بسم الله الرحمن الرحيم

دورة "استرجاع المعلومات" باللغة العربية ـ صيف ٢٠٢١

Information Retrieval – Summer 2021

7. Term Representation

Tamer Elsayed

Qatar University


21
POLL

21

Pseudo relevance feedback doesn't require any feedback from the user.

- Yes
- No

Sharing a post on Facebook is considered an implicit relevance feedback.

- Yes
- No

"Liking" a tweet on Twitter is considered implicit feedback.

- Yes
- No

Printing a retrieved page is considered explicit relevance feedback.

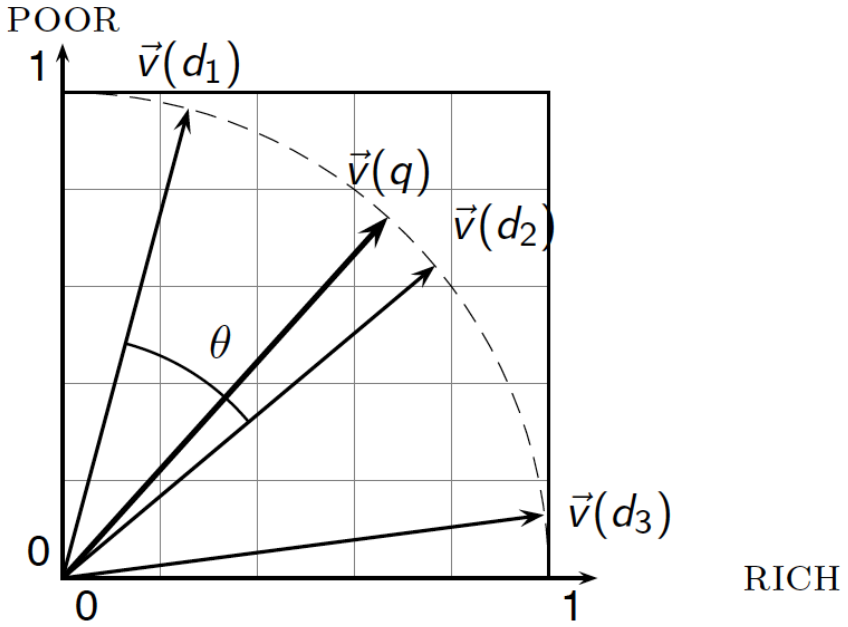
- Yes
- No

Today's Roadmap

- Local Representation

- Distributed Representation
  - Observed Representation
  - Latent Representation

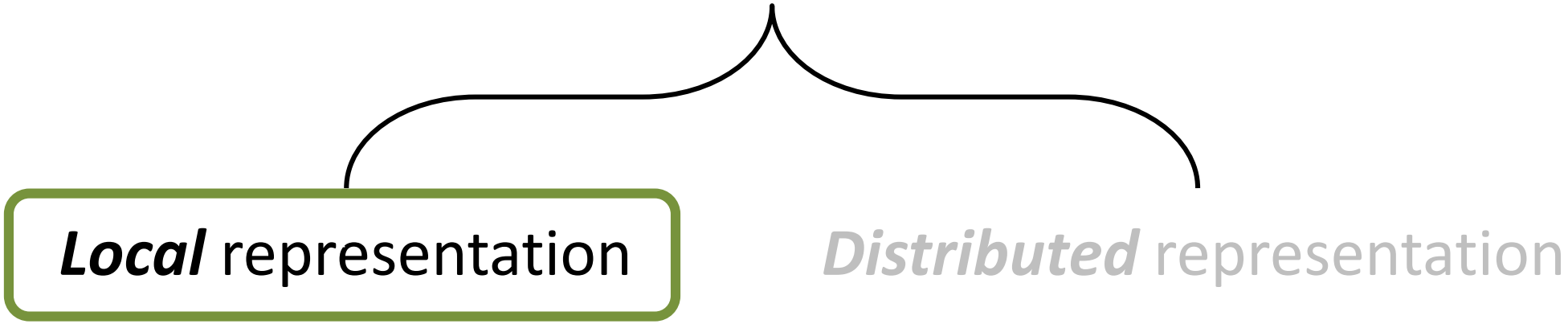
Recall: Vector Space Model …



- Terms are axes of the space
- Each document is a real-valued vector

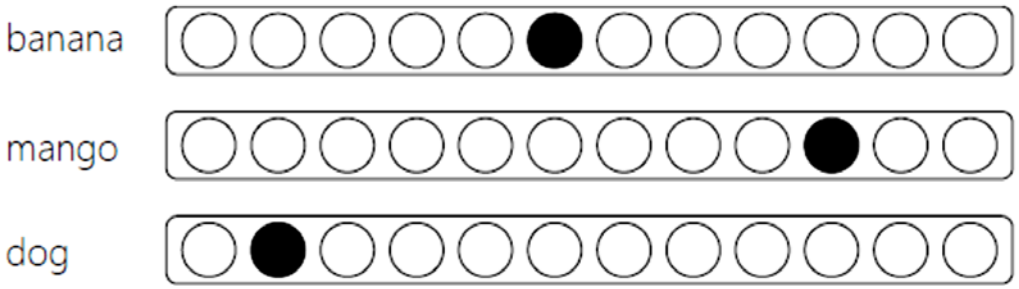
Representation of terms?

Types of Term Representations



Local Representation of Terms

- aka one-hot representation
- Every term in vocabulary T is represented by a binary vector of length |T|, where ***one*** position in the vector is set to one and the rest to zero.



No notion of similarity!

