NLP: Vectorization, Embeddings and more

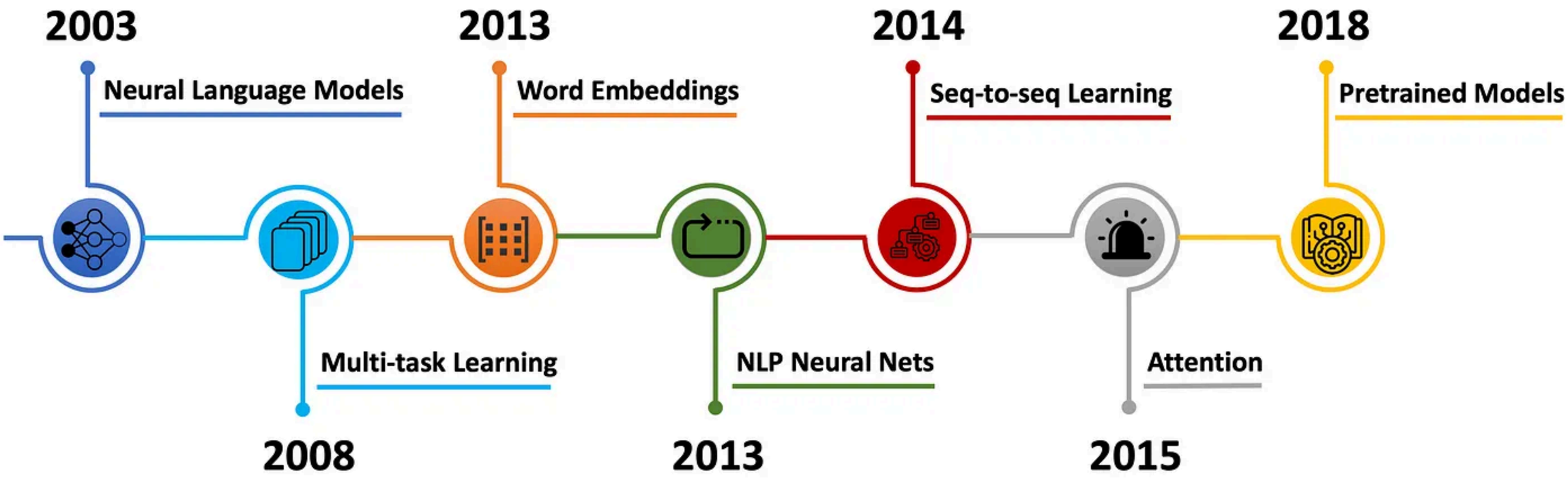
GenAI 2025

Linda Belkessa
Computer Science State Eng & MsC, PhD candidate @GRETTIA, CLEAR-DOC Fellow
linda.belkessa@univ-eiffel.fr

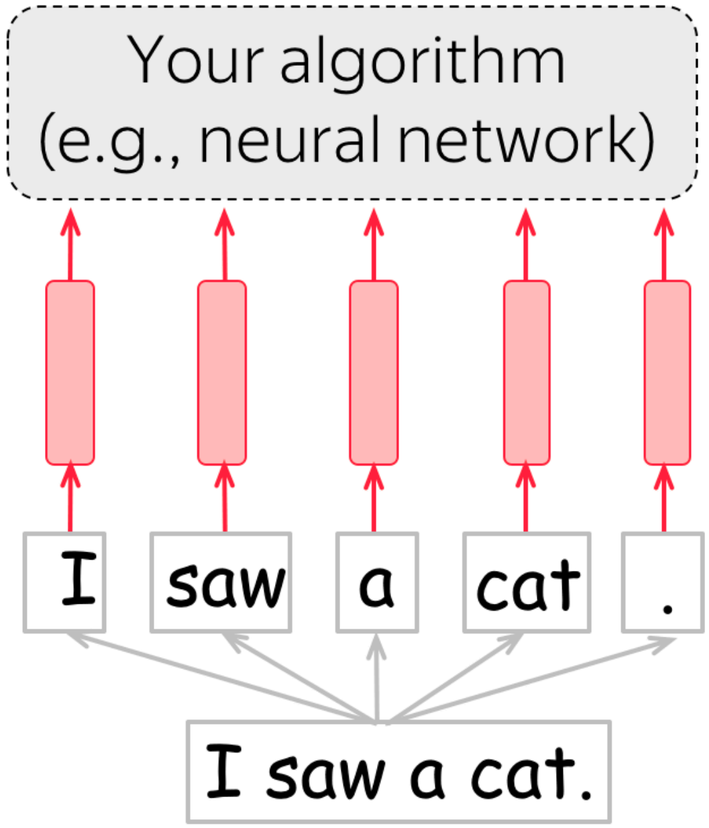
https://www.linkedin.com/in/linda-belkessa/

Lynda-Starkus

History of NLP (TALN)



