A tutorial based on:

- “Distributional Semantics” in CS388 NLP at UTexas
  (Baroni & Boleda)
- Tutorials by Stephanie Evert & Alessandro Lenci
- Lectures by Alessandro Lenci at SNS Pisa

Distributional Semantics

Ludovica Pannitto, Ph.D.
ludovica.pannitto@unibo.it
lilec.lab@unibo.it

Outline

Theory session (1h)

- The Distributional Hypothesis
- From corpus to vectors
- What DSMs are good for

Demo (20mins)

- Exploring the corpus
- The co-occurrence matrix
- Nearest neighbours
- Evaluation on a task

The core idea

“You shall know a word
by the company it keeps”

— J.R. Firth, 1957

The meaning of a word does not exist in the abstract:
it **emerges from the contexts** in which the word appears.

An experiment

What does *ongchoi* mean?

An experiment

What does *ongchoi* mean?

(take a moment to think)

Some contexts. . .

- *"**Ongchoi** is delicious sautéed with garlic"*
- *"**Ongchoi** is superb over rice"*
- *"**Ongchoi leaves** with salty sauces"*

Some contexts. . .

- *"**Ongchoi** is delicious sautéed with garlic"*
- *"**Ongchoi** is superb over rice"*
- *"**Ongchoi leaves** with salty sauces"*

What does this tell you?

The same environment as other known words

Contexts of *ongchoi*

- delicious sautéed with garlic
- superb over rice
- leaves with salty sauces

Similar contexts (known words)

- *spinach* sautéed with garlic over rice
- *chard* stems and leaves are delicious
- *collard greens* and other leafy greens

Conclusion

Ongchoi is a **leafy green vegetable** — we didn't need a definition, just its **distributional neighbours**.

The Distributional Hypothesis

Formulation

Words with **similar distributions** have **similar meanings**.

Harris (1954)

Difference of meaning correlates with difference of distribution in texts.

Wittgenstein (PI §43)

"The meaning of a word is its use in the language."

The Distributional Hypothesis

Formulation

Words with **similar distributions** have **similar meanings**.

Harris (1954)

Difference of meaning correlates with difference of distribution in texts.

Wittgenstein (PI §43)

"The meaning of a word is its use in the language."

Operational question

How do we make this intuition **computationally measurable**?

Two versions of the hypothesis

Weak DH

Distributional properties **reflect** meaning.

*We use distribution as **evidence** for meaning.*

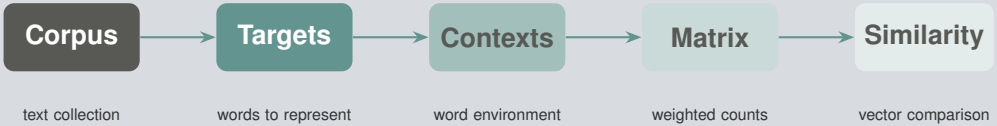
Strong DH

Cognitive representations are **built** (at least in part) through distributional experience.

*Meaning **is** (also) distributional.*

In both cases: distribution is a **privileged tool** for studying lexical meaning.

Building a DSM



Every choice in this pipeline is a **linguistic choice**, not just a technical one.

The corpus

Definition

A collection of texts in digital form, used as a **source of linguistic knowledge**.

Relevant dimensions

- Size (how many tokens?)
- Domain (news, literature, web. . .)
- Pre-processing: tokenisation, lemmatisation, POS tagging

For today

- Italian Wikipedia
- $\sim$1 million tokens, lemmatised

Note

A news corpus will give different results than a fiction corpus.
The corpus *is* the model.

Target words

Definition

The words we want to **represent** in the model.

Selection criteria

- A predefined list (from an experiment, a dictionary…)
- The *N* most frequent words in the corpus
- Only certain POS (nouns, verbs, adjectives)
- (Typically) a minimum frequency threshold

Why frequency matters

More frequent targets give more contexts — but very frequent words (*be*, *have*) carry little specific meaning.

Contexts

Definition

The words (or structures) that **surround** the target in the corpus — the *dimensions* of the vector space.

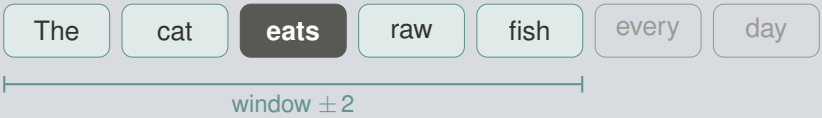
Where do we look?

- **Lexical window**: the *N* neighbouring words
- Syntactic collocates (subject, object. . .)
- Within a construction

Design questions

- How large is the window?
- Left and right symmetric?
- Include function words?
- Does distance matter?