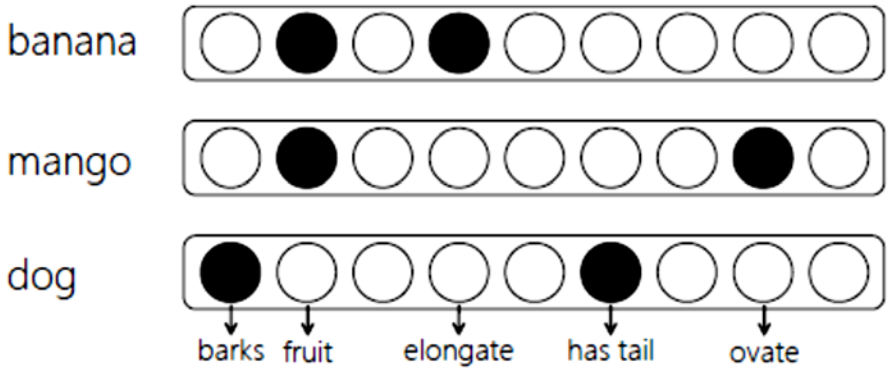
Types of Term Representations

Local representation

Distributed representation

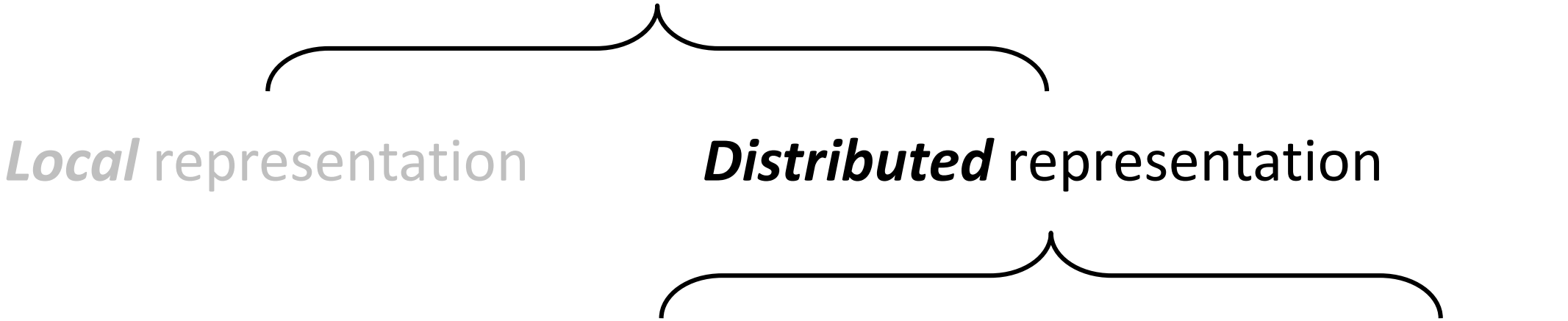
Distributed Representations

- Every term in vocabulary T is represented by a ***real-valued vector*** of length ***k***.
- The vector can be sparse or dense.
- The vector dimensions may be ***observed*** (e.g., hand-crafted features) or ***latent*** (e.g., embedding dimensions).



There is a notion of similarity!

Types of Term Representations



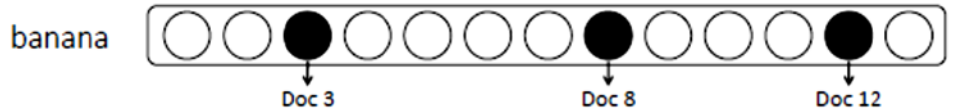
Observed Distributed Representation

- aka ***explicit*** distributed representation.
- The choice of features is a key consideration.

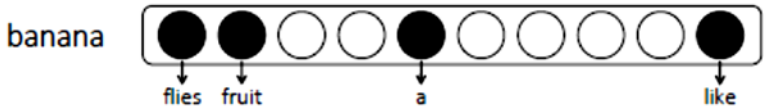
- One approach uses the **distributional hypothesis**: *terms that are used (or occur) in similar context tend to be semantically similar*.

> ***"a word is characterized by the company it keeps"***

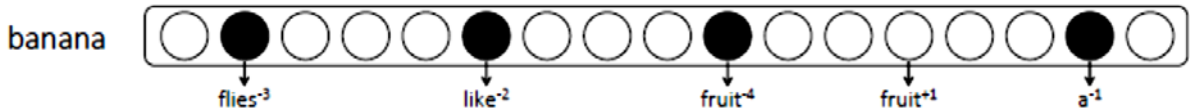
Examples of Distributed Representation



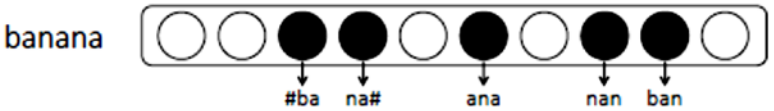
(a) In-document features



(b) Neighbouring-term features



(c) Neighbouring-term w/ distance features



(d) Character-trigraph features

Notion of Similarity

- Two terms are similar if their feature vectors are close.

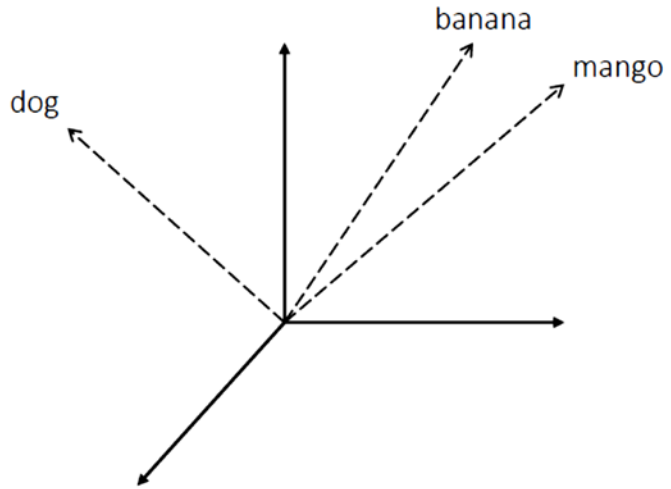
But different feature spaces may capture different notions of similarity.

