Deep learning for NLP

Eneko Agirre, Gorka Azkune
Ander Barrena, Oier Lopez de Lacalle
@eagirre @gazkune @4nderB @oierldl #dl4nlp
http://ixa2.si.ehu.eus/eneko/dl4nlp

Session 3: Representation learning, word embeddings and recurrent neural networks

Plan for the course

- Introduction: machine learning and NLP
- Multilayer perceptron
- **Word representation and Recurrent neural networks (RNN)**
- Sequence-to-Sequence (seq2seq) and Machine Translation
- Attention, transformers and Natural language inference
- Pre-trained transformers, BERT, GPT
- Bridging the gap between natural languages and the visual world

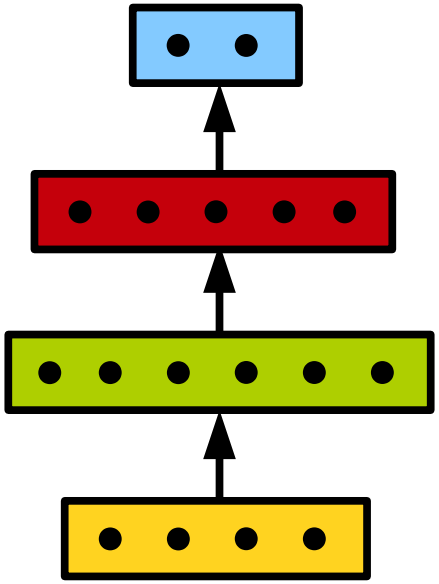
Representation learning

- Document as **bag-of-words**
  - Each word ~ a **one-hot-vector**
    (0 0 0 0 0 … 1 … 0 0 0)
    dog
  - Each document ~ **summation of words in the document**
    (0 0 0 1 0 … 1 … 1 0 0)
    sweet dog

Source: Sam Bowman

Previously...

Multilayer perceptron (MLP) $P(c|x)$



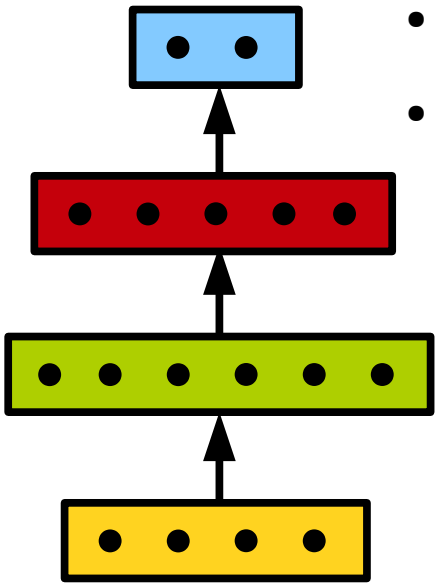
$$y = softmax(W_2 h_1 + b_2) \quad y$$

$$h_1 = f(W_1 h_0 + b_1)$$

$$h_0 = f(W_0 x + b_0)$$

$$J = -\log(\exp(y_{correct}) / \sum_c^C \exp(y_c)) = -y_{correct} + \log \sum_c^C \exp(y_c)$$

Deep: Multilayer perceptron



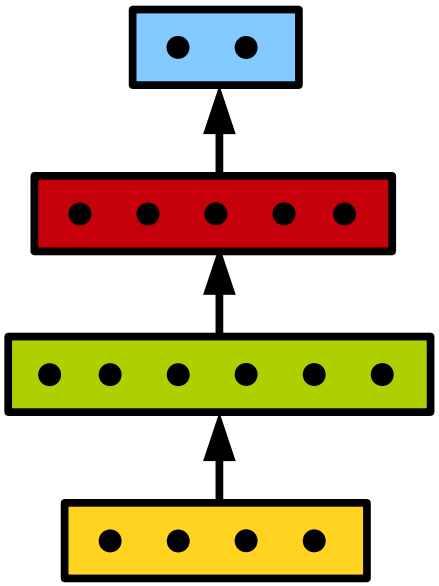
Training: SGD

- Start with random parameters: W (includes bias)
- Each epoch
  - Shuffle training data
  - For each mini-batch (set of K examples)
    - **Compute the loss function (forward)**
    - **Compute the gradient of the loss function (backward)**
    - Update parameters: (learning rate η)
  - Measure train and dev. accuracy

$$W = W - \eta \frac{1}{K} \sum_{i}^{K-1} \nabla J_i(W)$$

- Continue until loss function converges / time is up / **dev. accuracy stops increasing**

Previously...



Topology: number and size of layers

Non-linearity

Optimizer

Learning-rate

Size of mini-batch

Weight of L2 regularization

Dropout rate

Plan for this session

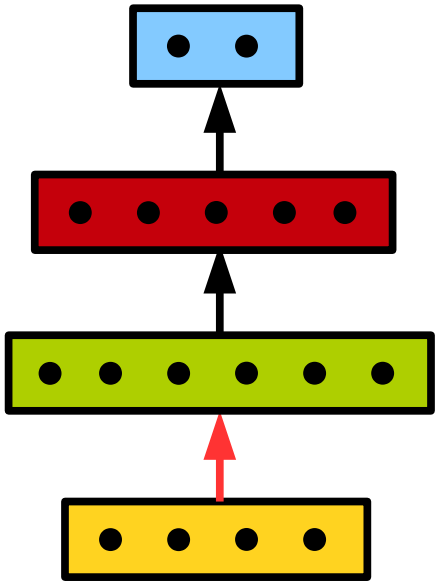
- Representation learning
  - Word embeddings
  - *Amazing abilities with word embeddings (Artetxe et al. 2018)*
- From words to sequences: Recurrent Neural Networks (RNN)

Representation learning

- Document as **bag-of-words**
  - Each word ~ a **one-hot-vector**
    (0 0 0 0 0 … 1 … 0 0 0)
  - Each document ~ **summation of words in the document**
    (0 0 0 1 0 … 1 … 1 0 0)
- NLP has been representing **words as distinct features** for many years!
  - Is there a better alternative?

Source: Sam Bowman

Representation learning



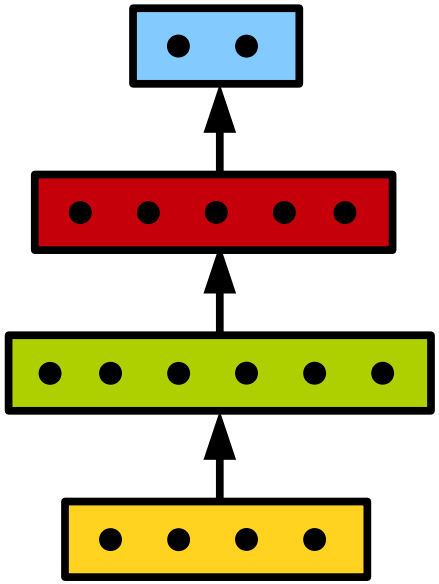
MLP: What are we learning in W_0 when we backpropagate?

$$y = softmax(W_2 h_1 + b_2)$$

$$h_1 = f(W_1 h_0 + b_1)$$

$$h_0 = f(W_0 x + b_0)$$

Representation learning



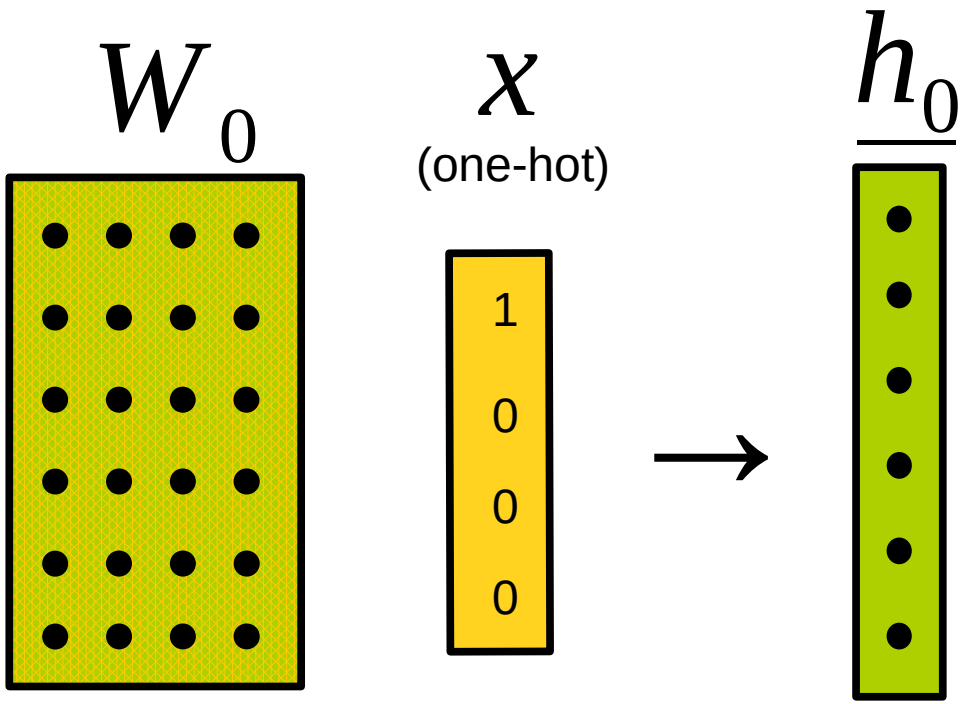
MLP: What are we learning in W_0 when we backpropagate?

$$y = softmax(W_2 h_1 + b_2)$$

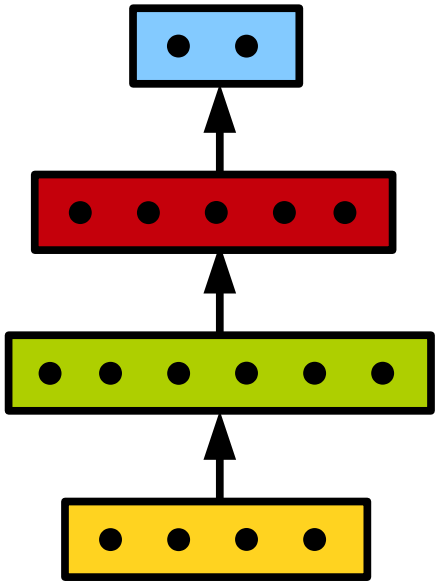
$$h_1 = f(W_1 h_0 + b_1)$$

$$h_0 = f(W_0 x + b_0)$$

$\underline{h_0}$



Representation learning

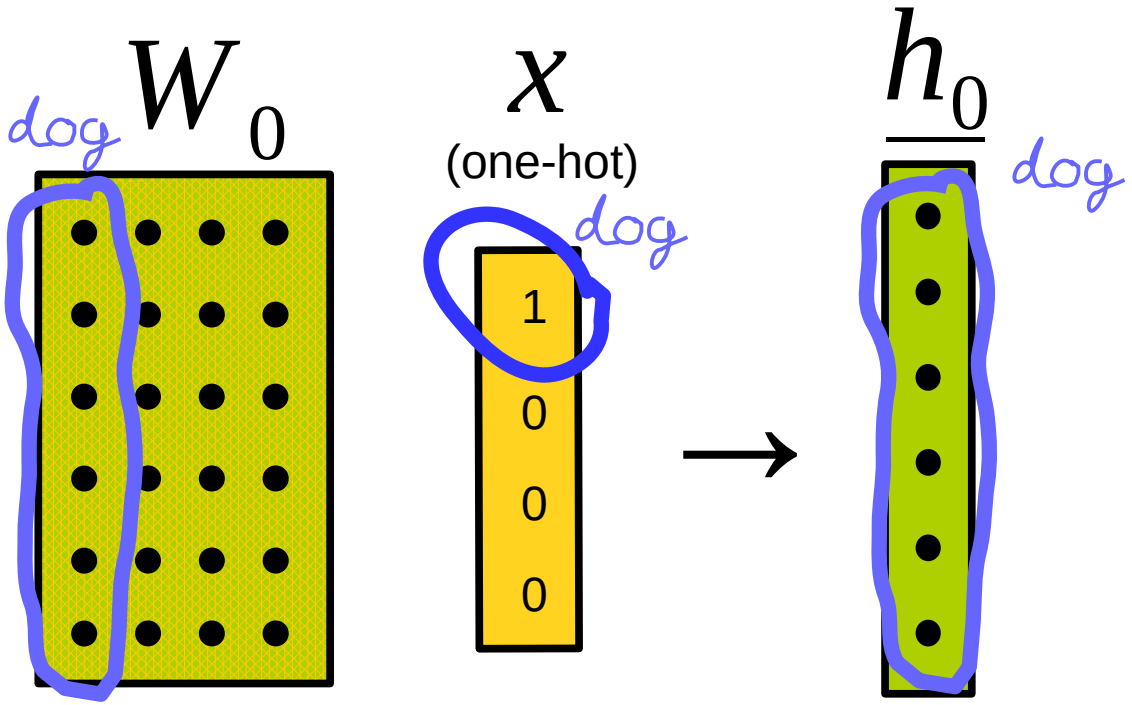


Back-propagation allows Neural Networks to **learn representations for words** while training: **word embeddings!**