Window-based contexts



Small window (±1–2)

Captures **tight syntactic** relations: subject–verb, modifier–noun

Large window (±5–10)

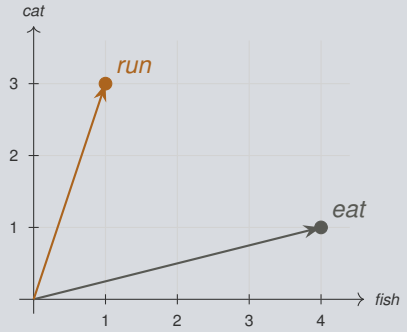
Captures **topical similarity**: words from the same domain

The co-occurrence matrix

From the corpus, we count how often each target appears near each context word:

	fish	*cat*
eat	4	1
run	1	3

Each word becomes a **vector**:
eat = (4, 1) *run* = (1, 3)



With many contexts each word lives in a **very high-dimensional** space — the geometric intuition stays the same.

From text to raw counts

Mini corpus

The **dog** **is** in the **garden** and the **dog** **barked** loudly near its owner. The **cat** **is** on the wall and the **cat** **meowed** softly at night. A **car** **is** in the street and the **car** moved quickly past the gate. The **child** **is** near the dog and the **child** laughed happily in the **garden**.

Suppose our **target words** are *dog*, *cat*, *car*, and *child*. We count how often they appear near context words such as *be*, *garden*, *barked*, and *meowed*.

Raw co-occurrence matrix

	be	*gard.*	*bark.*	*meow.*
dog	2	2	1	0
cat	2	0	0	1
car	2	0	0	0
child	2	1	0	0

Each row is a target word; each column is a context.

The problem with raw frequencies

The problem

The word ***be*** is very frequent: it appears near almost everything. With raw counts, all words look similar because they share this uninformative context.

Example — top contexts of *dog* (raw frequency)

Context	Freq.	Informative?
be	1,240	No — appears near everything
bark	87	Yes — specific to *dog*
stray	34	Yes
leash	29	Yes

PPMI — The intuition

Core idea

A context is informative if it appears with a target **more often than expected by chance**.

PPMI asks:

- How frequent is the target?
- How frequent is the context?
- How often do they co-occur?
- Is the association **stronger than chance**?

Interpretation

Pair	Raw freq.	PPMI
dog – be	high	low
dog – bark	lower	high

PPMI — The formula

Pointwise Mutual Information

$$\mathrm{PMI}(w, c) = \log \frac{P(w, c)}{P(w)\,P(c)}$$

Positive PMI

$$\mathrm{PPMI}(w, c) = \max(\mathrm{PMI}(w, c), 0)$$

Meaning of the terms

- $P(w, c)$: probability of seeing w and c together
- $P(w)$: probability of the target word
- $P(c)$: probability of the context word

Why “positive”?

If $\mathrm{PMI}(w, c) < 0$, we set it to zero.

PPMI — From counts to a weighted matrix

Raw co-occurrence counts

	be	*bark*
dog	50	12
car	48	0

Raw counts make *be* look important simply because it is everywhere.

After PPMI weighting

	be	*bark*
dog	0.1	2.4
car	0.0	0.0

Now the salient context is visible: *dog* is strongly associated with *bark*.

PPMI turns a frequency matrix into a **more informative semantic representation**.

A typical PPMI matrix

	bark	*yelp*	*engine*	*garden*
dog	2.6	2.2	0.0	1.1
puppy	2.4	2.0	0.0	0.8
wolf	1.8	1.5	0.0	0.0
car	0.0	0.0	2.7	0.0
truck	0.0	0.0	2.4	0.0

Problem 1: sparsity

Many cells are exactly zero, because most words do not have strong associations with most contexts.

Problem 2: correlated columns

Columns like *bark* and *yelp* carry almost the same information: they are both high for *dog*, *puppy*, *wolf*, and zero elsewhere.

This is why PPMI is informative, but still often **sparse and redundant**.

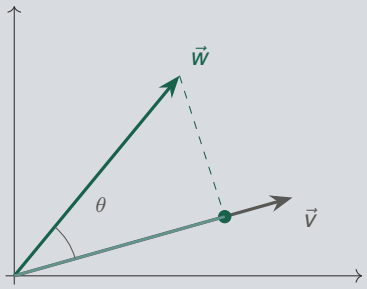
How similar are two words?

Cosine similarity

$$\cos(\theta) = \frac{\vec{v} \cdot \vec{w}}{\|\vec{v}\| \, \|\vec{w}\|}$$

Think of it as the **projection of one vector onto the direction of the other**, after removing the effect of vector length.