- Is *Seattle* more similar to…

  Sydney (similar *type*)

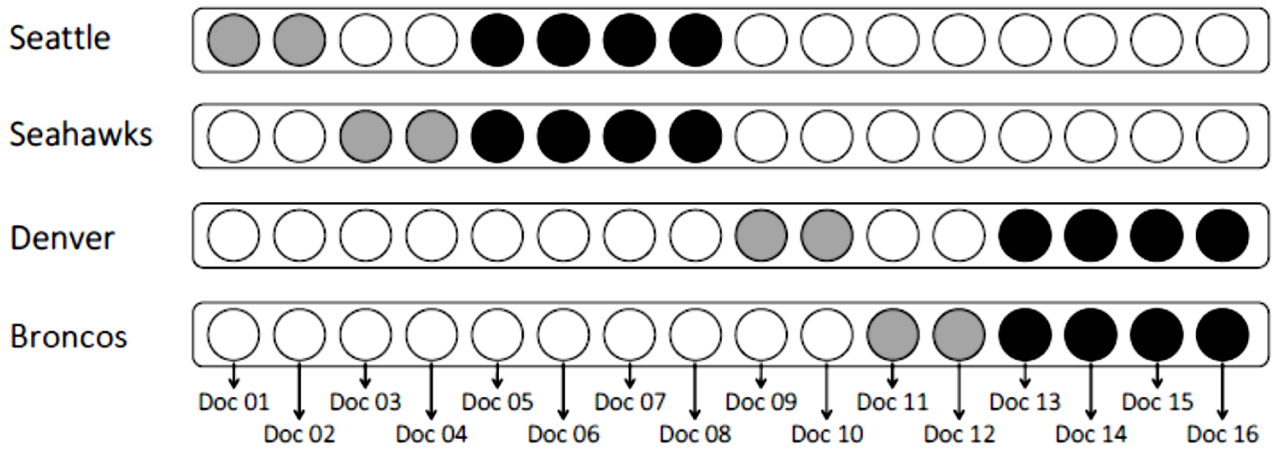
  or

  Seahawks (similar *topic*)



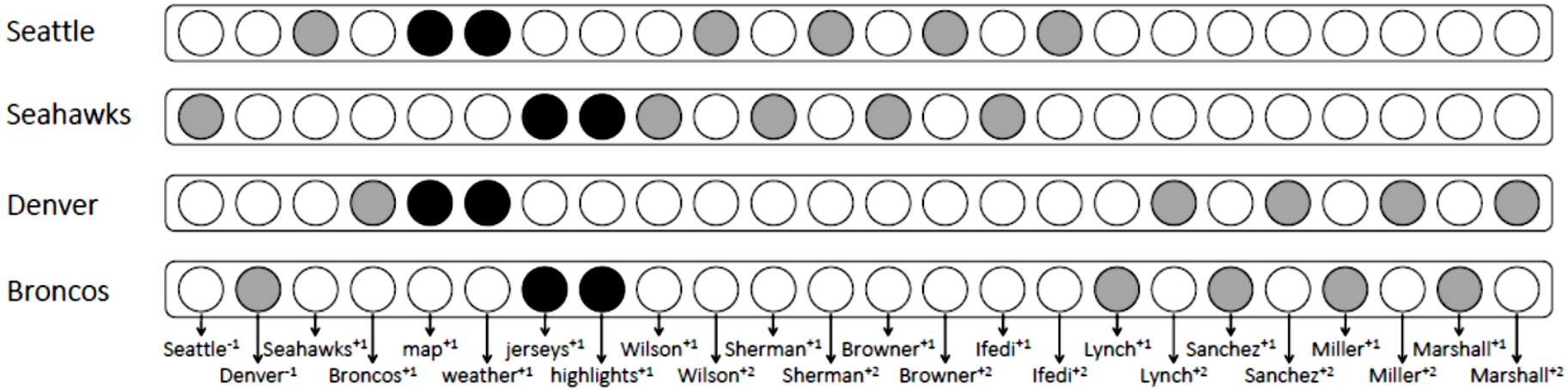
- Depends on your choice of features!

In-document features



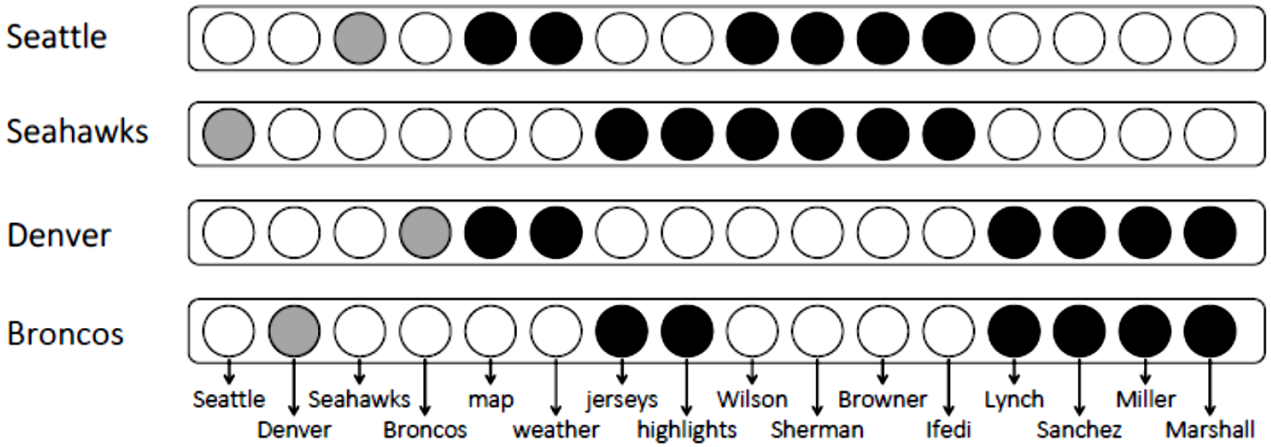
Topical similarity

“Neighbouring w/ distances” features



Typical similarity

“Neighbouring” features



Mix of Topical & Typical similarity

Key Message!