$$y = softmax(W_2 h_1 + b_2)$$

$$h_1 = f(W_1 h_0 + b_1)$$

$$h_0 = f(W_0 x + b_0)$$



Representation learning

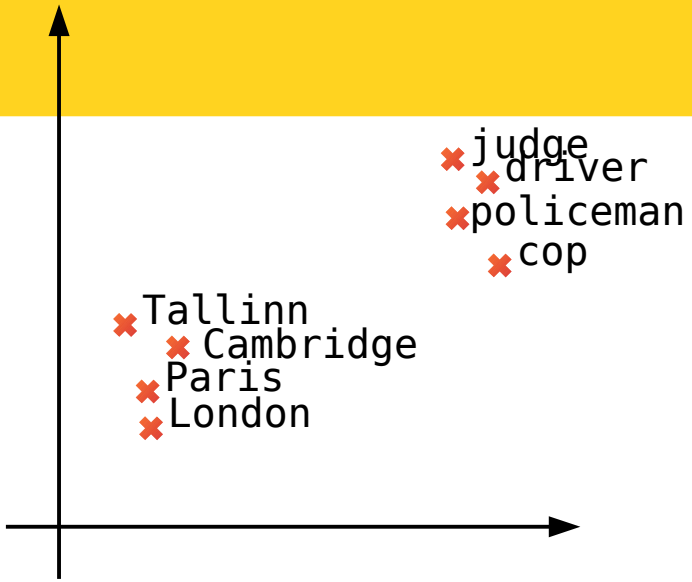
- Word embeddings
  aka continuous vector space,
  distributed representations
  - Are they useful for anything?
  - Can we transfer this representation
    from one task to the other?
  - Which task would be the best to learn embeddings
    that can be used in other tasks?
  - Can we have all languages in one embedding space?

Plan for this session

- Representation learning
  - **Word embeddings**
  - *Amazing abilities with word embeddings (Artetxe et al. 2018)*
- From words to sequences: Recurrent Neural Networks (RNN)

Word embeddings

- Let's represent words as vectors: Similar words should have vectors which are close to each other



WHY?

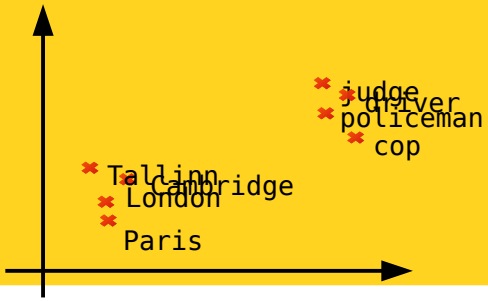
- If an AI has seen these two sequences

```
I live in Cambridge
I live in Paris
```

- … then which one should be more plausible?

```
I live in Tallinn
I live in policeman
```

Word embeddings



- Distributional hypothesis (Harris 54, Firth 57)

  "You shall know a word by the company it keeps"

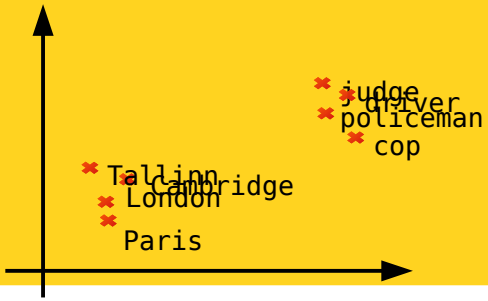
- Keep track of word co-occurrences with a huge co-occurrence matrix
  - Each row is the vector representation for that word

	furry	*dangerous*	*mammal*
bear	0.9	0.85	1
cat	0.85	0.15	1
frog	0	0.05	0

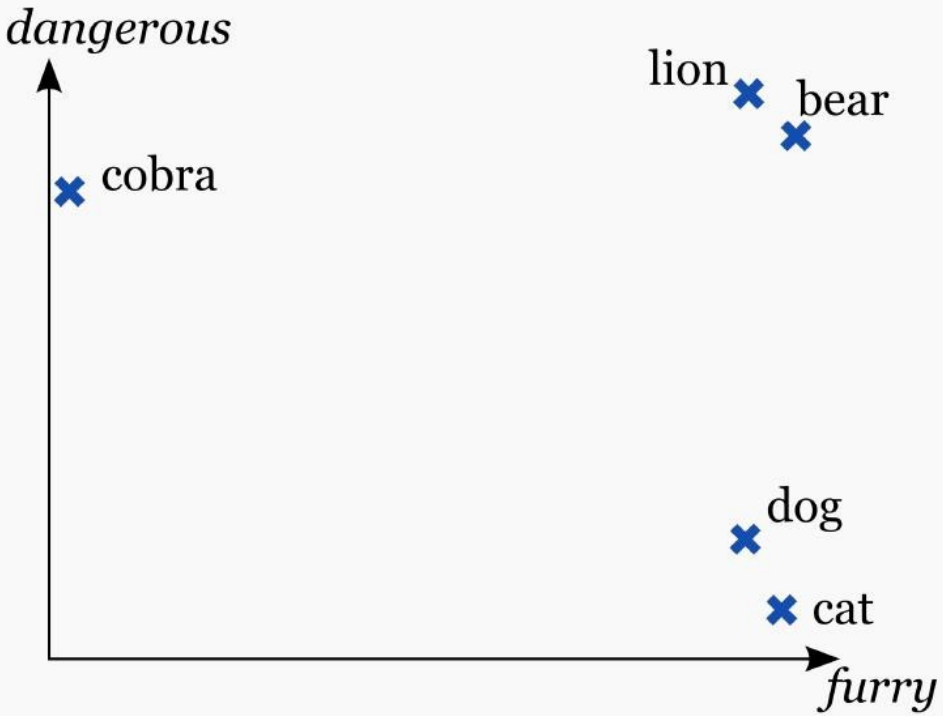
```
bear = [0.9, 0.85, 1.0]    cat = [0.85, 0.15, 1.0]
```

source:http://www.marekrei.com/teaching/cewe/

Word embeddings



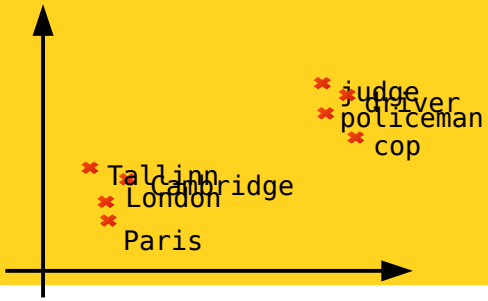
- E.g. 2-dimensional space



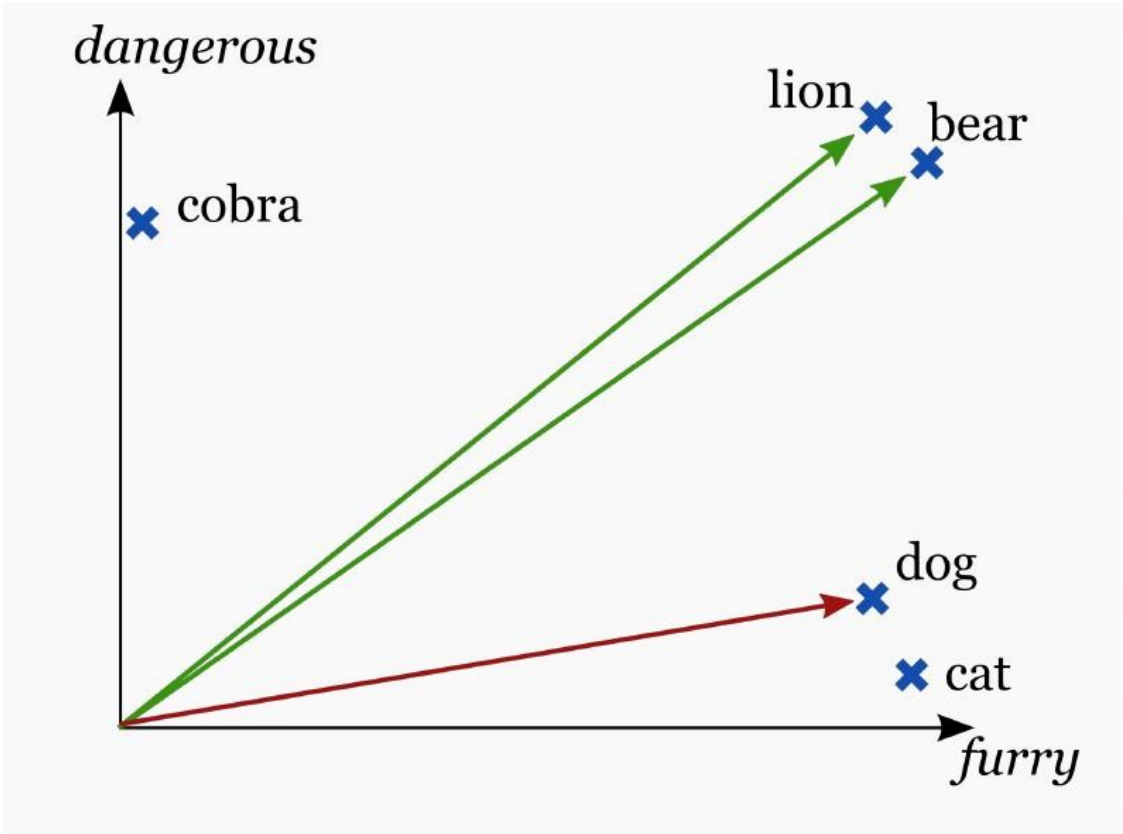
	furry	*dangerous*
bear	0.9	0.85
cat	0.85	0.15
cobra	0.0	0.8
lion	0.85	0.9
dog	0.8	0.15

source:http://www.marekrei.com/teaching/cewe/

Word embeddings



- Word similarity using cosine



$$\cos(\vec{a},\vec{b}) = \frac{\vec{a}\,\vec{b}^{T}}{\|\vec{a}\|\|\vec{b}\|}$$

$$\cos(\vec{a},\vec{b}) = \frac{\sum_i a_i b_i}{\sqrt{\sum_i a_i^2}\sqrt{\sum_i b_i^2}}$$

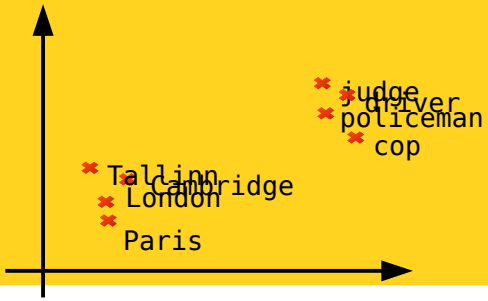
$$\cos(lion, bear) = 0.998$$

$$\cos(lion, dog) = 0.809$$

$$\cos(cobra, dog) = 0.727$$

source:http://www.marekrei.com/teaching/cewe/

Word embeddings

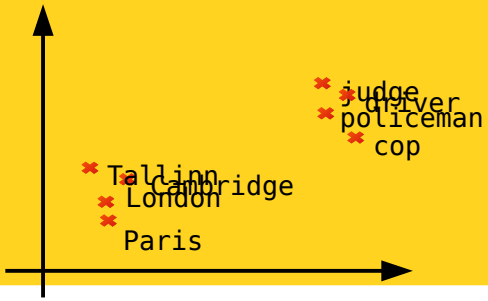


Creating distributional vector spaces:

- Define vocabulary size V
- Define window size W
- Count pairs of words in vocabulary that co-occur at distance equal or less than window size
- Numbers in the matrix:
  - Raw counts
  - Binary
  - TF.IDF $TF.IDF(w,c) = count(w,c) \log \frac{D}{docs(c)}$
  - Positive Pointwise Mutual Information

$$PPMI(w,c) = max(0, \log \frac{p(x,c)}{p(x)p(c)}), p(x) = \frac{count(x)}{C}$$

Word embeddings

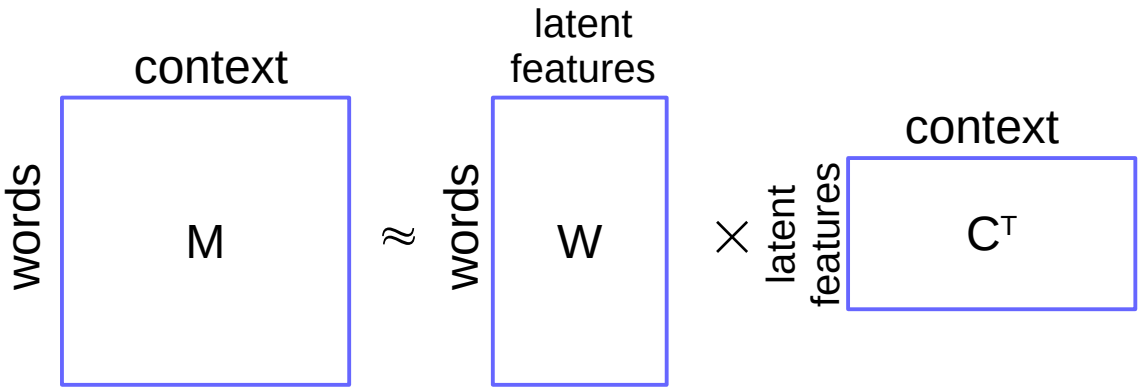


Distributional vector spaces:

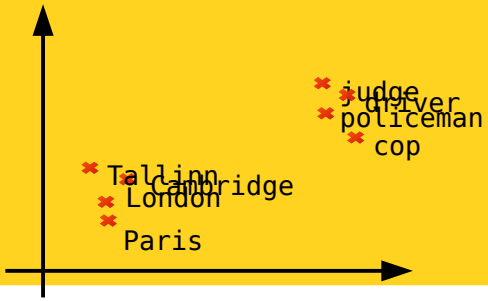
- High-dimensional (V), unmanageably large (Web1T5 1Mx1M)
- Sparse, many zeroes (Web1T5 99.95%)

It is possible to decompose M using SVD and low-rank approximation:

- Apply SVD $M = U \Sigma V^T$
- Select D singular values (latent dimensions) $M \approx M_D = U_D \Sigma_D V_D^T$
- Decompose M in two: $W = U_D \sqrt{\Sigma_D}, C = V_D \sqrt{\Sigma_D}$ $M \approx W C^T$



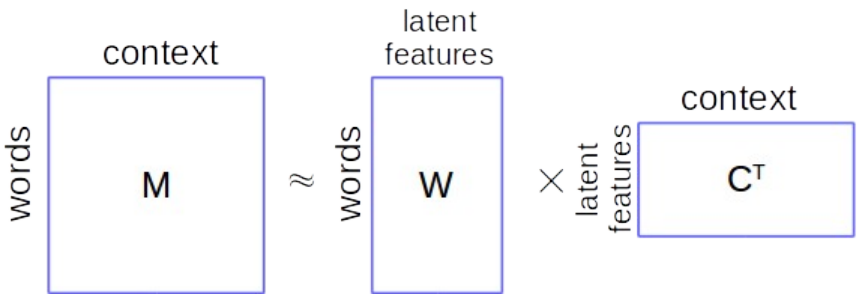
Word embeddings



Distributional vector spaces:

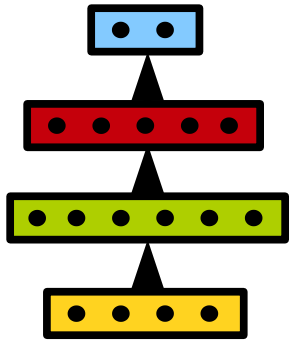
Option 1: use co-occurrence matrix PPMI

- 1A: large sparse matrix
- 1B: factorize it and use low-rank dense matrix (SVD)

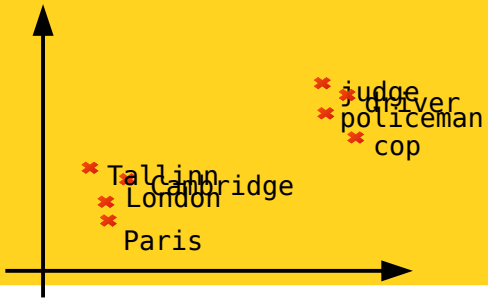


Option 2: learn low-rank dense matrix directly

- 2A: MLP on particular classification task
- 2B: **find a general task**



Word embeddings



General task with large quantities of data: **guess the missing word** (language models)

CBOW: given context guess middle word

> *… people who keep pet dogs or **cats** exhibit better mental and physical health …*

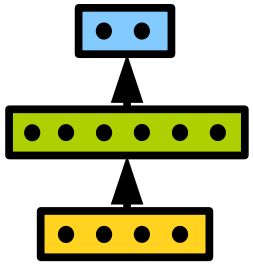
SKIP-GRAM given middle word guess context

> *… people who keep pet dogs or **cats** exhibit better mental and physical health …*

Proposed by Mikolov et al. (2013)

CBOW

Like MLP, one layer, LARGE **vocabulary**



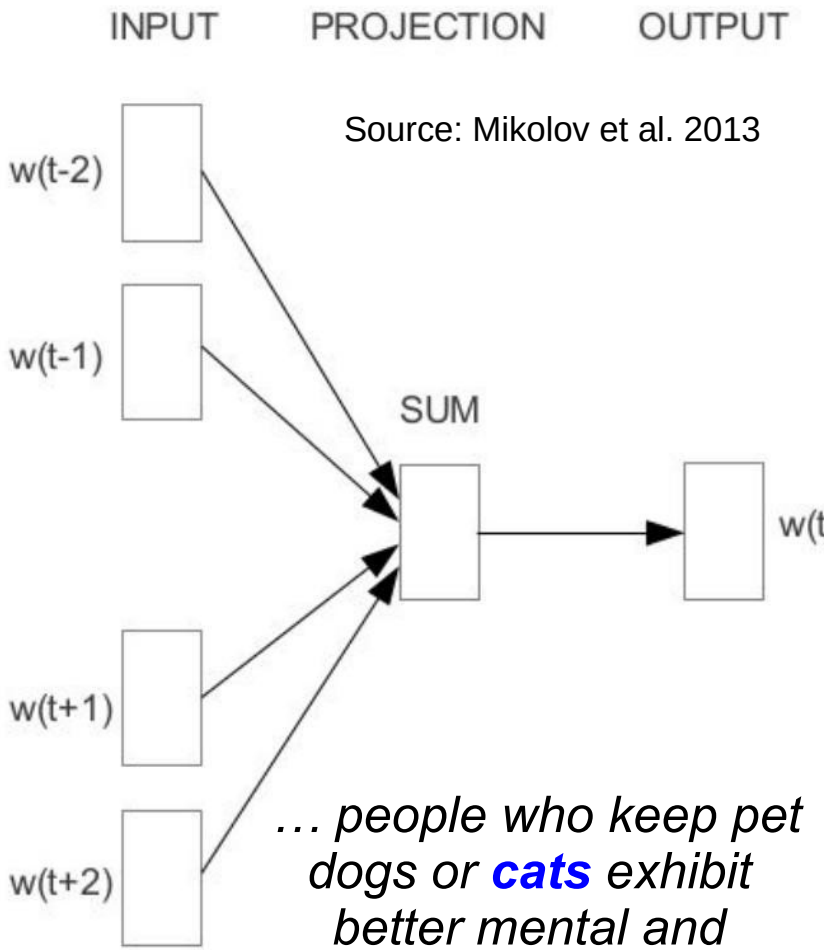
$$h_0 = W_0 x$$

$$y = softmax(W_1 h_0 + b_1)$$

$$y_i = \exp(\underline{h_{0i}}) / \sum_j^V \exp(\underline{h_{oi}})$$

$$J_{w(t)} = \underline{h_{0i}} - \log \sum_j^V \exp(\underline{h_{oj}})$$

Cross-entropy loss, but softmax expensive!

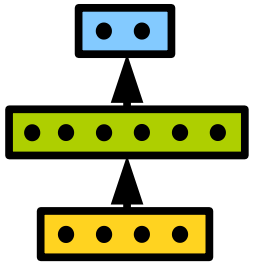


Source: Mikolov et al. 2013

*… people who keep pet dogs or **cats** exhibit better mental and physical health …*

CBOW

Like MLP, one layer, LARGE **vocabulary**

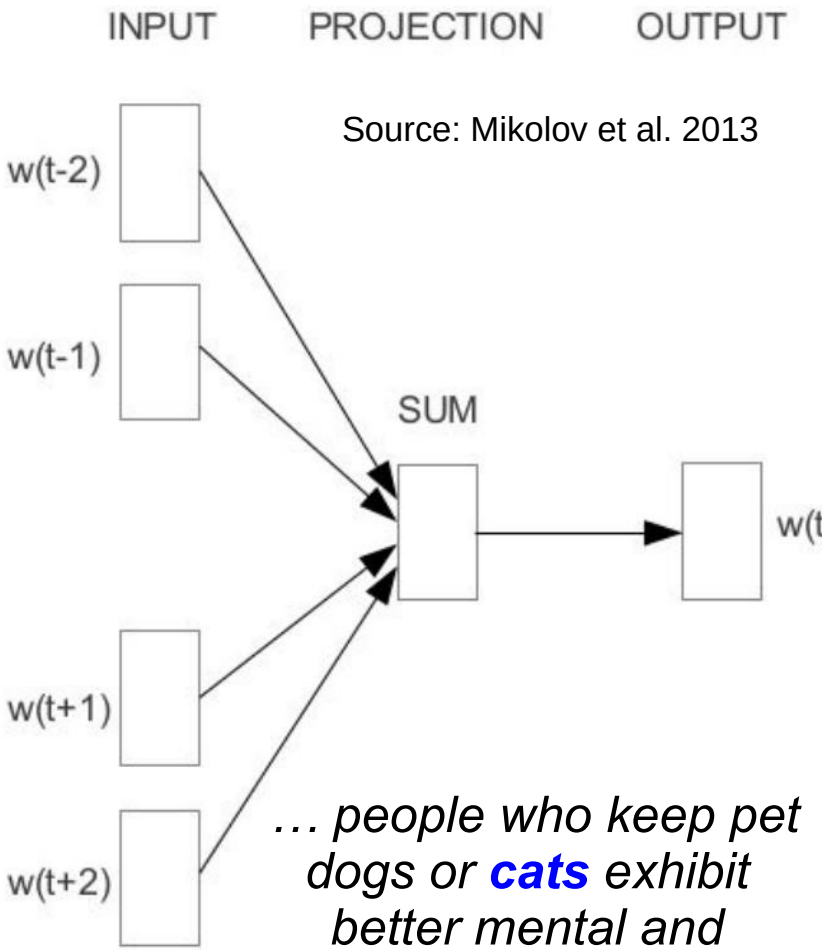


$$h_0 = W_0 x$$

$$y = softmax(W_1 h_0 + b_1)$$

$$y_i = \exp(\underline{h_{0i}}) / \sum_j^V \exp(\underline{h_{oi}})$$

$$J_{w(t)} = \underline{h_{0i}} - \log \sum_j^V \exp(\underline{h_{oj}})$$



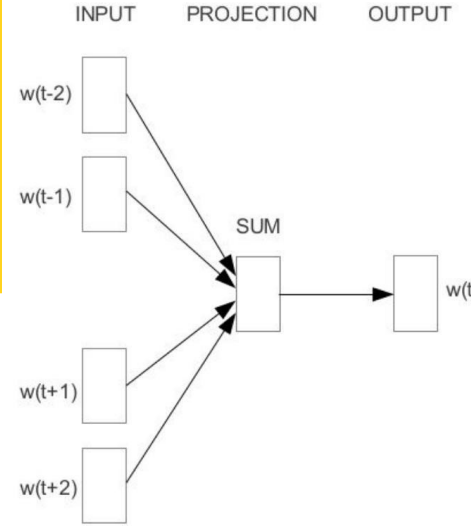
Source: Mikolov et al. 2013

*… people who keep pet dogs or **cats** exhibit better mental and physical health …*

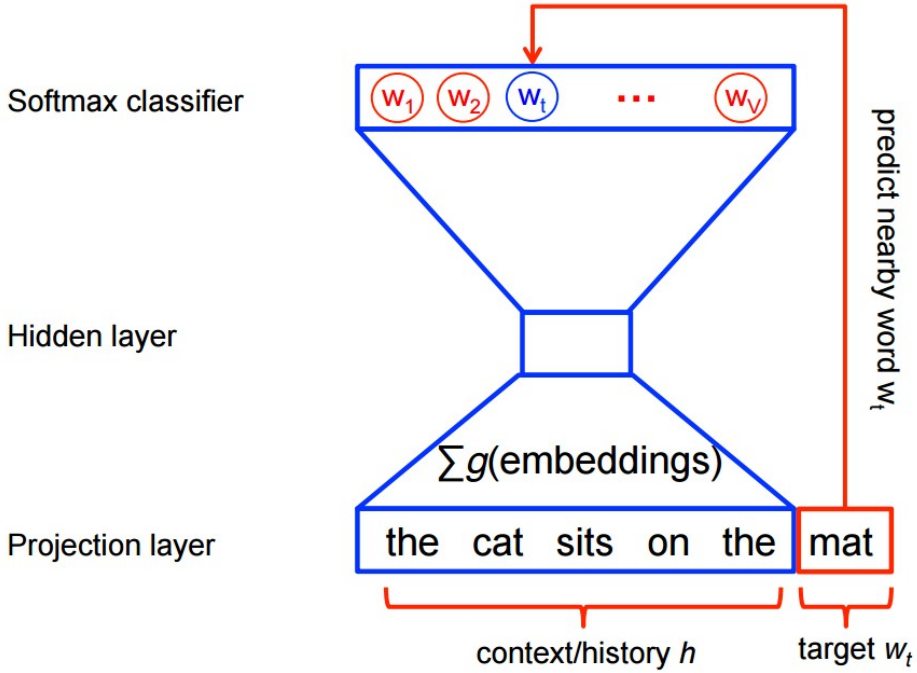
Cross-entropy loss, but softmax expensive! **Negative sampling**:

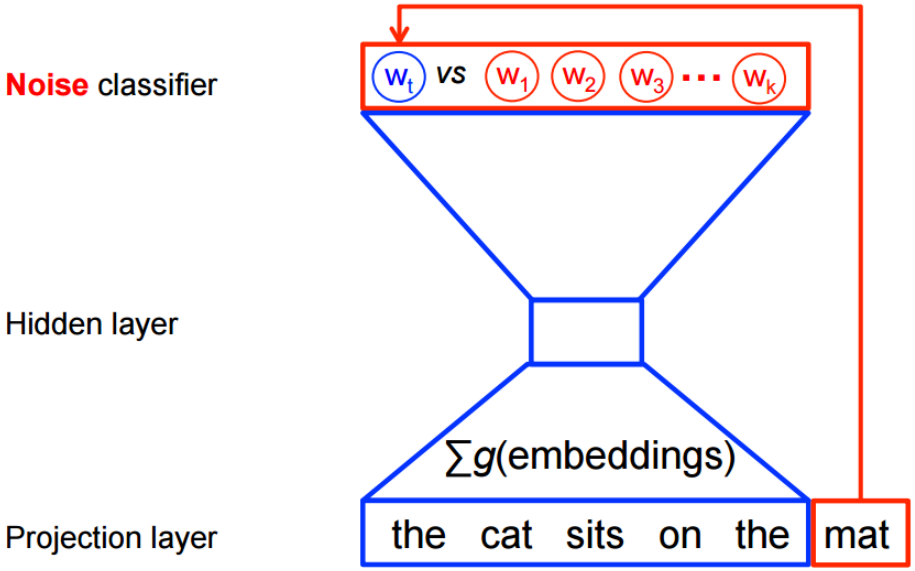
$$J_{\text{NEG}w(t)} = \log \sigma(\underline{h_{0i}}) + \sum_{k=1}^{K} E_{w_n \sim P_{noise}} \log \sigma(-\underline{h_{on}})$$

CBOW



Softmax vs. negative sampling

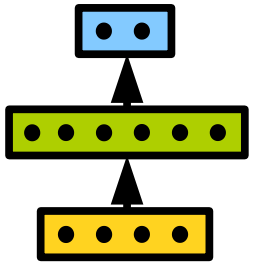




source: tensorflow

Skip-gram

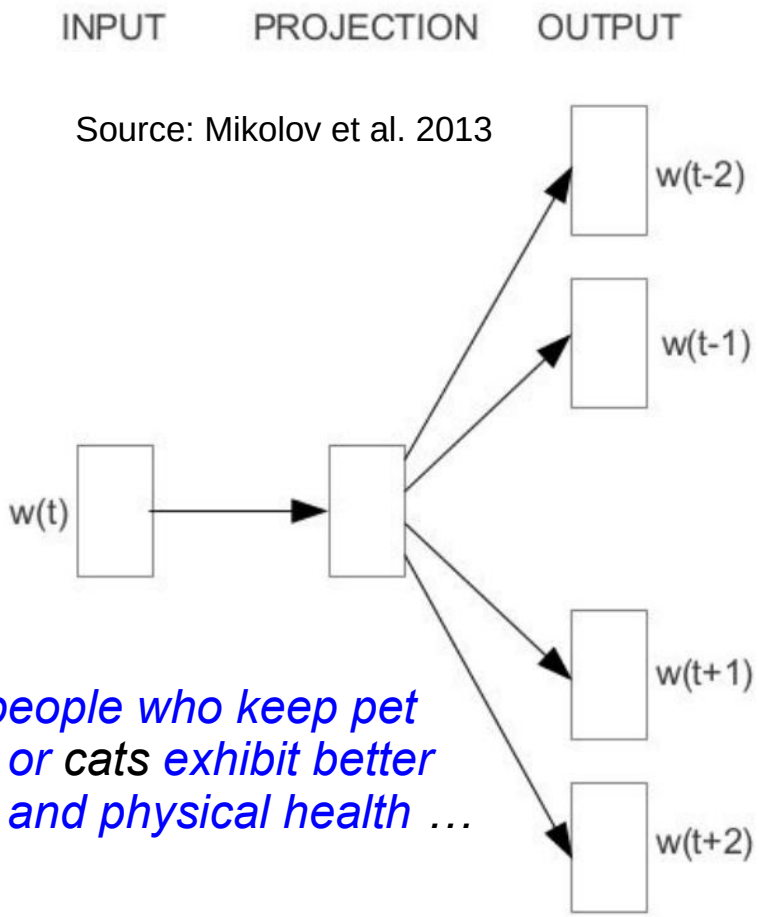
Like MLP, one layer, LARGE **vocabulary**



$$h_0 = W_0 x$$

$$y = softmax(W_1 h_0 + b_1)$$

$$y_i = \exp(\underline{h_{0i}}) / \sum_j^V \exp(\underline{h_{oi}})$$



Source: Mikolov et al. 2013

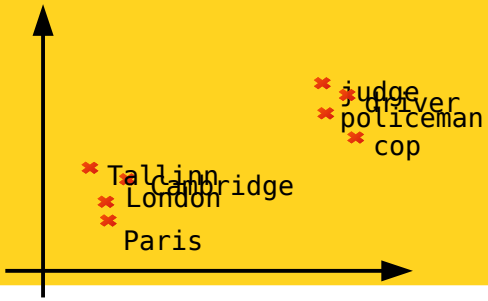
… people who keep pet dogs or cats exhibit better mental and physical health …

Losses:

$$J_{w(t)} = \frac{1}{n} \sum_{i=1}^{n} [\underline{h_{0i}} - \log \sum_j^V \exp(\underline{h_{oj}})]$$

$$J_{\text{NEG}w(t)} = \frac{1}{n} \sum_{i=1}^{n} [\log \sigma(\underline{h_{0i}}) + \sum_{k=1}^{K} E_{w_n \sim P_{noise}} \log \sigma(-\underline{h_{on}})]$$

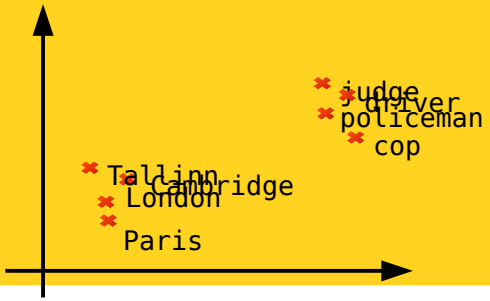
Word embeddings



Quality of word embeddings evaluated in:

- Word similarity
- Word analogy
- Other tasks like PoS tagging, NERC, sentiment analysis, etc.

Word embeddings

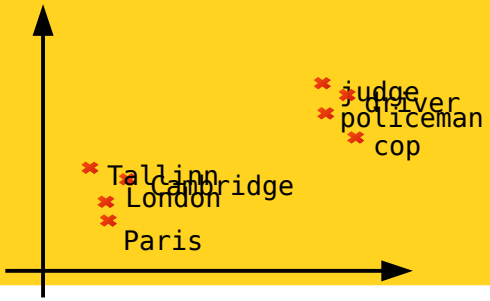


Word similarity

FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
454	1973	6909	11724	29869	87025
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

Source: Collobert et al. 2011

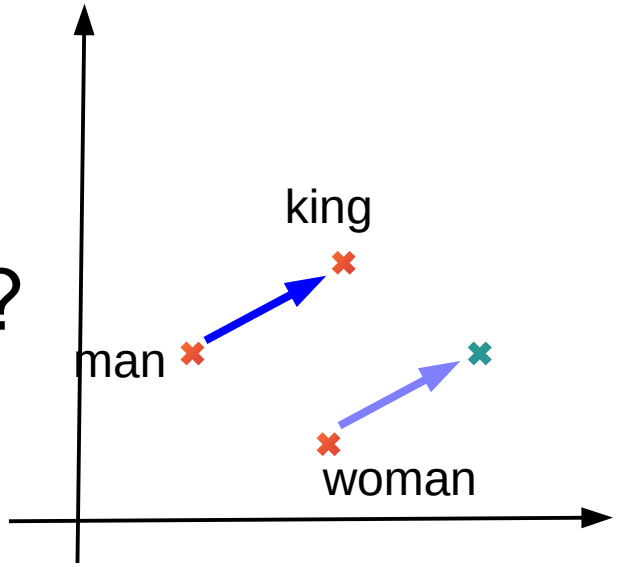
Word embeddings



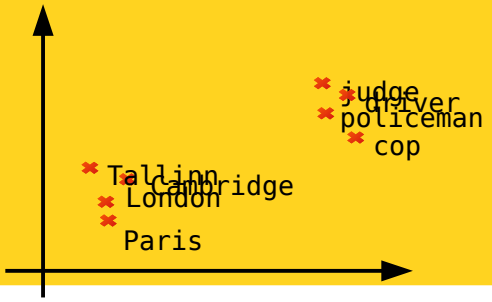
Word analogy:

a is to b as c is to ?

man is to king as woman is to ?



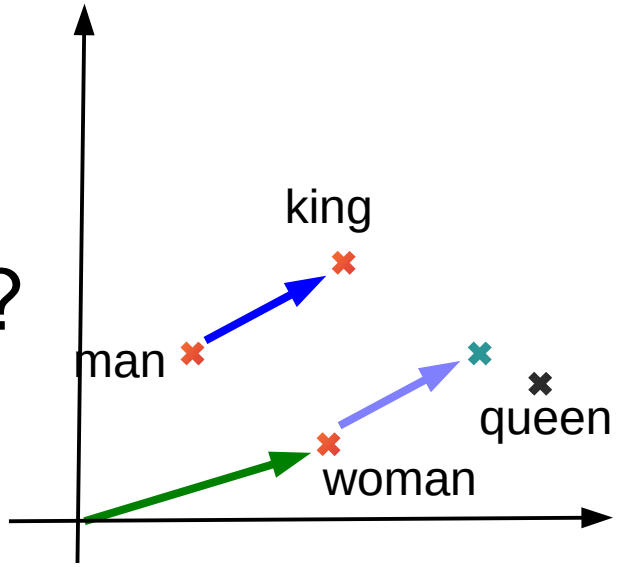
Word embeddings



Word analogy:

a is to b as c is to ?

man is to king as woman is to ?



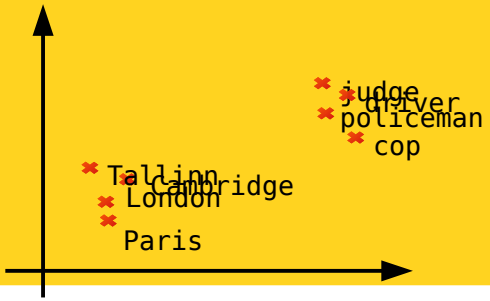
$$a - b \approx c - d$$

$$d \approx c - a + b$$

$$argmax_{d \in V}\left(\cos\left(d, c - a + b\right)\right)$$

$$king - man + woman \approx queen$$

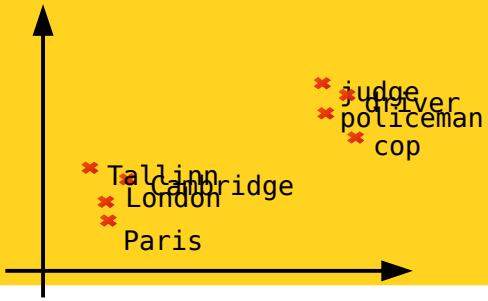
Word embeddings



Word analogy

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Word embeddings



Downstream tasks:
learn ML classifier/regressor reusing embeddings

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	SICK-R	SICK-E	STS14
word2vec BOW†	77.7	79.8	90.9	88.3	79.7	83.6	72.5/81.4	0.803	78.7	.65/.64
fastText BOW†	76.5	78.9	91.6	87.4	78.8	81.8	72.4/81.2	0.800	77.9	.63/.62
GloVe BOW†	**78.7**	78.5	**91.6**	87.6	79.8	83.6	72.1/80.9	0.800	78.6	.54/.56

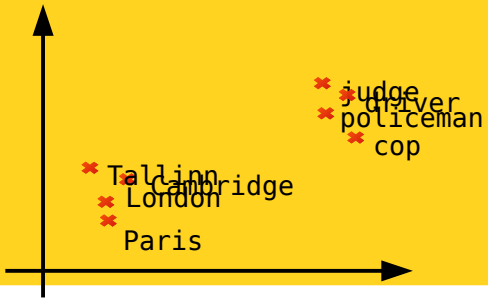
Source: Senteval (Conneau et al. 2017)

MR (sentiment analysis) CR (product reviews)
SUBJ (subjective/objective) MPQA (opinion polarity)
SST (sentiment analysis) TREC (question classification)
MRPC (paraphase) SICK-R (sentence similarity) SICK-E (inference) STS14 (sentence similarity)
MSRCOCO (caption retrieval – not shown here)

Word embeddings

- How to use embeddings in a given task:
  - From scratch (like in the MLP)
  - Initialize using embeddings from some other task (e.g. word2vec embeddings)
  - Initialize using embeddings from some other task but keep them fixed (block backpropagation)
- Other embeddings:
  - GloVe (Pennington et al. 2014)
  - Fastext (Mikolov et al. 2017)

Word embeddings



Many options?