How to think about words



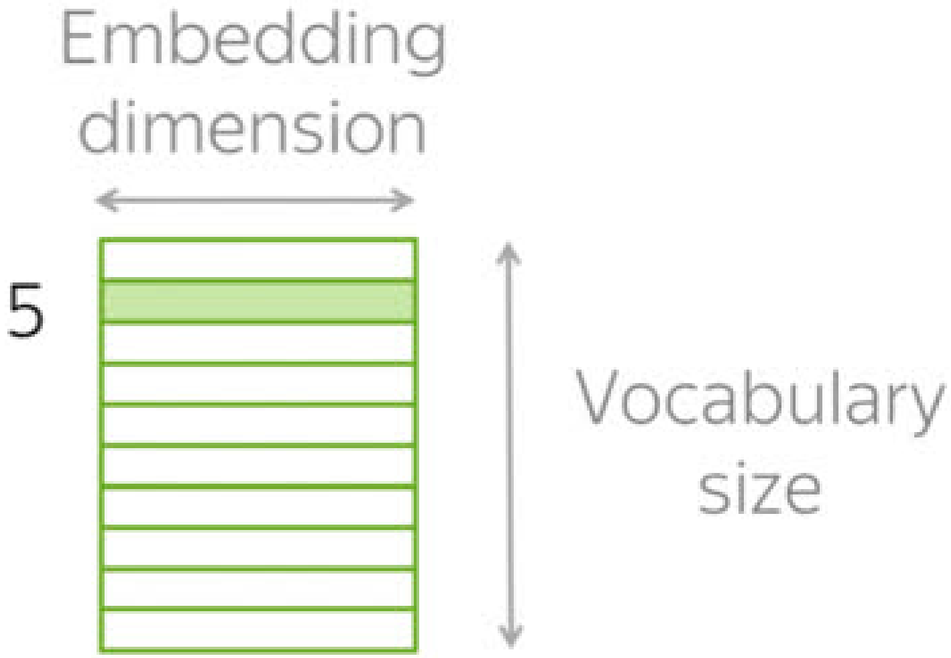
Any algorithm for solving a task

Word representation - vector
(input for your model/algorithm)

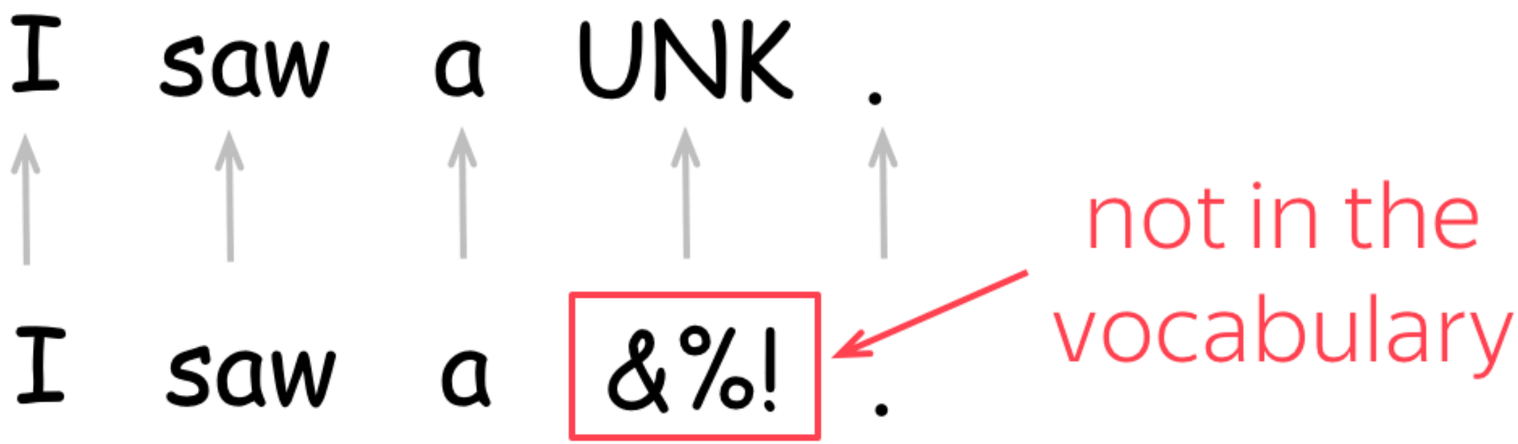
Sequence of tokens

Text (your input)

Look-up Table (Vocabulary)

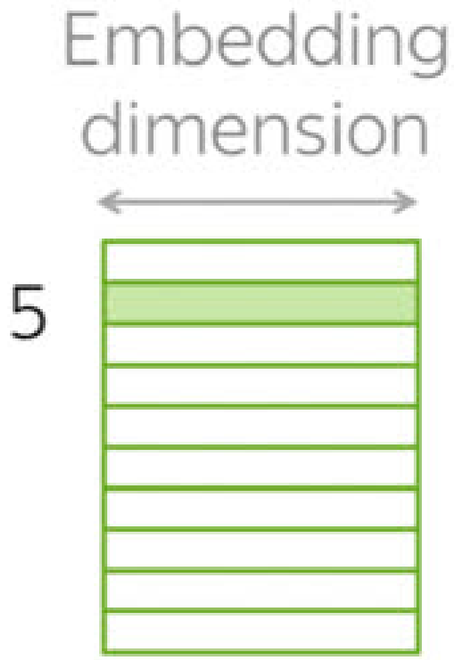


Look-up Table (Vocabulary)



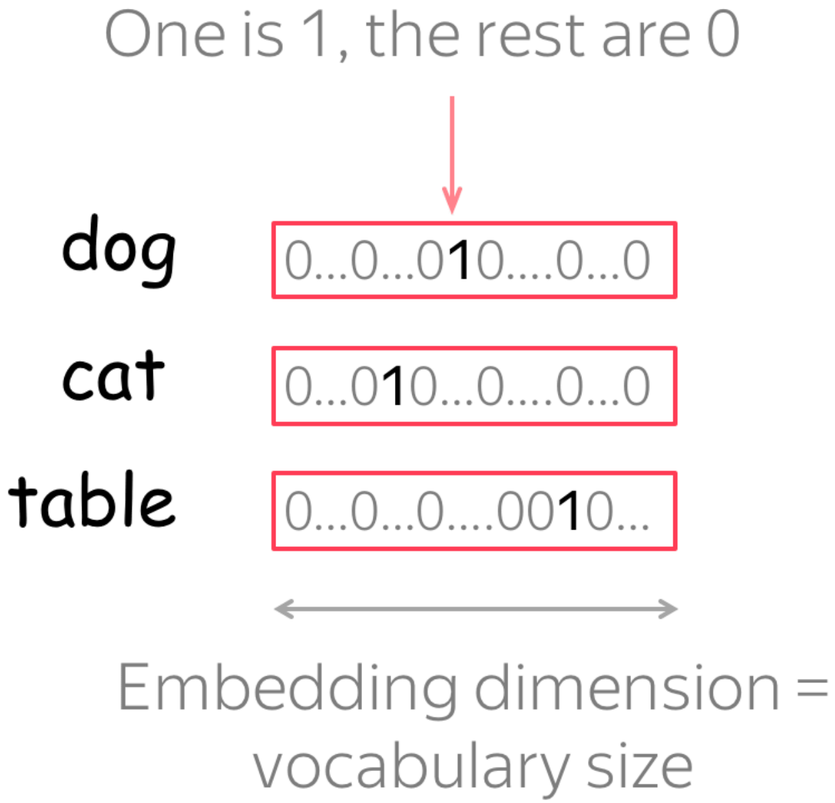
What if a word is NOT in the vocabulary ?