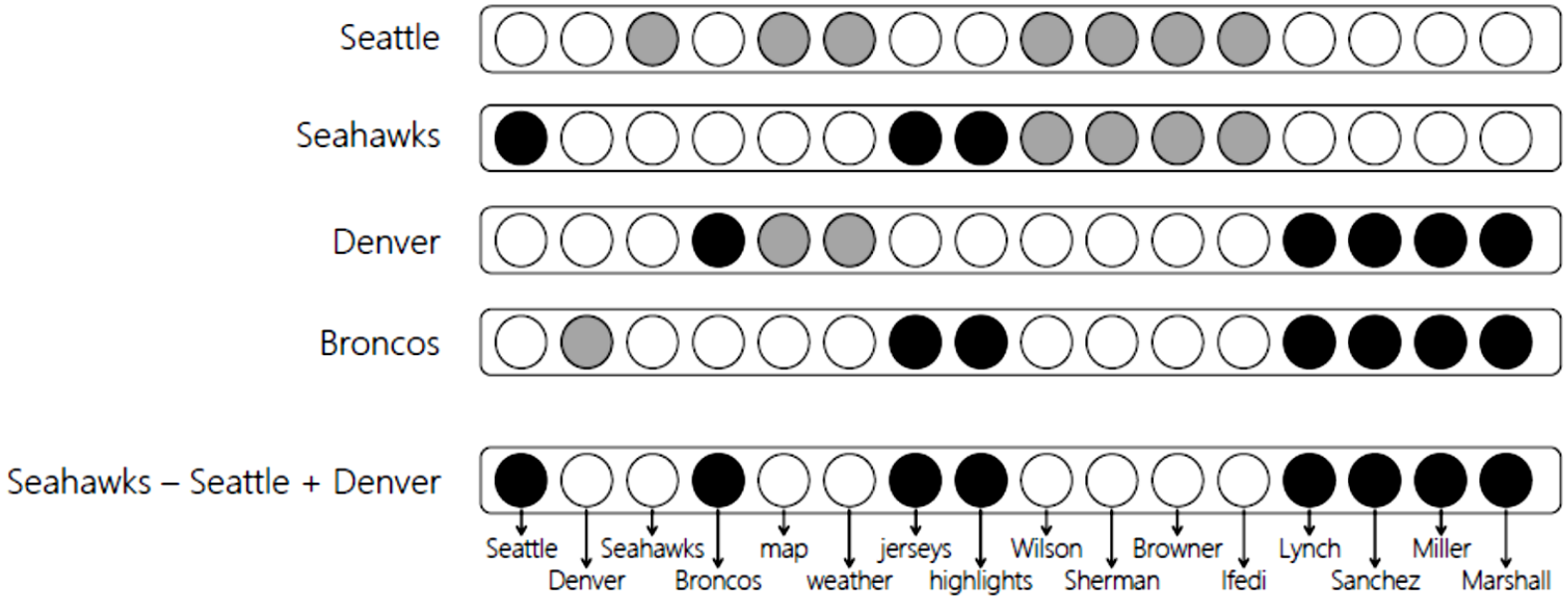
Different vector representations capture different notions of similarity between terms.

Underlying feature space needs to align with the notion of similarity that we are interested in.

Linguistic Regularities

- Some feature spaces capture interesting linguistic regularities.
- e.g., simple vector algebra in the term-neighboring term space may be useful for word analogy tasks.

$$\vec{v}_{Seahawks} - \vec{v}_{Seattle} + \vec{v}_{Denver} \approx \vec{v}_{Broncos}$$




22
POLL

22

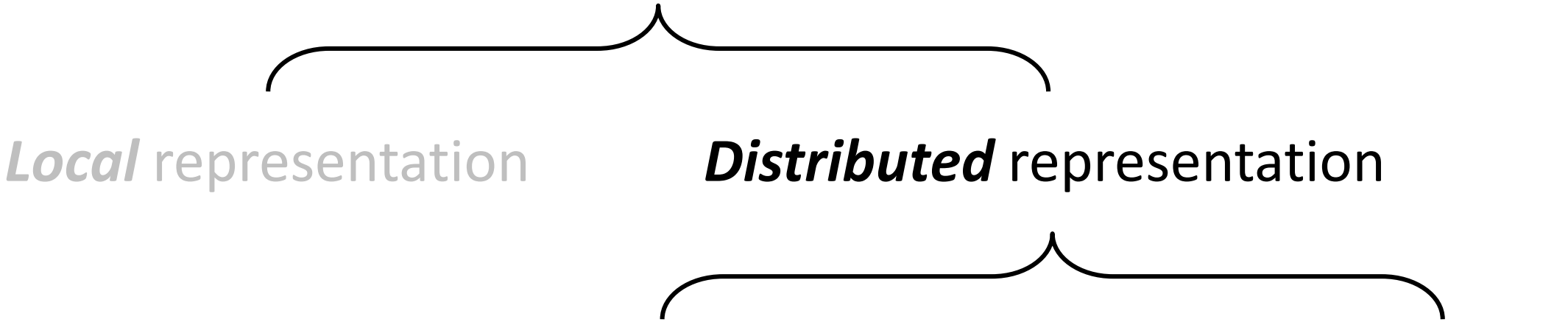
Representing terms as vectors of documents is a local representation.

- Yes
- No

The similarity between "Physics" and "Biology" words is "typical".