- We project $\vec{w}$ onto the direction of $\vec{v}$, then we normalize by the lengths of the two vectors
- So cosine measures **directional similarity**, not magnitude



If the angle θ is small, the projection is large, so cosine is high.

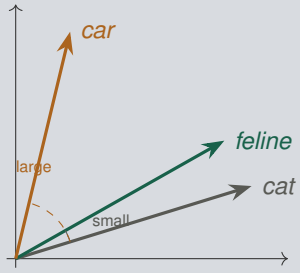
How do we interpret cosine?

Geometric intuition

Cosine depends on the **angle** between two vectors.

- Same direction $\Rightarrow$ cosine ≈ 1
- Large angle $\Rightarrow$ cosine is lower

In DSMs, two words are similar when their vectors point in a similar direction, that is, when they occur in **similar contexts**.



From explicit to implicit vectors

Our current matrix

So far, our vectors are **explicit**: each dimension corresponds to a context word.

This is useful because the space is interpretable, but it also creates some problems.

- very high dimensionality
- many dimensions are sparse or redundant
- correlated contexts may spread information across many axes

Main issue

In very high-dimensional spaces, distances and similarities become harder to interpret reliably.

Goal

We would like a **smaller, denser** vector space that keeps the important distributional structure.

Dimensionality reduction with SVD

Singular Value Decomposition

A matrix factorization technique that rewrites the same data using **new axes** that better capture its structure.

The idea is:

- find more informative dimensions
- keep only the top k dimensions
- obtain an approximation of the original matrix

Result

The vectors become **implicit**: each dimension is now a *latent feature*, not a directly observable context word.

Pros

- denser vectors
- faster computations
- often a better semantic space

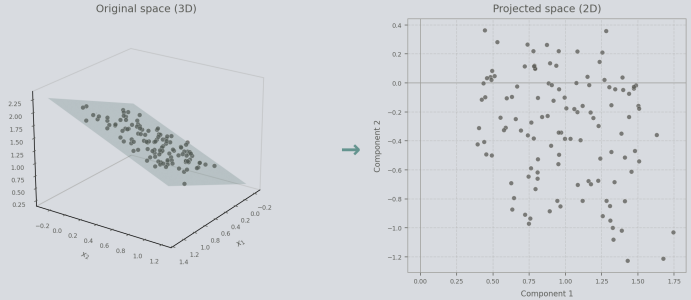
Con

- less interpretability

SVD as a graphical idea

Geometric intuition

SVD finds the plane that captures the **most variance** in the data, then projects every point onto it — discarding only the distances from that plane.



word2vec

What is it?

word2vec is a family of algorithms for learning **word embeddings** with a predictive, neural-inspired approach.

Instead of counting co-occurrences directly, the model is trained to solve a *fake task* based on context prediction.

- **CBOW**: predict the target word from its context
- **Skip-gram**: predict context words from the target word

Key idea

We are not really interested in the prediction task itself.

We want the **internal representation** learned by the model: the vector associated with each word.

This is why word2vec is often described as **representation learning**.

Skip-gram intuition

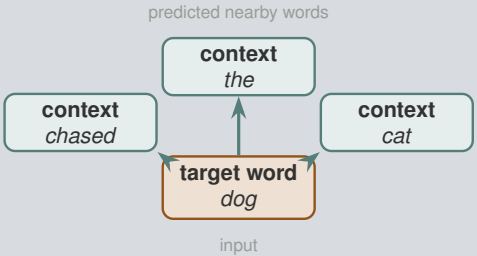
Example sentence:

the dog chased the cat in the garden

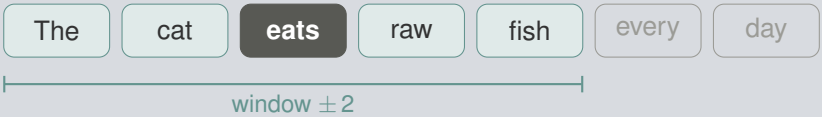
If the target is *dog*, the model should assign high probability to nearby words such as *the*, *chased*, *cat*.

Consequence

If two words have similar contexts, the model is pushed to learn similar vectors for them.



Skip-gram: training pairs



Sliding the window over the sentence generates **(target, context) pairs**:

target	context
The	*cat*
The	*eats*
cat	*The*
cat	*eats*
cat	*raw*

target	context
eats	*The*
eats	*cat*
eats	*raw*
eats	*fish*
raw	*cat*

target	context
raw	*eats*
raw	*fish*
fish	*eats*
fish	*raw*
fish	*every*

The model is trained to **predict the context given the target** — on millions of such pairs.