Goal of NLP



How to represent these word vectors (embeddings) ?

Solution 1 (Bag of Words) : One-Hot Encoding



Advantage: simple to implement, unique values

Drawbacks: Large vocabulary → Large calculation. No relationship between words (lack of meaning)

Solution 1: One-Hot Encoding

Drawbacks: Large vocabulary → Large calculation. No relationship between words **(lack of meaning)**

How to capture “Words meaning and context ?”

Solution 2 (TF-IDF): Distributional Semantics

Do you know what Awamori means ?

Solution 2: Distributional Semantics

Do you know what Awamori means ?

- (Context 1) I bought a bottle of Awamori in Okinawa?
- (Context 2) If you drink lots of Awamori you get drunk?
- (Context 3) Awamori is made out of rice ?

Solution 2: Distributional Semantics

Do you know what Awamori means ?

Solution 2: Distributional Semantics

How did your brain find it ?

	Context 1	Context 2	Context 3	.....	Context n
Water	1	0	0	.....	
Alcohol	1	1	0	.....	
Noodles	0	0	1	.....	
Oil	1	0	0	.....	
Awamori	1	1	1	.....	

- (Context 1) I bought a bottle of _____ in Okinawa?
- (Context 2) If you drink lots of _____ you get drunk?
- (Context 3) _____ is made out of rice ?

Solution 2: Distributional Semantics

How did your brain find it ?

	Context 1	Context 2	Context 3	.....	Context n
Water	1	0	0	.....	
Alcohol	1	1	0	.....	
Noodles	0	0	1	.....	
Oil	1	0	0	.....	
Awamori	1	1	1	.....	

- (Context 1) I bought a bottle of _____ in Okinawa?
- (Context 2) If you drink lots of _____ you get drunk?
- (Context 3) _____ is made out of rice ?

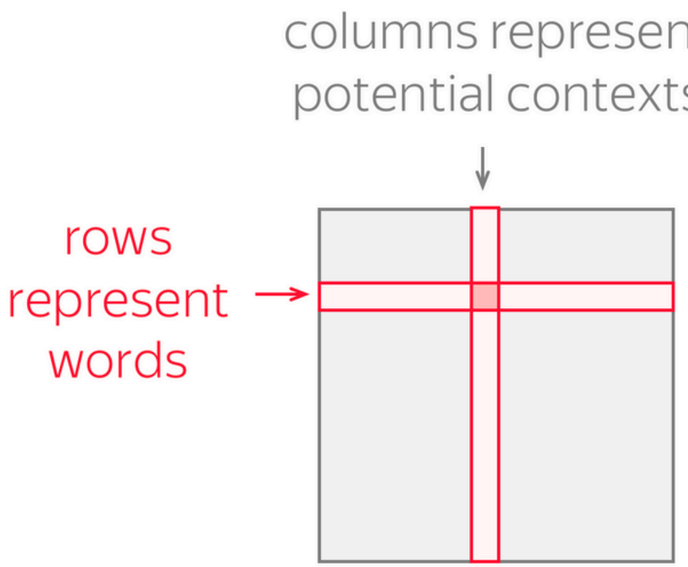
Solution 2: Distributional Semantics

How to show a machine how find it ?

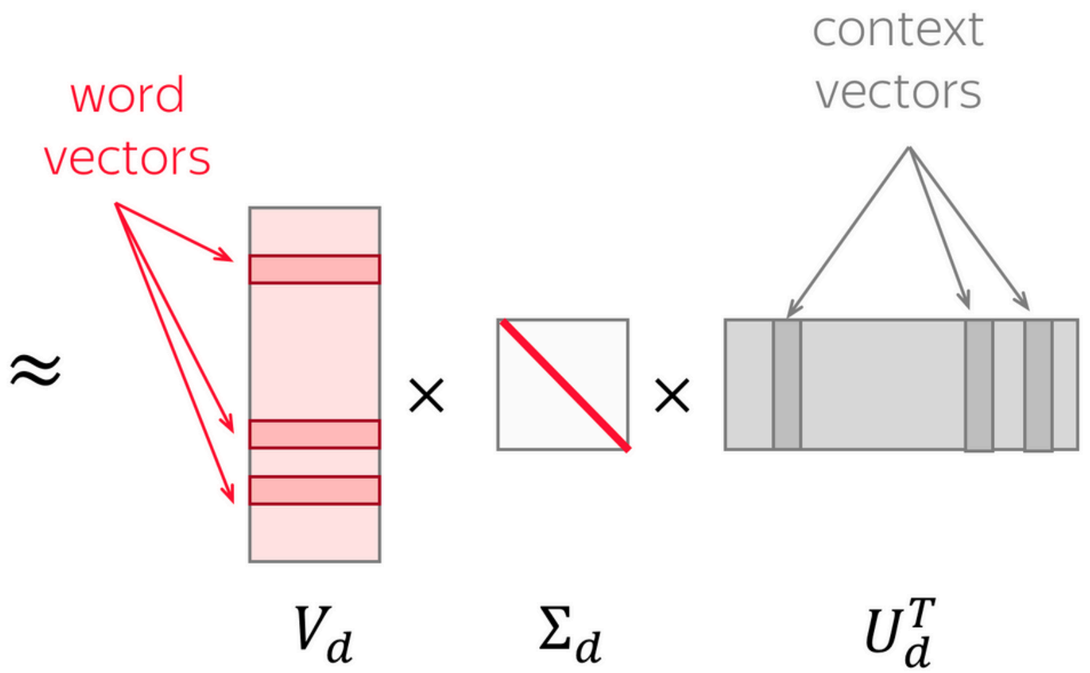
Main idea: We need to put information about word contexts into word representation.