Option 1: use co-occurrence matrix PPMI

- 1A: large sparse matrix
- **1B: factorize it and use low-rank dense matrix (SVD)**

Option 2: learn low-rank dense matrix directly

- 2A: MLP on particular classification task
- **2B: word2vec**

Goldberg & Levy (2014) show that the loss of skip-gram is related to factorizing PPMI with SVD.

Recap

- Are they useful for anything?
- Can we transfer this representation from one task to the other?
- Which task would be the best to learn embeddings that can be used in other tasks?
- Can we have all languages in one embedding space?

Plan for this session

- Representation learning
  - Word embeddings
  - ***Amazing abilities with word embeddings (Artetxe et al. 2018)***
- From words to sequences: Recurrent Neural Networks (RNN)

Cross-linguality and machine translation without bilingual data

Eneko Agirre
@eagirre

Joint work with: Mikel Artetxe, Gorka Labaka

IXA NLP group – University of the Basque Country (UPV/EHU)

http://ixa.eus

Motivation

Cross-lingual word representations:

- **Word embeddings key** for Natural Language Processing
- **Mapped embeddings** represent languages in a single space
  - Depend on seed **bilingual dictionaries**
- **Exciting results** in dictionary induction, transfer learning, crosslingual applications, interlingual semantic representations

Motivation

Cross-lingual word representations:

- **Word embeddings key** for Natural Language Processing
- **Mapped embeddings** represent languages in a single space
  - Depend on seed **bilingual dictionaries**
- **Exciting results** in dictionary induction, transfer learning, crosslingual applications, interlingual semantic representations

Our focus: **extend mappings to any pair of languages**

- Most language pairs have **very few bilingual resources**
- Key research area for **wide adoption** of NLP tools

Motivation

Cross-lingual word representations:

- **Word embeddings key** for Natural Language Processing
- **Mapped embeddings** represent languages in a single space
  - Depend on seed **bilingual dictionaries**
- **Exciting results** in dictionary induction, transfer learning, crosslingual applications, interlingual semantic representations

Our focus: **extend mappings to any pair of languages**

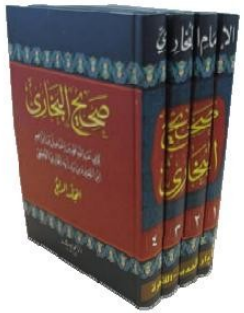
- Most language pairs have **very few bilingual resources**
- Key research area for **wide adoption** of NLP tools

In particular: **no bilingual resources at all**

- **Unsupervised** embedding mappings
- **Unsupervised** neural machine translation

Overview

Arabic monolingual corpora



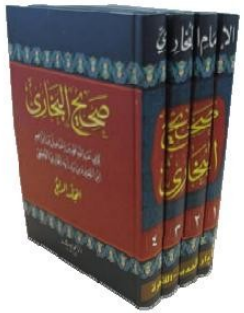
Arabic embeddings

Chinese monolingual corpora

Chinese embeddings

Overview

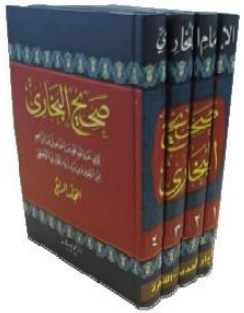
Arabic monolingual corpora



Chinese monolingual corpora

Overview

Arabic monolingual corpora



Chinese monolingual corpora

Arabic embeddings

Chinese embeddings

Bilingual embeddings

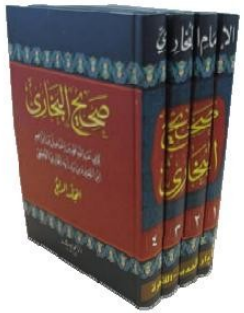
Bilingual dictionaries

Crosslingual & multilingual applications

Machine translation

Overview

Arabic monolingual corpora



No bilingual resource

Chinese monolingual corpora

Arabic embeddings

Chinese embeddings

Bilingual embeddings

Bilingual dictionaries

Crosslingual & multilingual applications

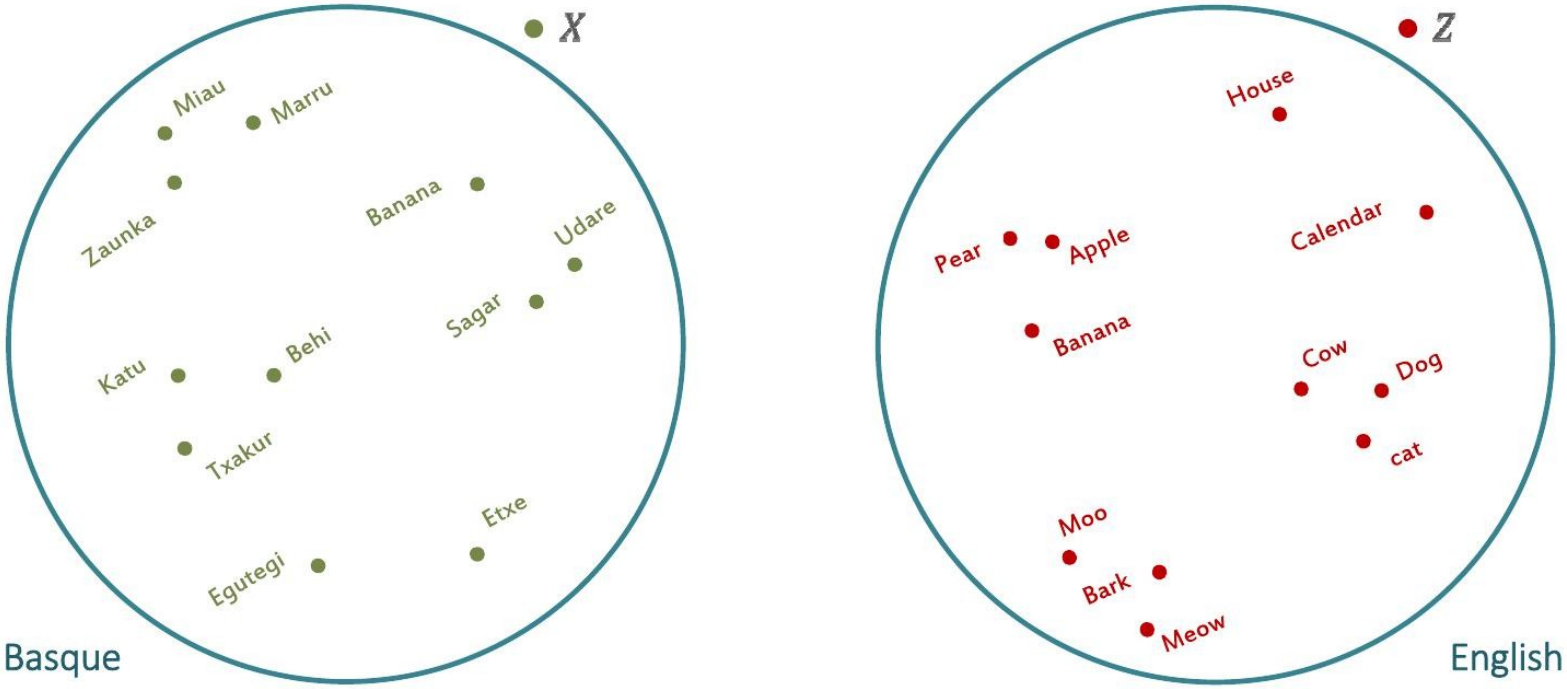
Machine translation

Outline

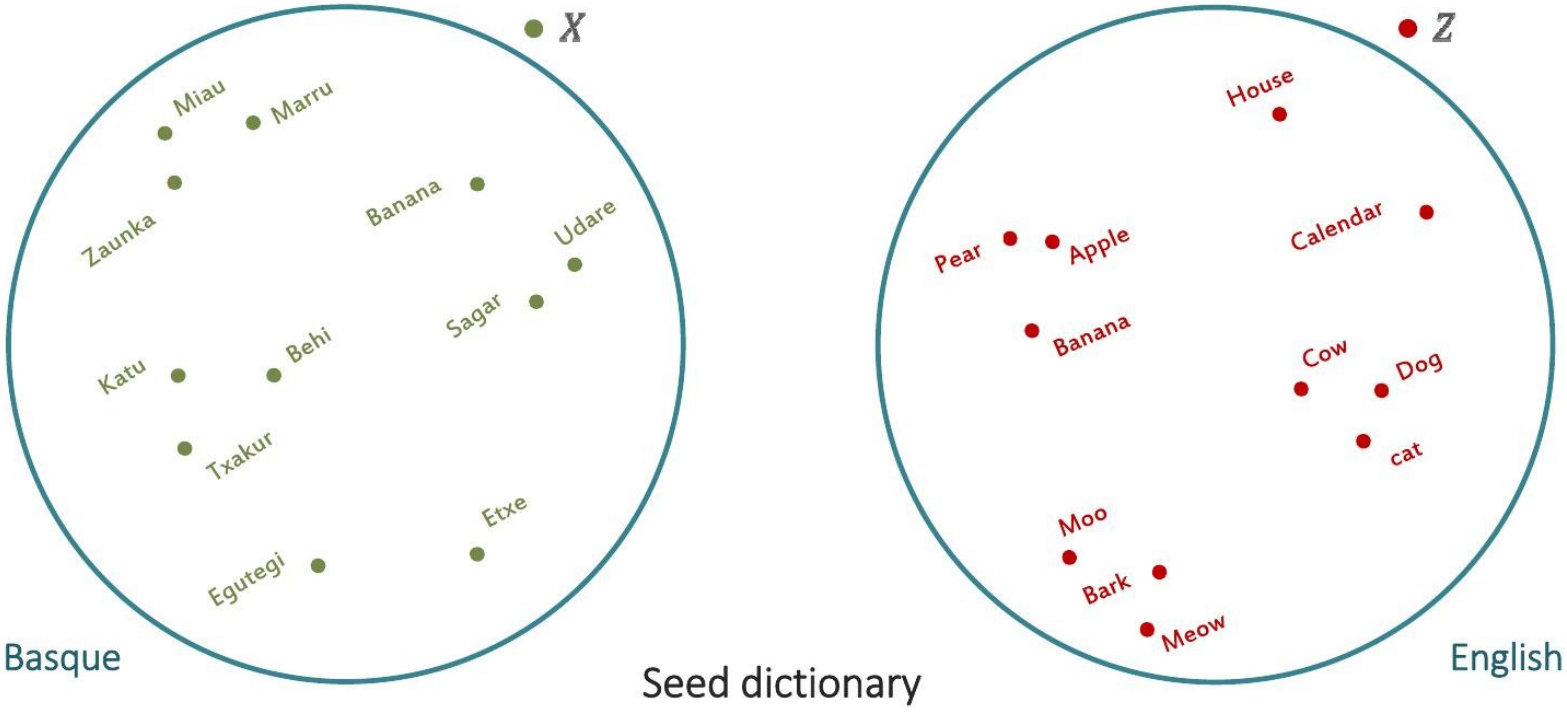
- Bilingual embedding mappings
  - *Introduction to vector space models (embeddings)*
  - *Bilingual embedding mappings* (AAAI18)
  - *Reduced supervision*
    - Self-learning, semi-supervised (ACL17)
    - Self-learning, fully unsupervised (ACL18)
  - *Conclusions*
- Unsupervised neural machine translation
  - *Introduction to NMT*
  - *From bilingual embeddings to uNMT (ICLR18)*
  - *Unsupervised statistical MT (EMNLP18)*
  - *Conclusions*