- Yes
- No

Types of Term Representations



Embeddings

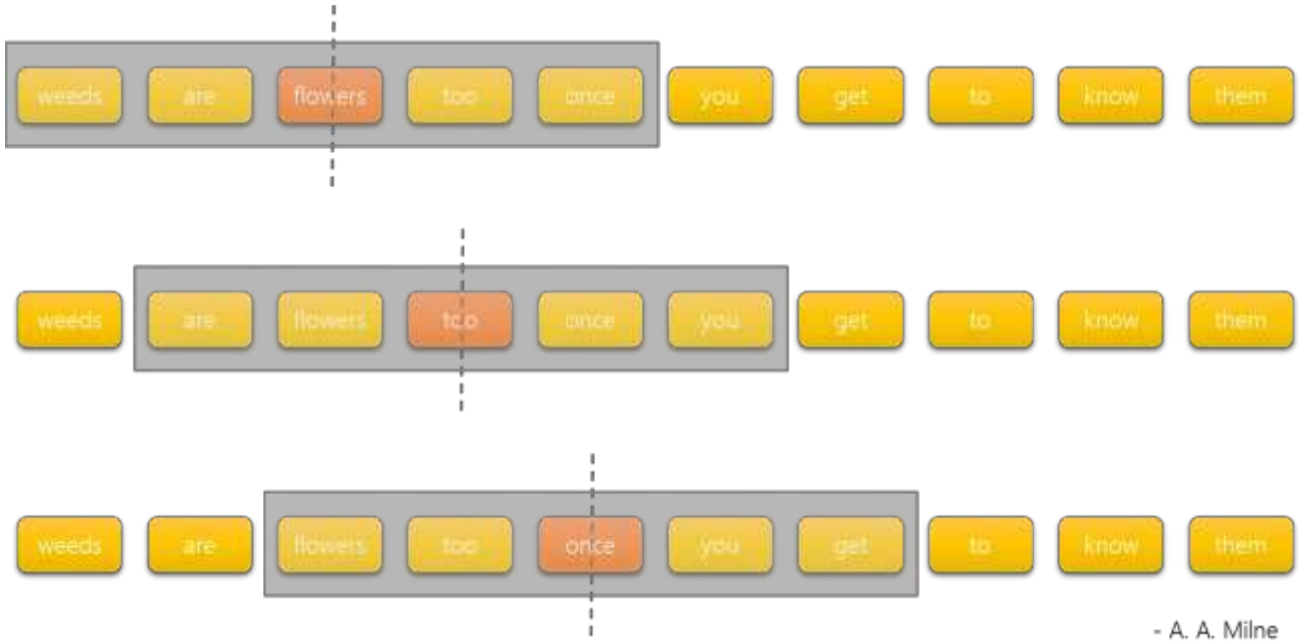
Representation of terms in a new space such that the properties of, and the relationships between, the terms are preserved from the original representation.

- Compared to observed feature spaces:
  - usually learned from observed features.
  - typically have fewer dimensions.
  - more dense.
  - dimensions may be less interpretable.
  - may generalize better.

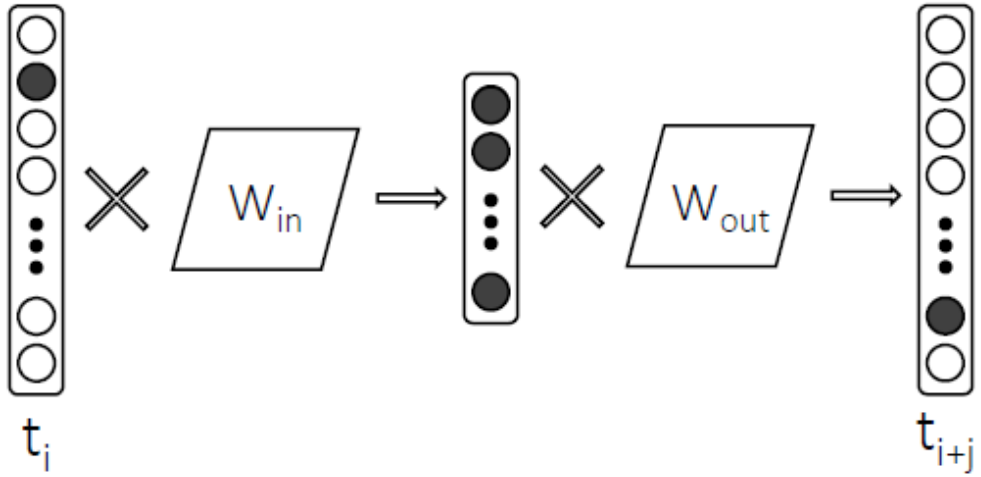
Think sparse, but Act dense!

Word2Vec

- Goal: shallow neural model learning from billion words corpus.
- Predict middle word from neighbors within fixed size context window.
- Two different architectures: Skip-Gram, CBOW



Skip-Gram



Predict a term t_i given one of its neighbors

Skip-Gram Loss

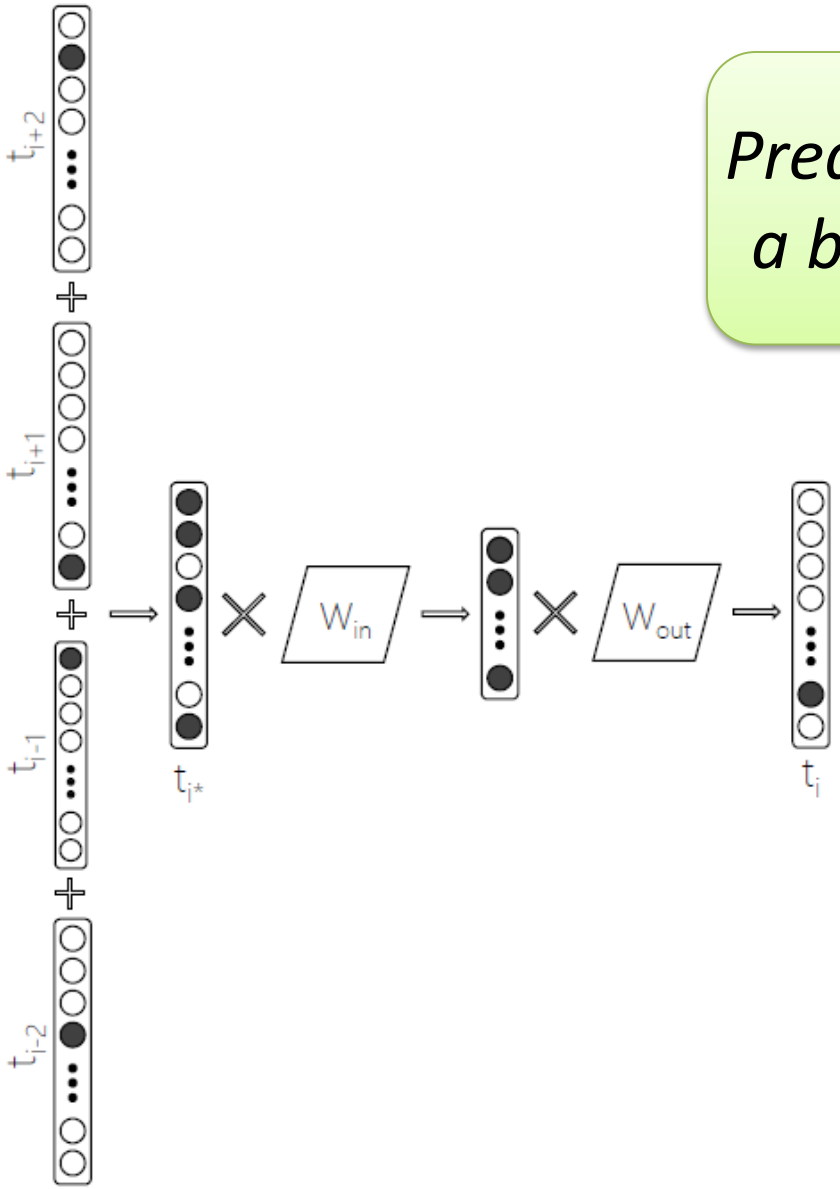
$$\mathcal{L}_{skip-gram} = -\frac{1}{|S|}\sum_{i=1}^{|S|}\sum_{-c\leq j\leq +c, j\neq 0} log(p(t_{i+j}|t_i))$$