- Count-Based Methods
- Positive Pointwise Mutual Information (PPMI)
- Word2Vec
- GloVe

Solution 2: Count-Based Methods

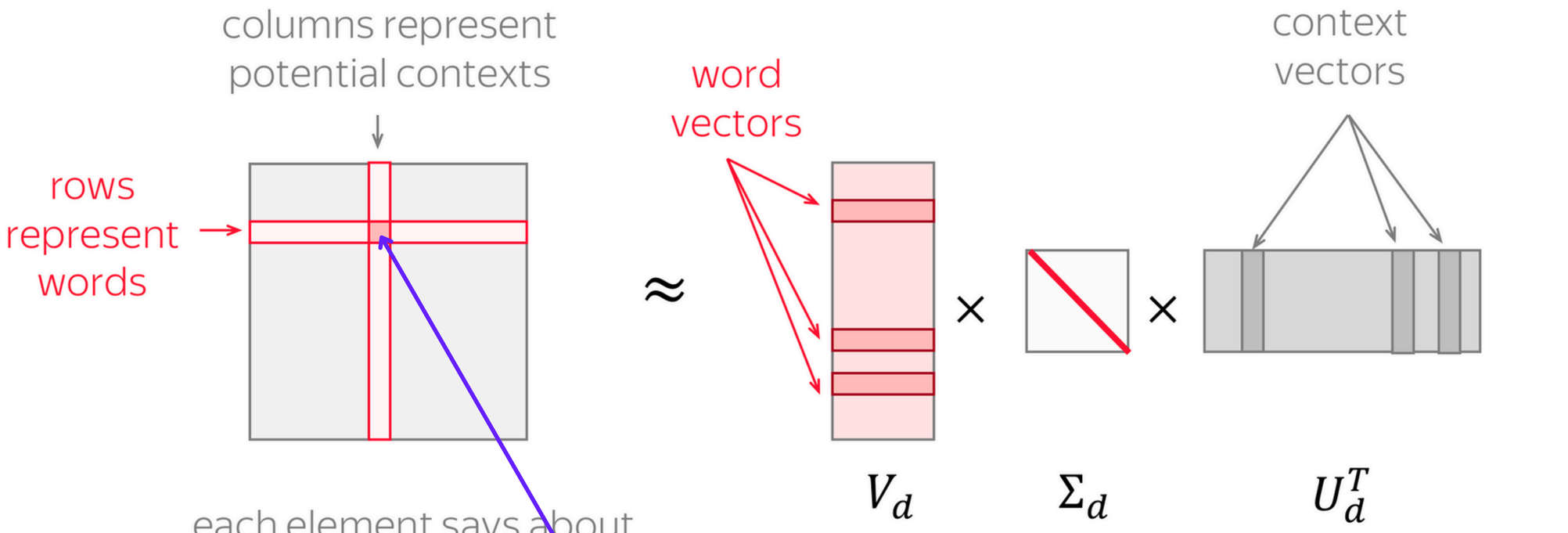


each element says about
the association between a
word and a **context**



Reduce dimensionality:
Truncated Singular Value Decomposition (SVD)

Solution 2: Count-Based Methods

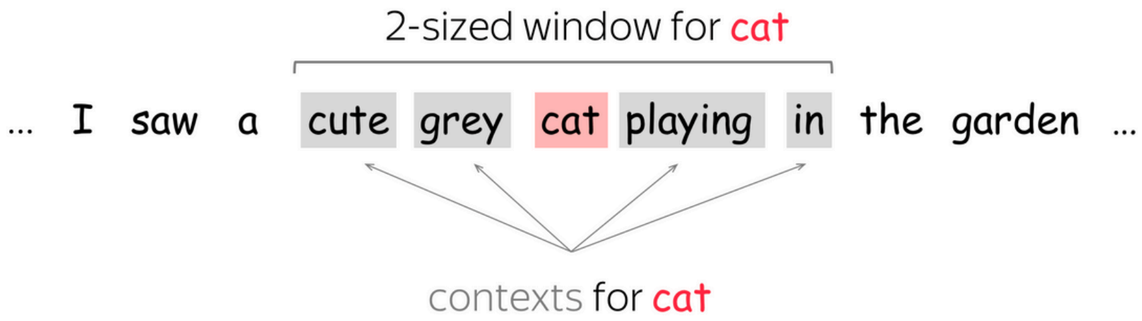


Similarity function (simple count, cosine similarity, ….etc)

Solution 2: Count-Based Methods

Similarity function (simple count, cosine similarity, ….etc)

Simple: Co-Occurence Counts



N(w, c) – number of times word w appears in context c

Positive Pointwise Mutual Information (PPMI)

- PPMI(w, c) = max(0, PMI(w, c)),

where

$$\text{PMI}(w, c) = \log \frac{P(w, c)}{P(w)P(c)} = \log \frac{\text{N}(w, c)|(w, c)|}{\text{N}(w)\text{N}(c)}$$

Is there another clever way ?

Solution 1: One-Hot Encoding

Solution 2: Distributional Semantics

Static word-context matrix

Never seen before words (or contexts) ?

Is there another clever way ?