Introduction to embedding mappings

Introduction to embedding mappings



Introduction to embedding mappings

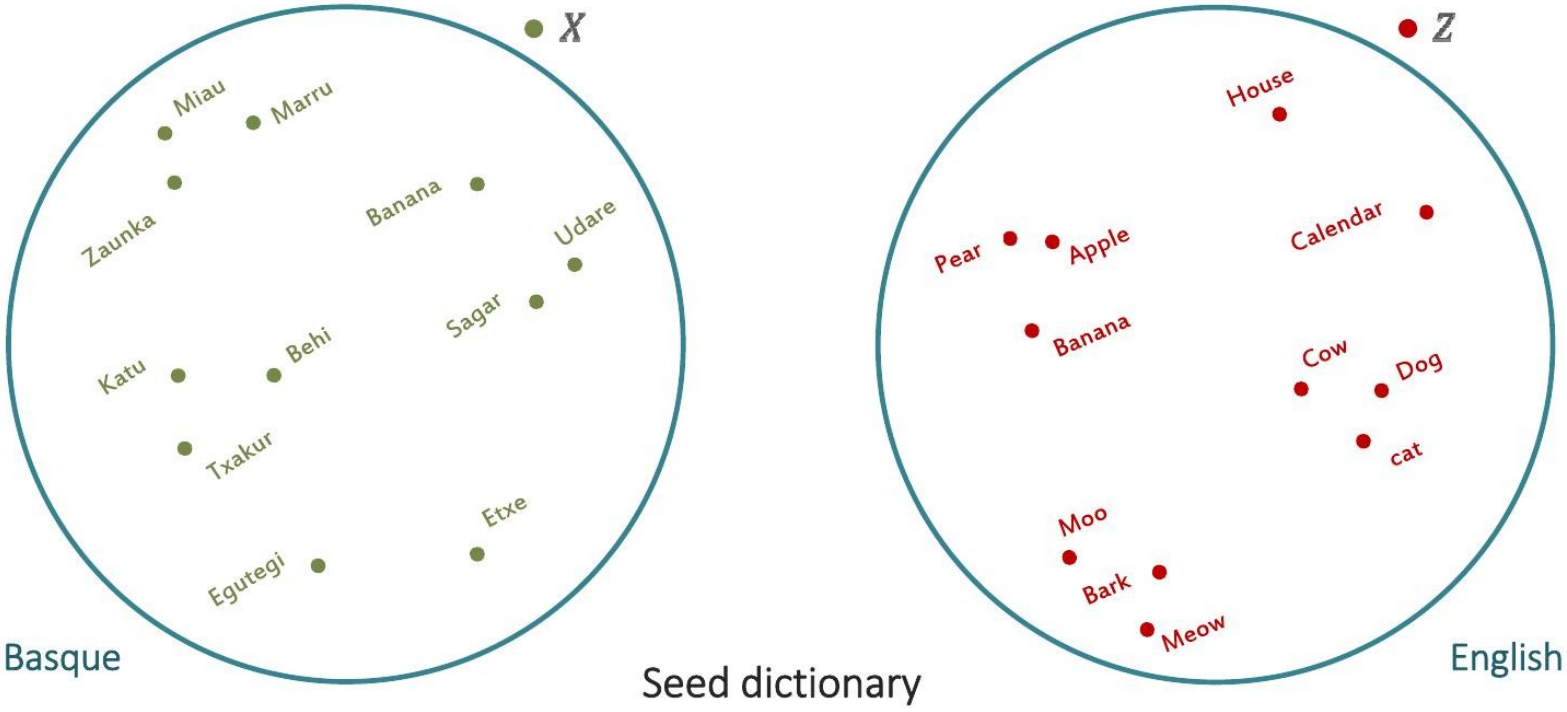


Basque

Seed dictionary

English

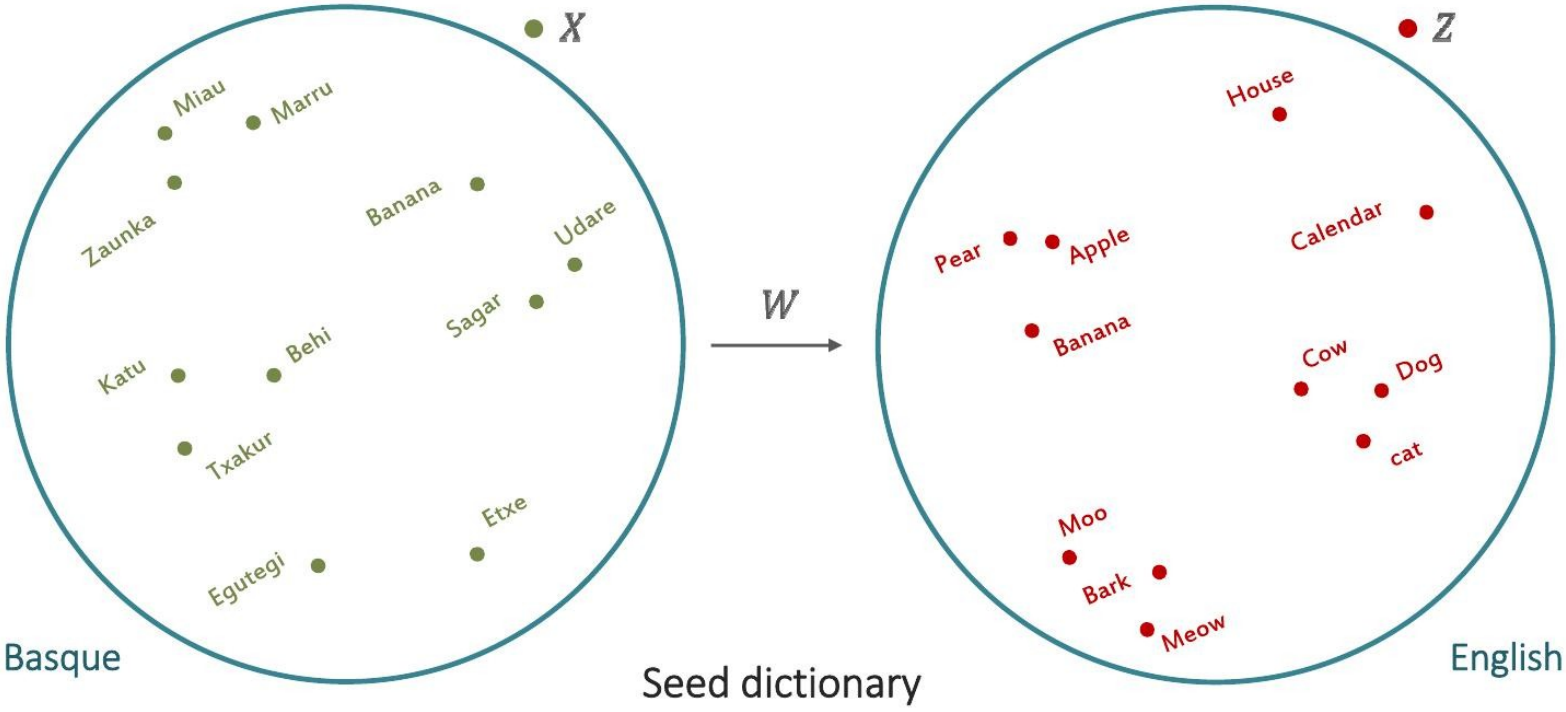
Introduction to embedding mappings



Seed dictionary

Txakur	Dog
Sagar	Apple
⋮	⋮
Egutegi	Calendar

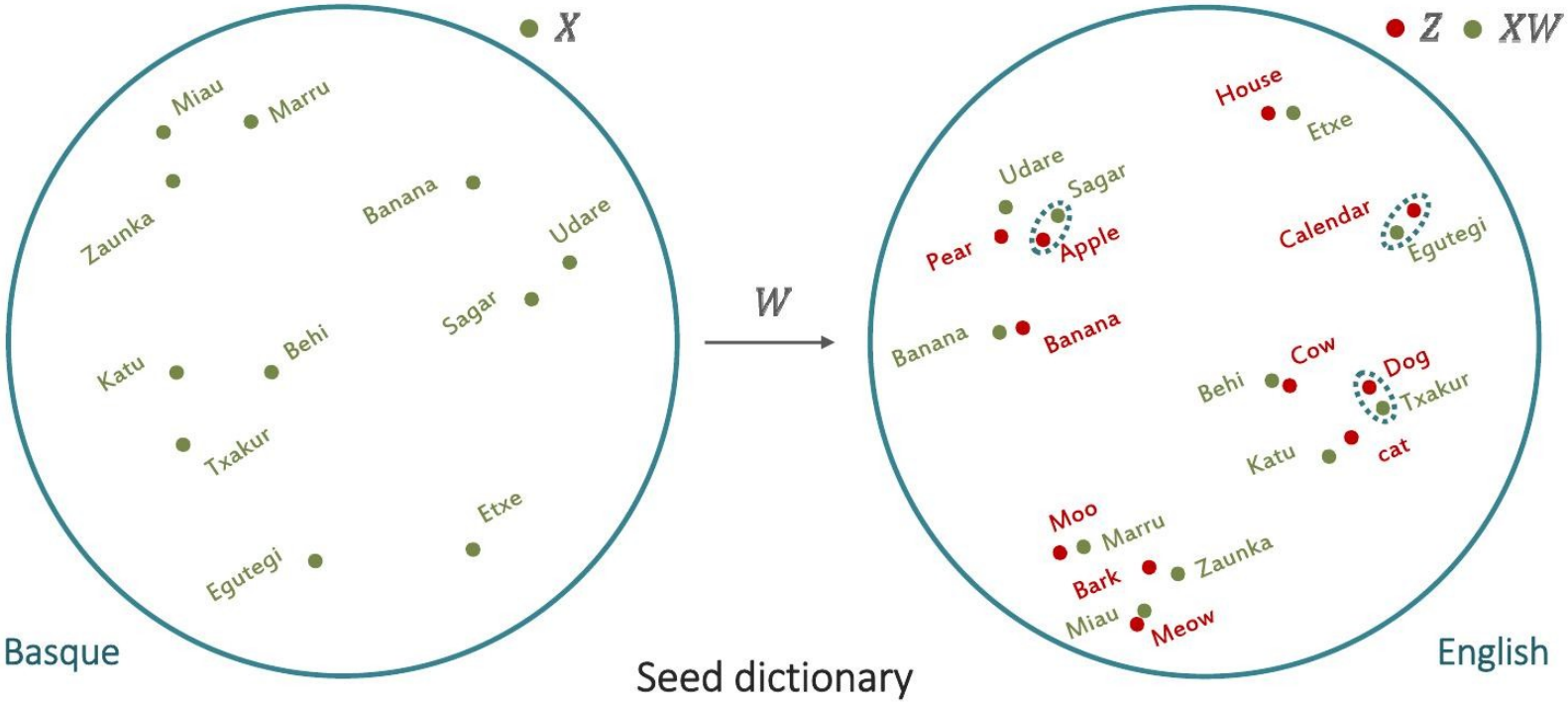
Introduction to embedding mappings



Seed dictionary

Txakur	Dog
Sagar	Apple
⋮	⋮
Egutegi	Calendar

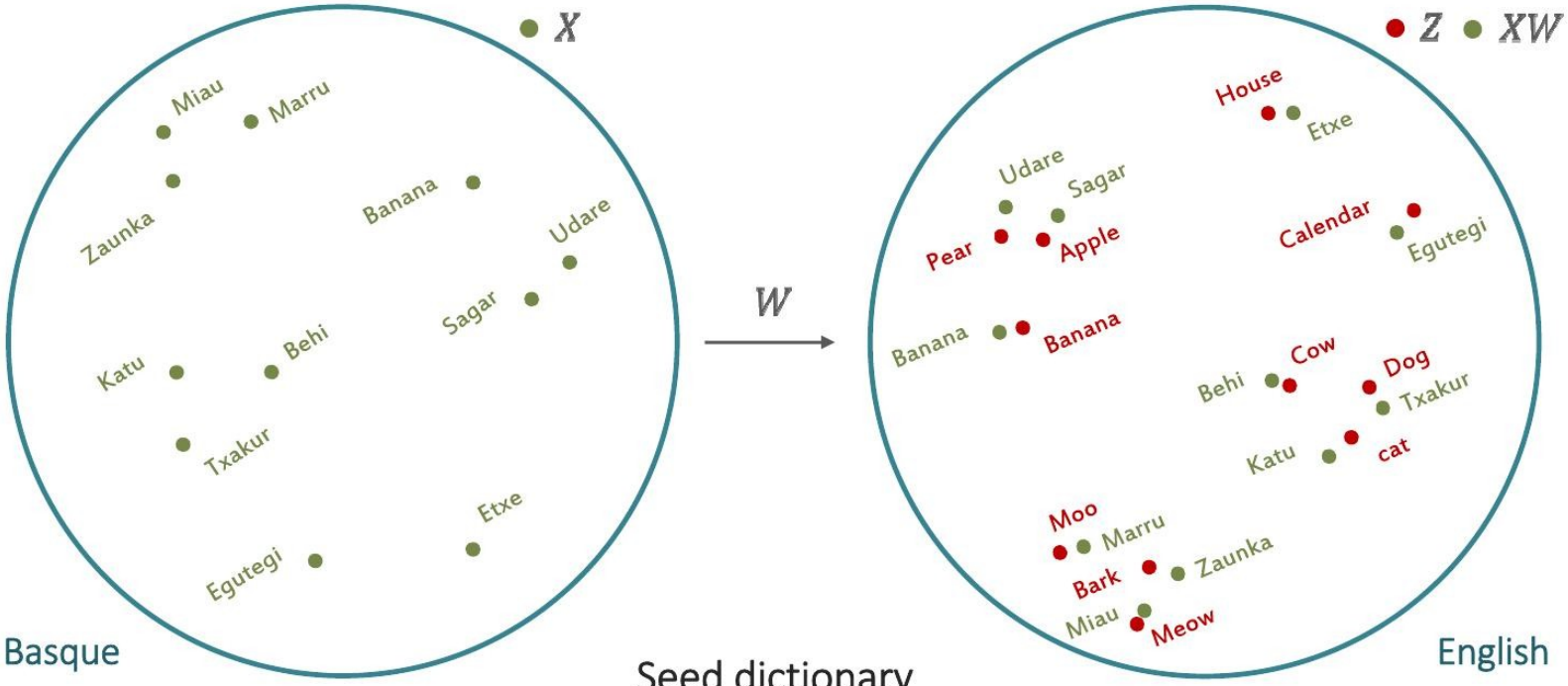
Introduction to embedding mappings



Seed dictionary

Txakur	Dog
Sagar	Apple
⋮	⋮
Egutegi	Calendar

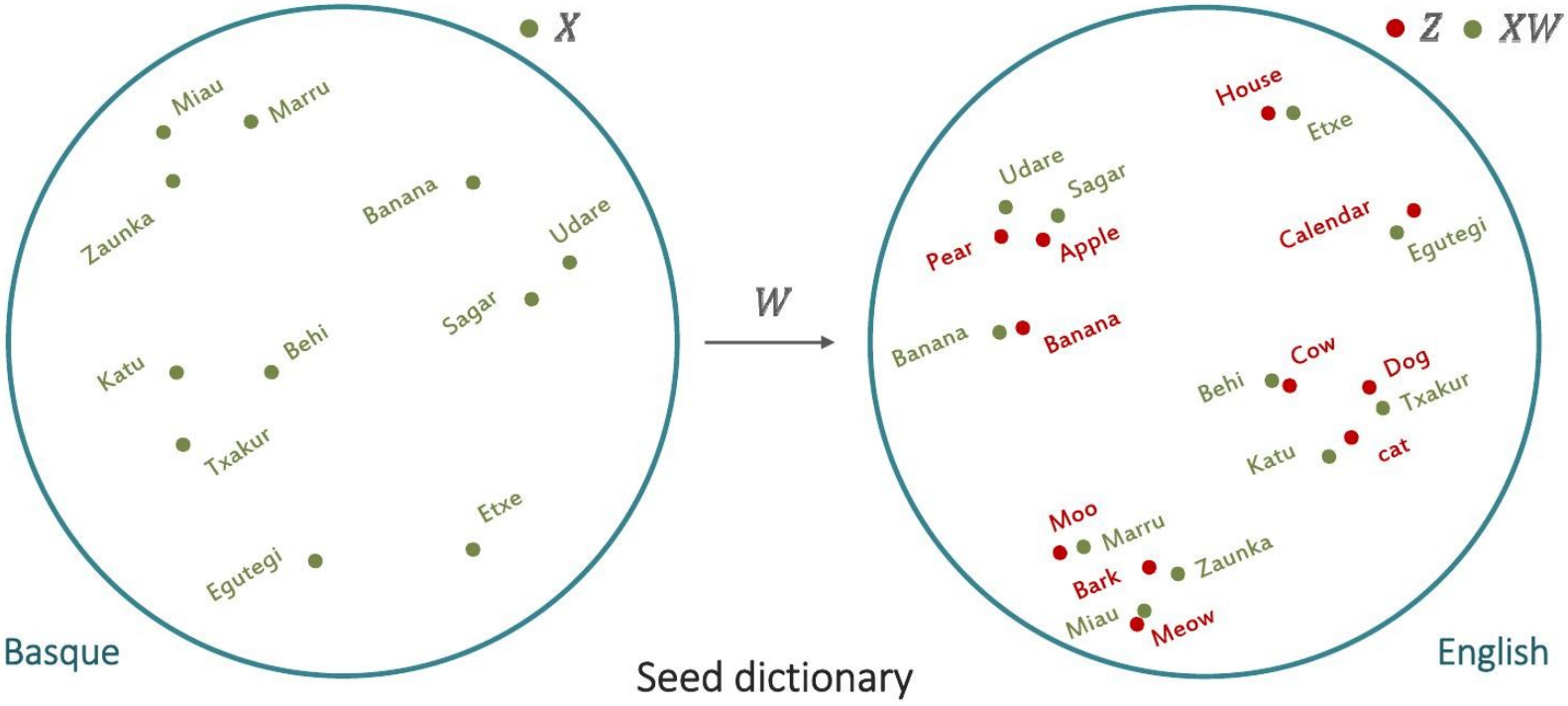
Introduction to embedding mappings



Seed dictionary

$$\begin{matrix} \text{Txakur} \\ \text{Sagar} \\ \vdots \\ \text{Egutegi} \end{matrix} \begin{bmatrix} X_{1,*} \\ X_{2,*} \\ \vdots \\ X_{n,*} \end{bmatrix} \quad \Big| \quad \begin{bmatrix} Z_{1,*} \\ Z_{2,*} \\ \vdots \\ Z_{n,*} \end{bmatrix} \begin{matrix} \text{Dog} \\ \text{Apple} \\ \vdots \\ \text{Calendar} \end{matrix}$$