$$\text{where,} \qquad p(t_{i+j}|t_i) = \frac{\exp\left((W_{out}^{\intercal}\vec{v}_{t_{i+j}})^{\intercal}(W_{in}\vec{v}_{t_i})\right)}{\sum_{k=1}^{|T|}\exp\left((W_{out}^{\intercal}\vec{v}_{t_k})^{\intercal}(W_{in}\vec{v}_{t_i})\right)}$$

S: set of all windows over the training text

c: number of neighbours to predict on either side of t_i

CBOW



Predict the middle term t_i given a bag of its neighboring terms

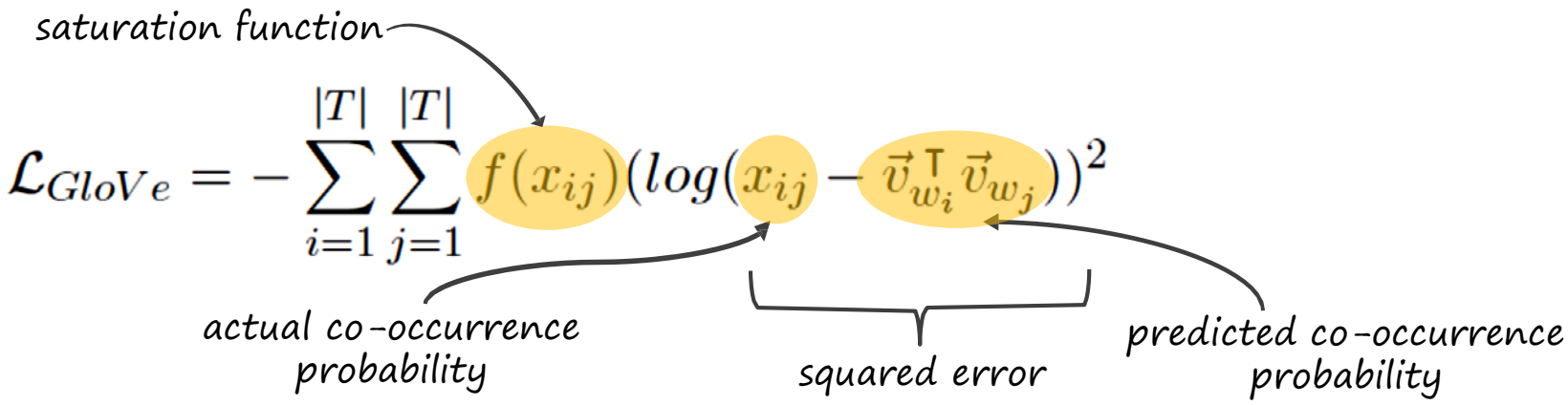
Trains faster?

CBOW Loss

$$\mathcal{L}_{CBOW} = -\frac{1}{|S|}\sum_{i=1}^{|S|} log(p(t_i| \sum_{-c \leq j \leq +c, j \neq 0} t_{i+j}))$$

GLOVE

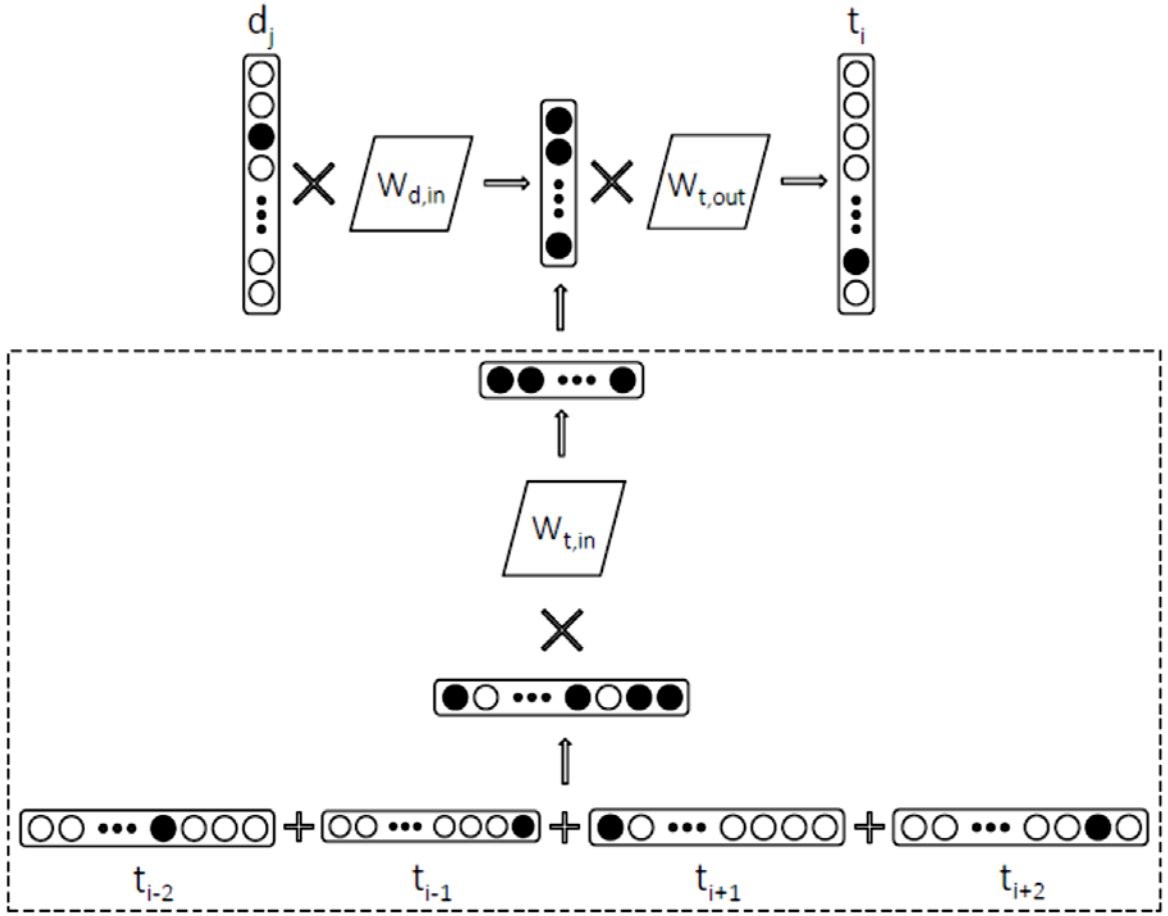
- Replace the cross-entropy error with a squared-error and apply a saturation function $f(\dots)$ over x_{ij}.