Solution 1: One-Hot Encoding

Solution 2: Distributional Semantics

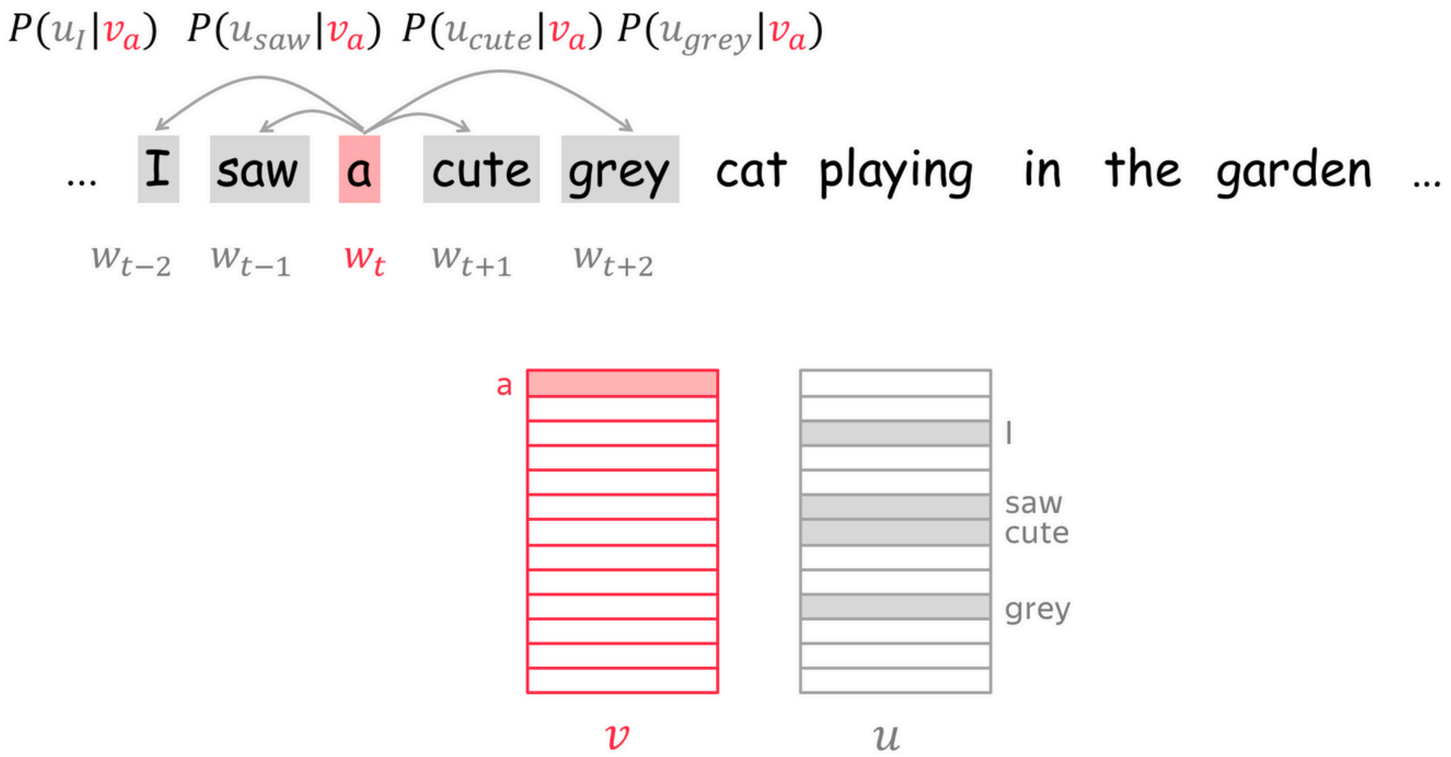
Static word-context matrix

Never seen before words (or contexts) ?