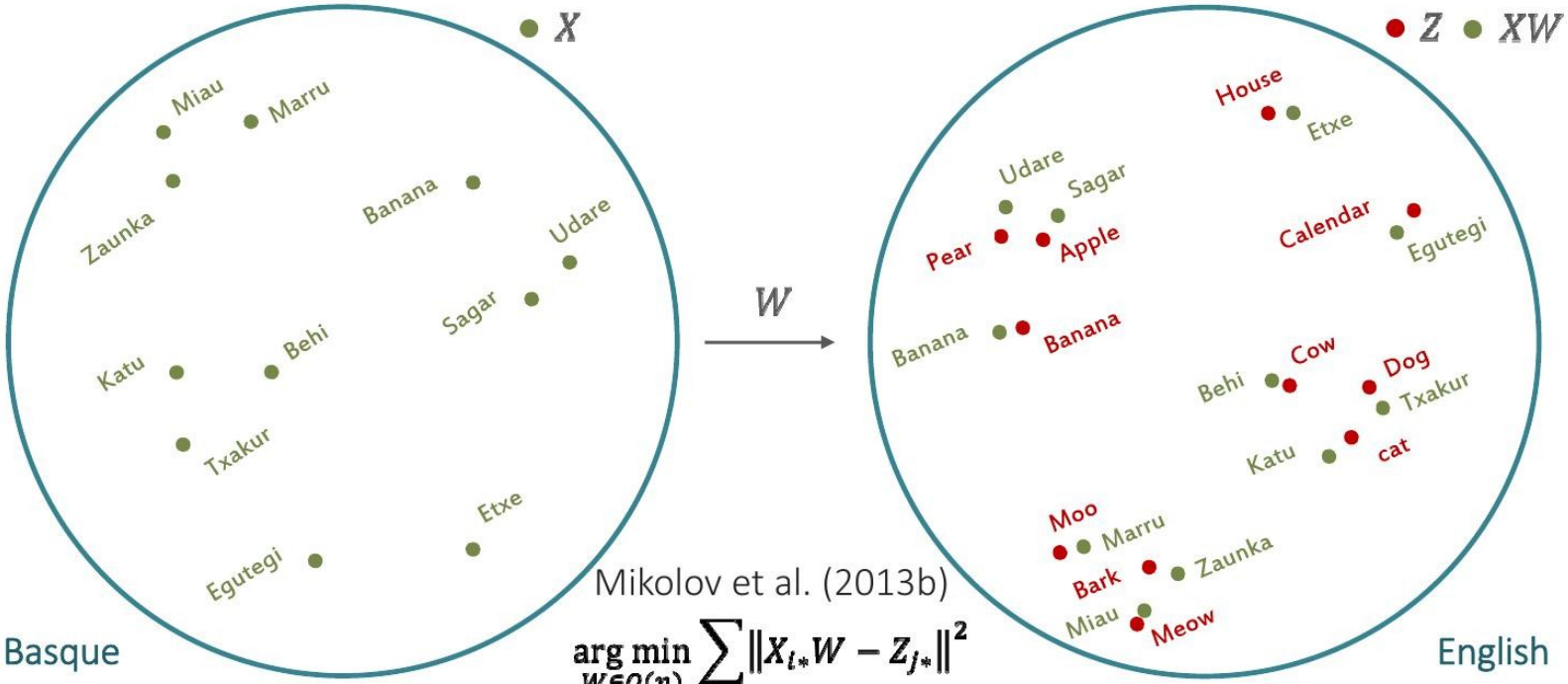
Introduction to embedding mappings



Seed dictionary

$$\begin{matrix} \text{Txakur} \\ \text{Sagar} \\ \vdots \\ \text{Egutegi} \end{matrix} \begin{bmatrix} X_{1,*} \\ X_{2,*} \\ \vdots \\ X_{n,*} \end{bmatrix} [W] \approx \begin{bmatrix} Z_{1,*} \\ Z_{2,*} \\ \vdots \\ Z_{n,*} \end{bmatrix} \begin{matrix} \text{Dog} \\ \text{Apple} \\ \vdots \\ \text{Calendar} \end{matrix}$$

Introduction to embedding mappings

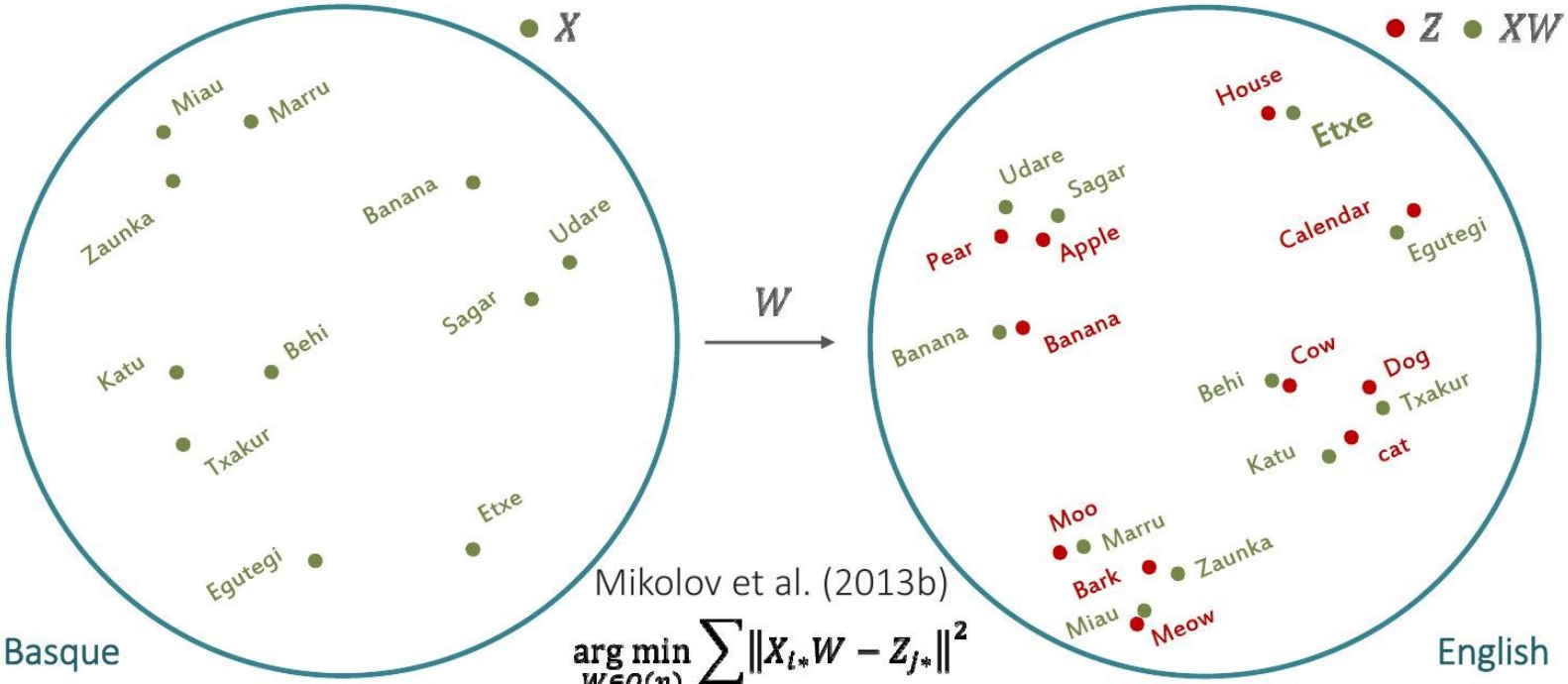


Mikolov et al. (2013b)

$$\arg\min_{W \in O(n)} \sum_{i} \|X_{i,*}W - Z_{j,*}\|^2$$

$$\begin{matrix} \text{Txakur} \\ \text{Sagar} \\ \vdots \\ \text{Egutegi} \end{matrix} \begin{bmatrix} X_{1,*} \\ X_{2,*} \\ \vdots \\ X_{n,*} \end{bmatrix} [W] \approx \begin{bmatrix} Z_{1,*} \\ Z_{2,*} \\ \vdots \\ Z_{n,*} \end{bmatrix} \begin{matrix} \text{Dog} \\ \text{Apple} \\ \vdots \\ \text{Calendar} \end{matrix}$$