- Generates two different (IN and OUT) embeddings
- Uses the sum of the IN and the OUT vectors as the embedding for each term in the vocabulary.

Pargraph2Vec

- W2V style model where context is document, not neighboring term




23
POLL

23

Word embeddings are trained in a completely unsupervised fashion.

- Yes
- No

The similarity of two documents that do not have any terms in common can be non-zero if we used word embeddings to represent their terms.

- Yes
- No

All different senses of the same word will share the same embeddings using SkipGram model.

- Yes
- No

What we got so far …

- Each term has one learned embeddings vector, called “word type” embeddings.

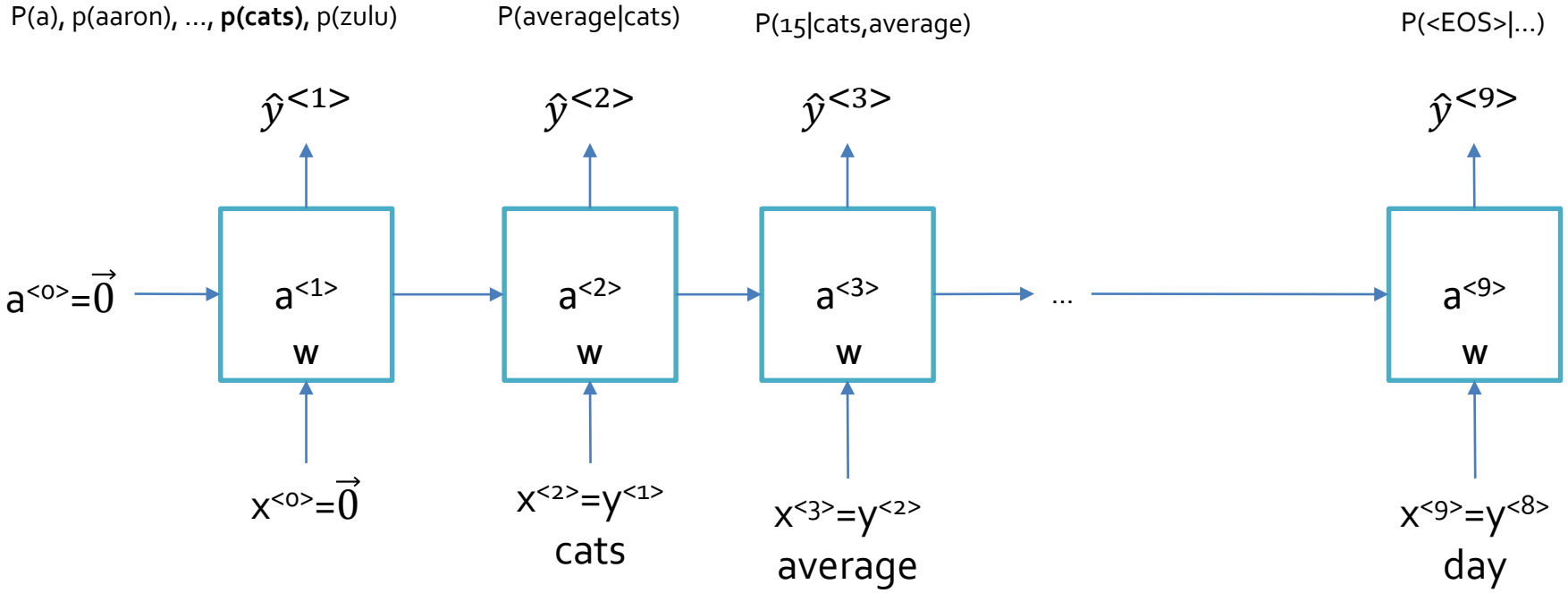
Problem?

- I passed over to the river *bank*
- I withdrew money from my *bank* account
- There is a shortage in the blood *bank*

Words have different meanings in different contexts!

- We need “Contextual Word Vectors”, called “word token” embeddings.