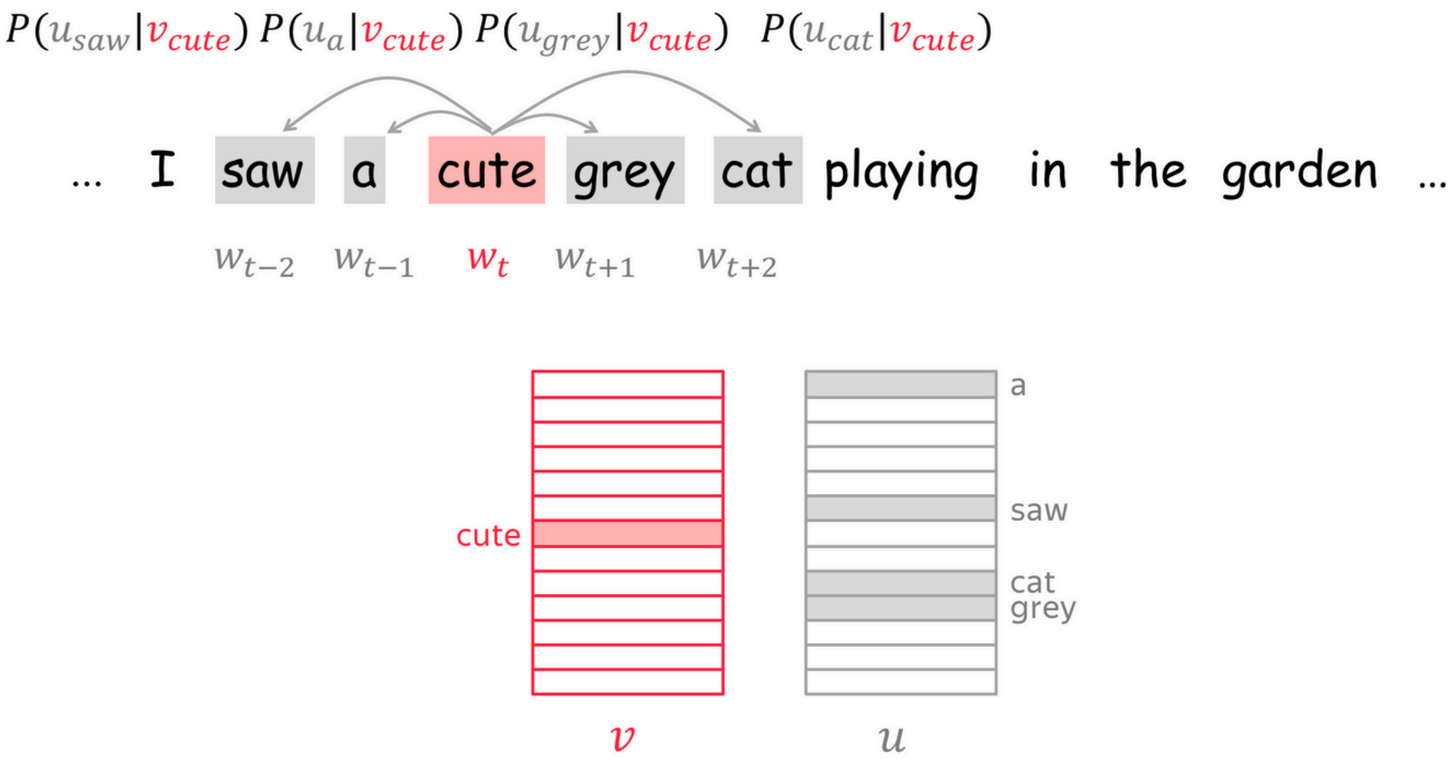
Solution 3: Word2Vec

How: Learn word vectors by teaching them to predict contexts

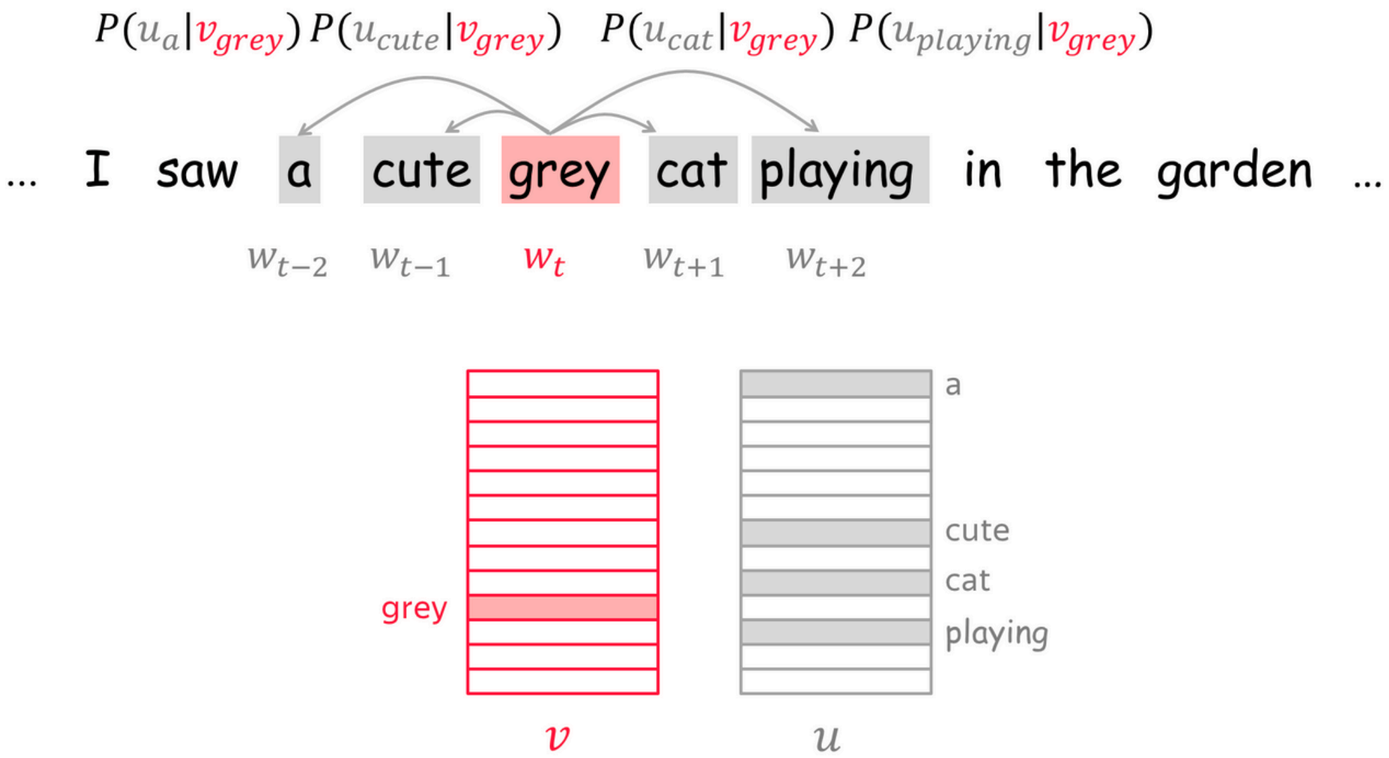
Solution 3: Word2Vec



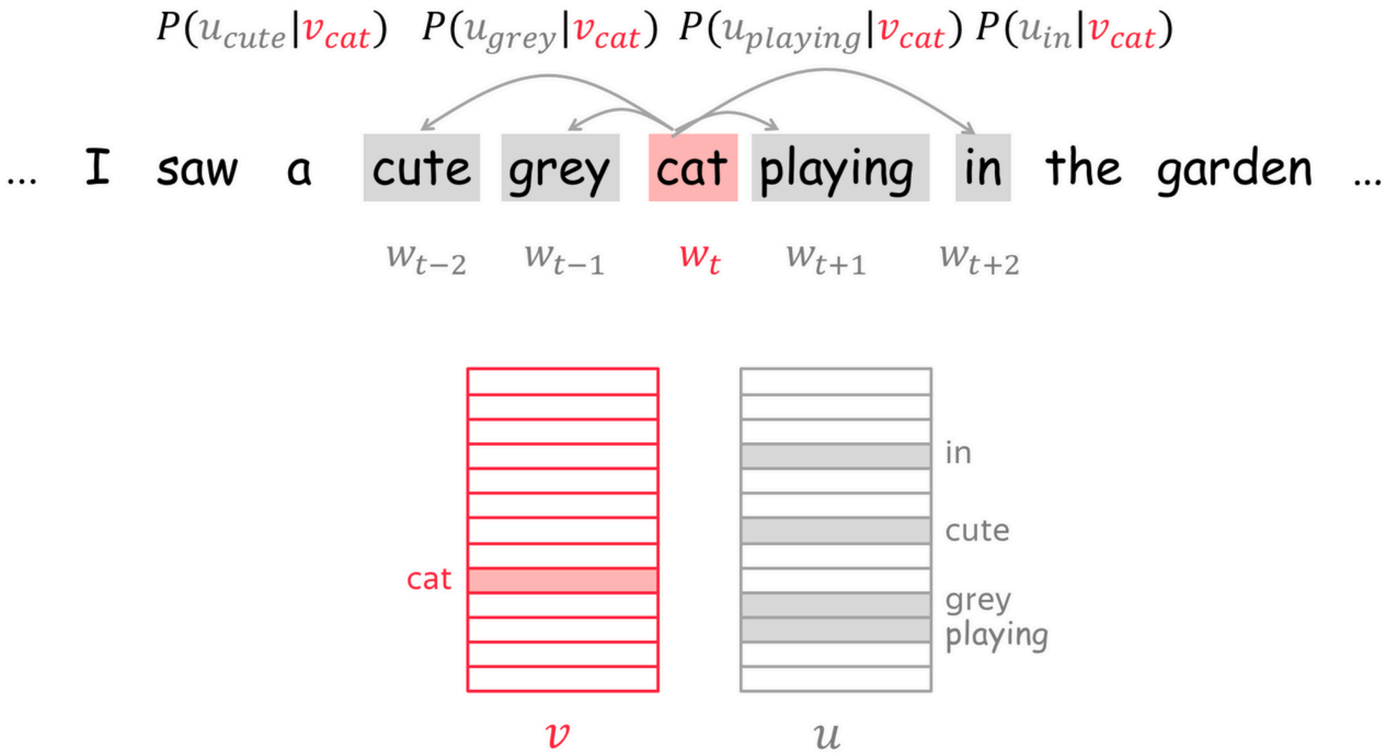
Solution 3: Word2Vec



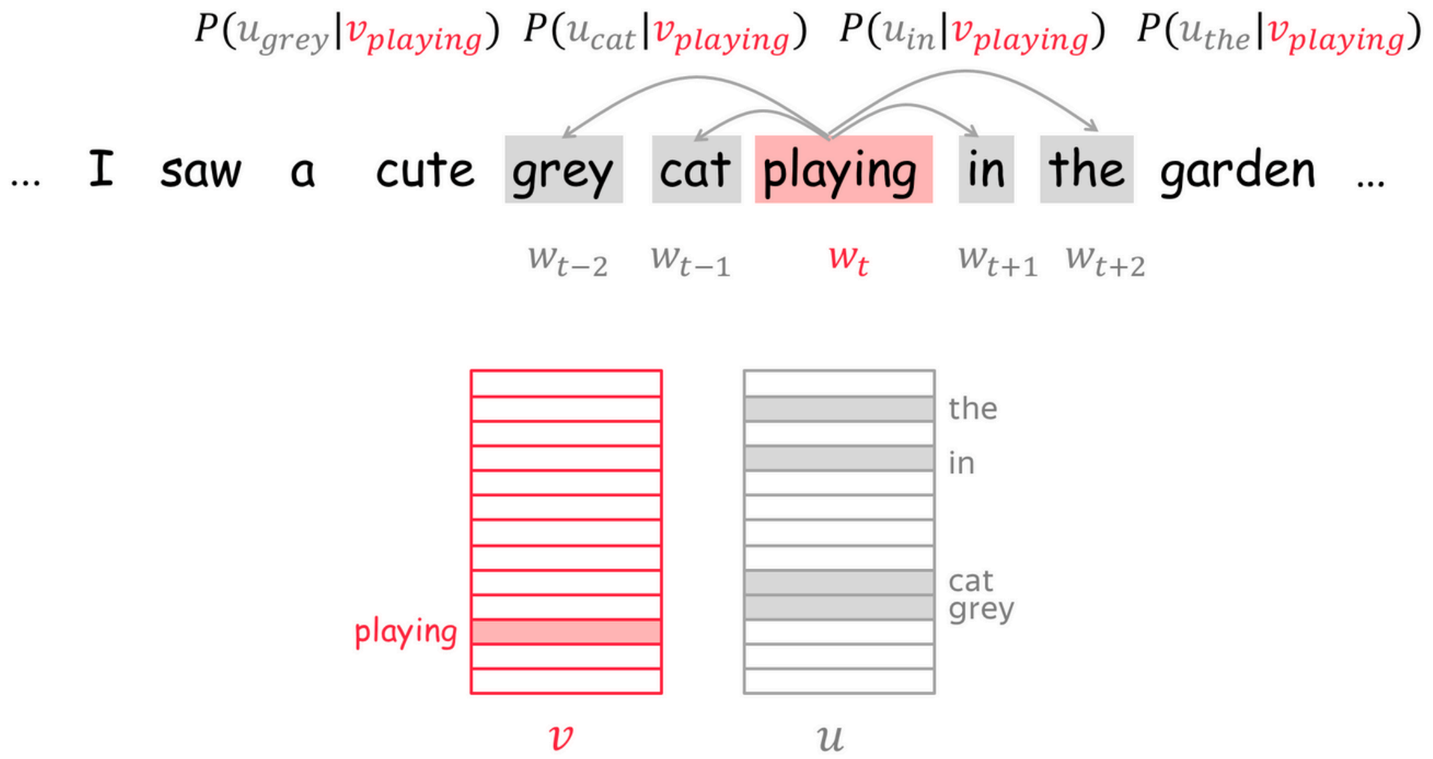
Solution 3: Word2Vec



Solution 3: Word2Vec

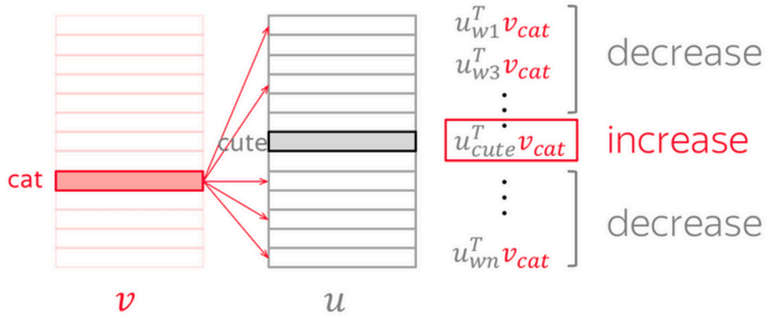


Solution 3: Word2Vec



Solution 3: Word2Vec | Faster Training: Negative Sampling

