the course!

Q&A

Resources and References

[Jurafsky&Martin, 2022] Jurafsky and Martin. Speech and Language Processing, Prentice Hall, third edition https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf

[Hill et al., 2015] Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation. Computational Linguistics, 41(4):665–695. https://aclanthology.org/J15-4004

[Harris, 1954] Harris, Z. (1954). Distributional structure. Word, 10(23): 146-162. https://www.tandfonline.com/doi/pdf/10.1080/00437956.1954.11659520

[Salton&Buckley, 1988] Gerard Salton, Christopher Buckley, Term-weighting approaches in automatic text retrieval, Information Processing & Management, Volume 24, Issue 5, 1988, Pages 513-523, ISSN 0306-4573,

https://doi.org/10.1016/0306-4573(88)90021-0.

[Church&Hanks, 1989] Kenneth Ward Church and Patrick Hanks. 1989. Word Association Norms, Mutual Information, and Lexicography. In 27th Annual Meeting of the Association for Computational Linguistics, pages 76–83, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

https://aclanthology.org/P89-1010/

[Levy et al.. 2015] Levy, Omer & Goldberg, Yoav & Dagan, Ido. (2015). Improving Distributional Similarity with Lessons Learned from Word Embeddings. Transactions of the Association for Computational Linguistics. 3. 211-225. 10.1162/tacl_a_00134.

https://aclanthology.org/Q15-1016.pdf

**Credits

The slides of this part of the course are the result of a personal reworking of the slides and of the course material from different sources:

1. The NLP course of Prof. Roberto Navigli, Sapienza University of Rome
2. The NLP course of Prof. Simone Paolo Ponzetto, University of Mannheim, Germany
3. The NLP course of Prof. Chris Biemann, University of Hamburg, Germany
4. The NLP course of Prof. Dan Jurafsky, Stanford University, USA