Skip-gram as a small neural network

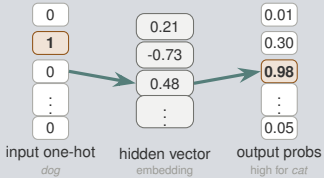
Architecture

A simple feed-forward network:

- **Input**: one target word
- **Hidden layer**: a small dense vector
- **Output**: scores for possible context words

The input is typically a one-hot vector: all zeros except one position for the current target word.

The hidden layer is exactly the compact representation we want to learn.



Important point: the training objective is context prediction, but the useful output for us is the hidden vector.

What does the hidden layer learn?

If two words have **similar contextual behaviour**, the network needs similar internal states to make similar predictions.

- similar contexts $\Rightarrow$ similar hidden states
- similar hidden states $\Rightarrow$ similar embeddings

Interpretation

The hidden layer becomes a **compact semantic representation**: words that behave similarly in context end up close in the vector space.

Take-away

If *dog* and *cat* support similar context predictions, the network is encouraged to place them in similar parts of the vector space, and this depends on their contexts.

From word2vec to transformers

Same general intuition

Modern models such as **BERT** still learn meaning from **distributional patterns in context**.

The main difference is what counts as a representation:

- **word2vec**: one stable embedding per type
- **transformers**: the token vector depends on the whole sentence

Useful parallel

The vectors are not given in advance: they are **learned as part of a prediction task**.

DSMs as a linguistic tool

A DSM is not an end in itself: it is a **tool for testing hypotheses** about lexical meaning.

Core question:
Can a model built purely on distribution approximate human performance on linguistic tasks?

Similarity	**Synonymy**	**Relations**
Semantically close words?	Which word is a synonym?	Hyponymy, meronymy, co-ord.?

Task: Word similarity

Research question

Can the model approximate **human similarity judgements** for word pairs?

Dataset (SimLex-999, WS-353. . .)

- Word pairs with human rating (0–10)

pair	score
cat — feline	8.9
cat — dog	4.2
cat — car	0.3

Procedure

1. Represent each word as a vector
2. Compute cosine similarity for each pair
3. Compare model scores with human ratings

$\Rightarrow$ Measure *correlation* between the two score series (Spearman's ρ)

Task: Synonymy

Research question

Given a target word, does the model correctly identify its **synonym**?

Example — TOEFL test

Target: *levied*

(a) **assessed** ← correct synonym
(b) believed
(c) requested
(d) correlated

Procedure: represent all alternatives as vectors and pick the one with the **highest cosine similarity** to the target.

Task: Semantic relations (BLESS)

Research question

Which **semantic relations** emerge naturally in distributional space?

Relations in the dataset

- **Hyponymy:** *dog* $\rightarrow$ *animal*
- **Meronymy:** *dog* $\rightarrow$ *paw*
- **Co-ordination:** *dog* $\rightarrow$ *cat*
- **Attribute:** *dog* $\rightarrow$ *loyal*

Open question

Does the model capture all relations equally? Or do some emerge more easily than others?

(Spoiler: co-ordination emerges well; hyponymy less so)

Evaluating the model

Regression

Model scores **correlate** with human judgements?

$\Rightarrow$ Spearman correlation
Used for: word similarity

Classification

Model assigns the **correct category**?

$\Rightarrow$ Accuracy
Used for: synonymy, relations

Don't forget the baseline

Before celebrating: how well does a **random baseline** perform?
The model's value only emerges in comparison.

The linguist's perspective

Every choice in the pipeline reflects an implicit **linguistic hypothesis**.

Choice	Implicit hypothesis
Corpus (Wikipedia)	meaning emerges from encyclopaedic text
Window $\pm$5	relevant context is local
Nouns + verbs only	lexical POS carry more semantic info
PPMI	salient co-occurrences are more informative

*Analysing results also means **reflecting on these choices**.*

Comparing alternative models

Experimental logic

We can build **different DSMs** by changing design choices: corpus, context definition, weighting, dimensionality reduction, and so on.

Then we evaluate each model on the same task:

- similarity
- synonymy
- semantic relations

Interpretation

If one set of choices leads to **better performance** on a task, we can infer that those choices capture information that is **more useful for that task**.

Linguistic payoff

Evaluation is not only model comparison: it also helps us understand which contextual assumptions are more informative.

DEMO

1. **Explore the corpus**
   How the data is organised; lemmas, POS, frequencies
2. **Inspect the matrix**
   Most salient contexts (PPMI) for some target words
3. **Nearest neighbours**
   Which words are closest to a given word?
4. **Similarity on WordSim-353**
   Comparing model scores against human judgements

Let’s go!