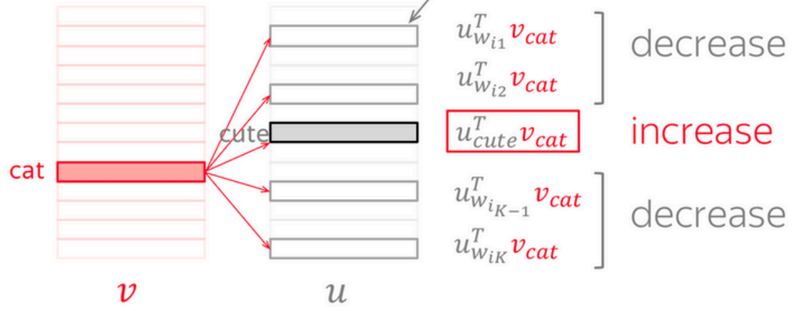
Dot product of v_{cat}:
- with u_{cute} - increase,
- with <u>all other</u> u - decrease



Parameters to be updated:
- v_{cat}
- u_w for all w in the vocabulary — |V| + 1 vectors

Dot product of v_{cat}:
- with u_{cute} - increase,
- with <u>a subset of other</u> u - decrease

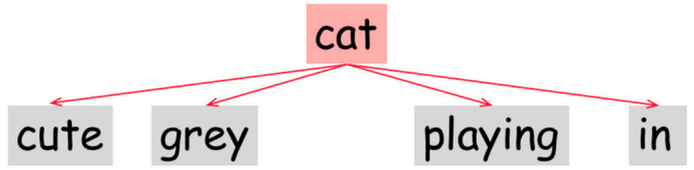
Negative samples: randomly selected K words

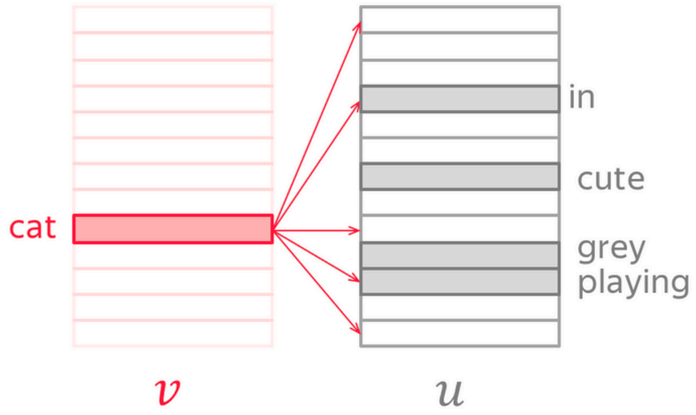