Learning a Language Model using RNN

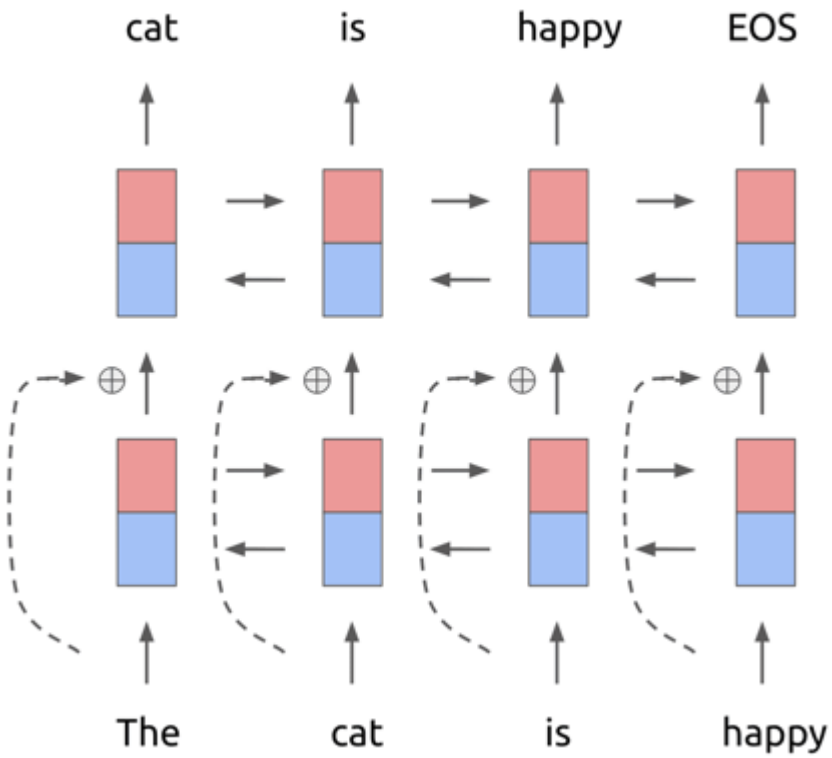


- Cats average 15 hours of sleep a day. <EOS>
  - P(sentence) = P(cats)P(average|cats)P(15|cats,average)...

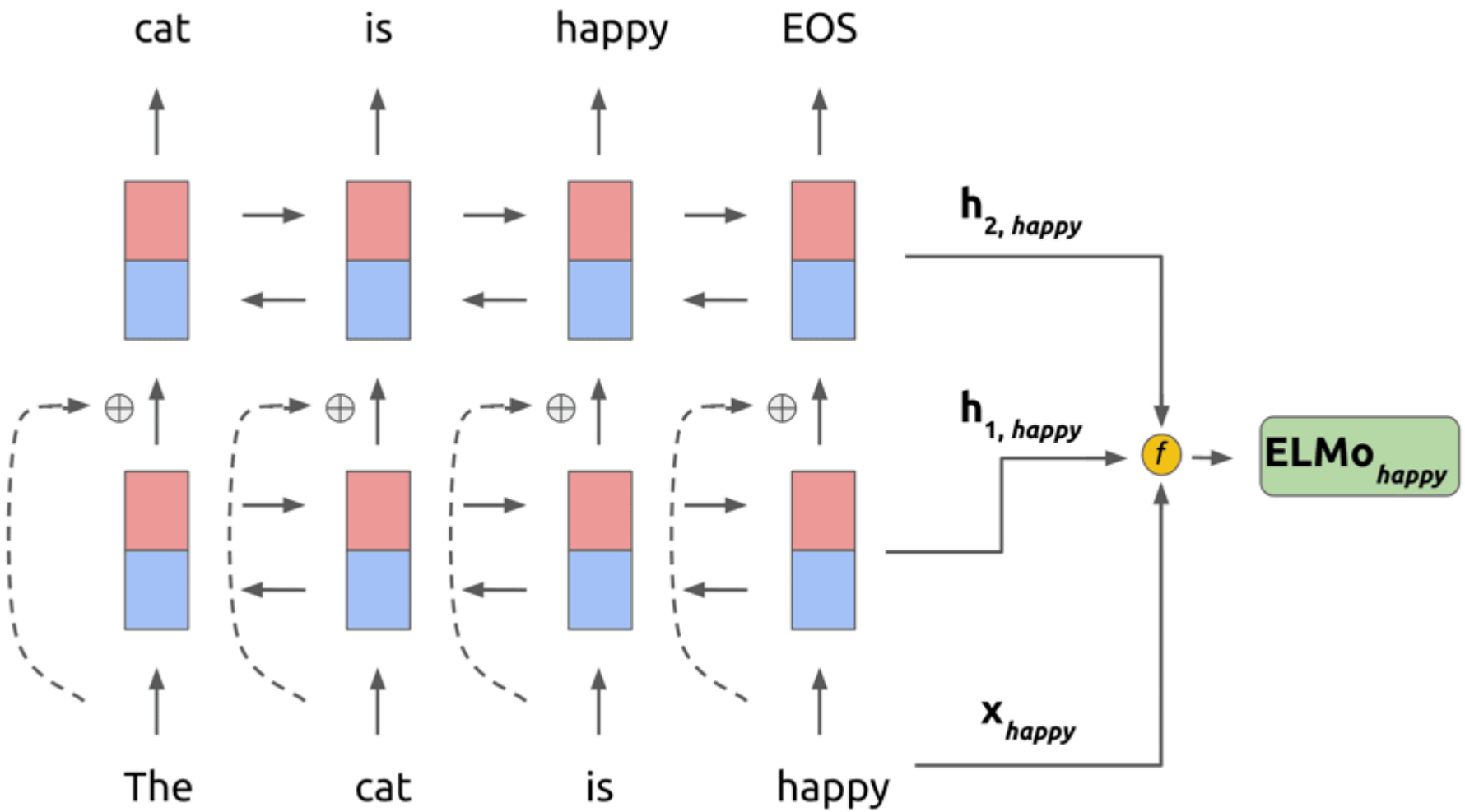
ELMo: Embeddings from Language Models

- Trains a language model using a 2-layer bi-directional LSTM (biLMs)

- Input: fixed-length word embedding.
  - One-hot encoding
  - Word2Vec
  - Glove
  - ….



ELMo: Embeddings from Language Models




24
POLL

24

ELMo provides different embeddings for the same word appearing in two different sentences.

- Yes
- No

To produce ELMo embeddings, we need ...

- the static embeddings of terms
- the learned ELMo model
- both

The new wave: Transformer & BERT