Parameters to be updated:
- v_{cat}
- u_{cute} and u_w for w in K negative examples — K + 2 vectors

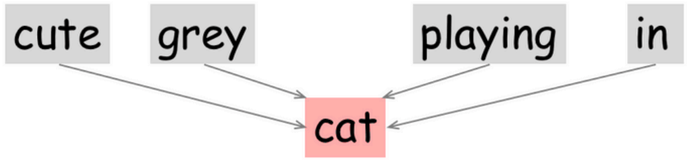
Solution 3: Word2Vec | Skip-Gram vs CBOW

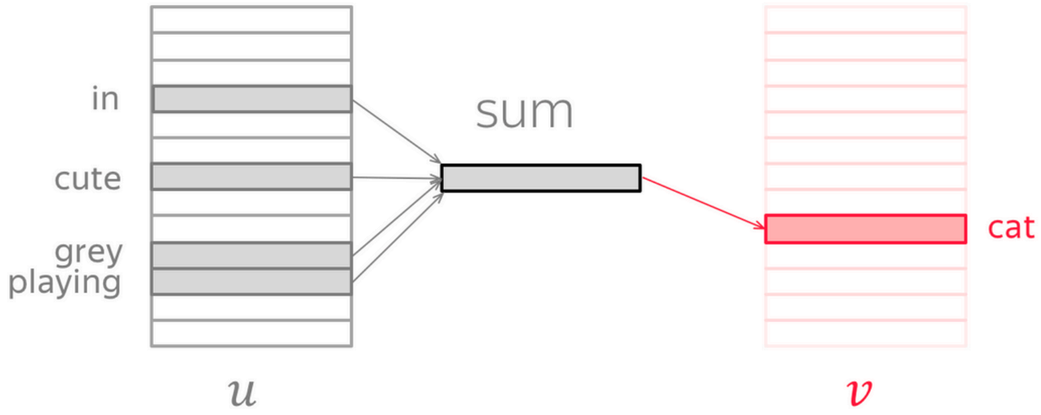
… I saw a cute grey cat playing in the garden …





Skip-Gram: from central predict context (one at a time)





CBOW: from sum of context predict central

Solution 4: [Hybrid] GloVe (Global Vectors for Word Representation)