Introduction to embedding mappings

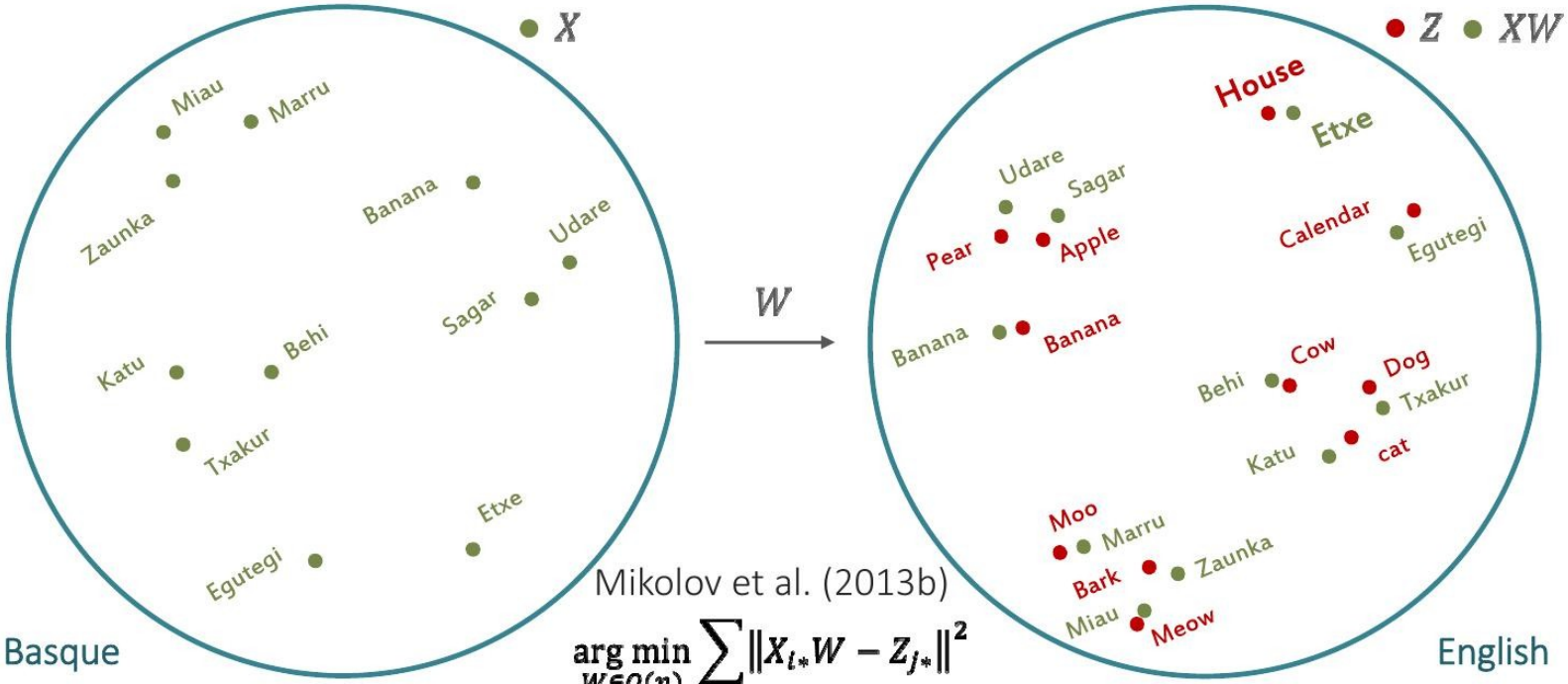


Mikolov et al. (2013b)

$$\arg\min_{W \in O(n)} \sum_{i} \|X_{i,*}W - Z_{j,*}\|^2$$

$$\begin{matrix} \text{Txakur} \\ \text{Sagar} \\ \vdots \\ \text{Egutegi} \end{matrix} \begin{bmatrix} X_{1,*} \\ X_{2,*} \\ \vdots \\ X_{n,*} \end{bmatrix} [W] \approx \begin{bmatrix} Z_{1,*} \\ Z_{2,*} \\ \vdots \\ Z_{n,*} \end{bmatrix} \begin{matrix} \text{Dog} \\ \text{Apple} \\ \vdots \\ \text{Calendar} \end{matrix}$$

Introduction to embedding mappings

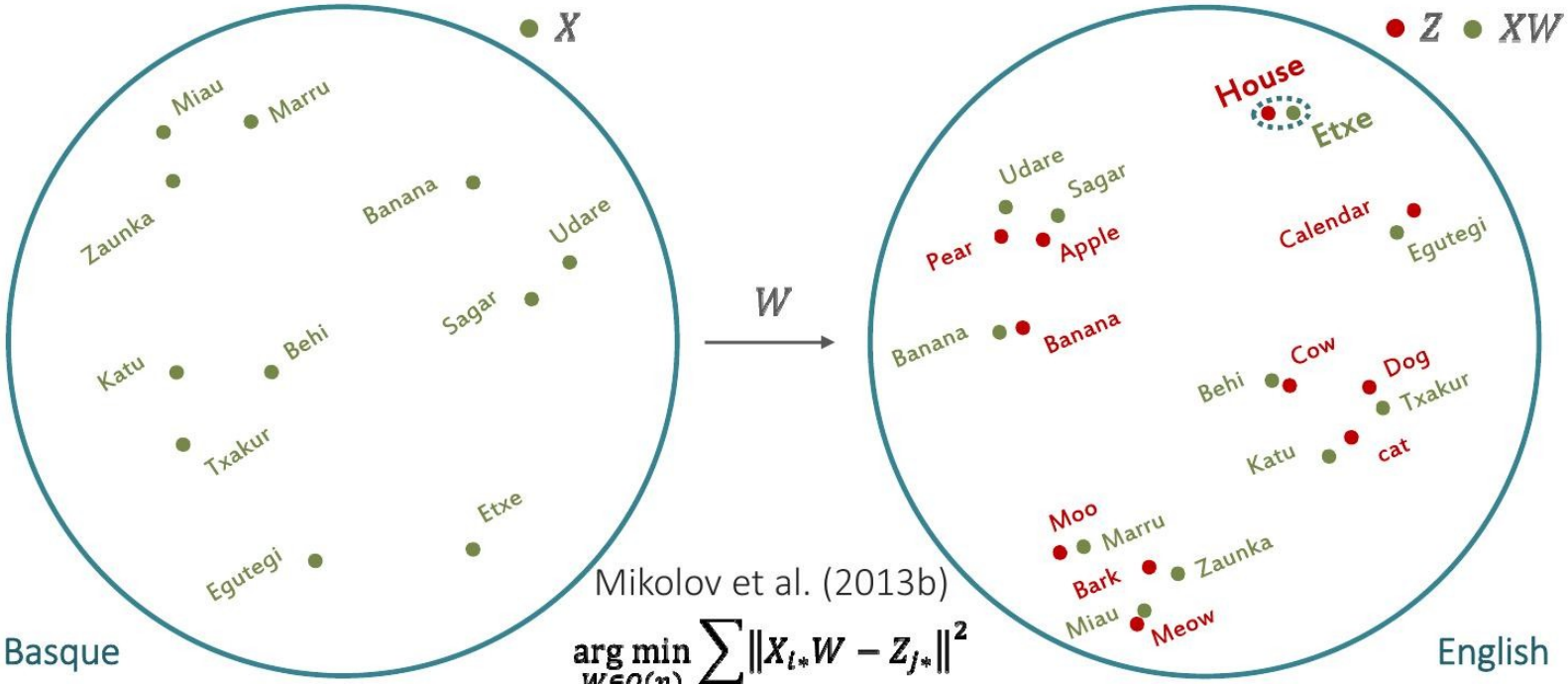


Mikolov et al. (2013b)

$$\arg\min_{W \in O(n)} \sum_{i} \|X_{i,*}W - Z_{j,*}\|^2$$

$$\begin{matrix} \text{Txakur} \\ \text{Sagar} \\ \vdots \\ \text{Egutegi} \end{matrix} \begin{bmatrix} X_{1,*} \\ X_{2,*} \\ \vdots \\ X_{n,*} \end{bmatrix} [W] \approx \begin{bmatrix} Z_{1,*} \\ Z_{2,*} \\ \vdots \\ Z_{n,*} \end{bmatrix} \begin{matrix} \text{Dog} \\ \text{Apple} \\ \vdots \\ \text{Calendar} \end{matrix}$$

Introduction to embedding mappings



Mikolov et al. (2013b)

$$\arg\min_{W \in O(n)} \sum_{i} \|X_{i,*}W - Z_{j,*}\|^2$$

$$\begin{matrix} \text{Txakur} \\ \text{Sagar} \\ \vdots \\ \text{Egutegi} \end{matrix} \begin{bmatrix} X_{1,*} \\ X_{2,*} \\ \vdots \\ X_{n,*} \end{bmatrix} [W] \approx \begin{bmatrix} Z_{1,*} \\ Z_{2,*} \\ \vdots \\ Z_{n,*} \end{bmatrix} \begin{matrix} \text{Dog} \\ \text{Apple} \\ \vdots \\ \text{Calendar} \end{matrix}$$

State-of-the-art in **supervised** mappings Artetxe et al. AAAI 2018

- Use 5000 sized seed bilingual dictionary
- Framework subsuming previous work that learns two mappings W_X W_Z as **sequences of (optional) linear mappings**:

  (opt.) Pre-process

  1. (opt.) Whitening
  2. **Orthogonal mapping**
  3. (opt.) Re-weighting
  4. (opt.) De-whitening

- The optional steps, properly combined, bring up to 5 points improvement

Evaluating via Bilingual Dictionary induction

Dataset by Dinu et al. (2015) extended to German, Finnish, Spanish

⇒ Monolingual embeddings (CBOW + negative sampling)

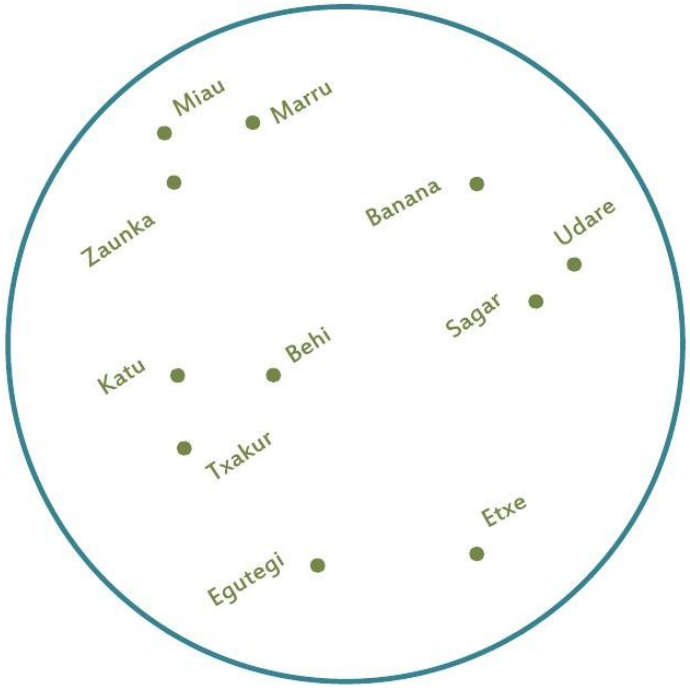
⇒ Seed dictionary: 5,000 pairs

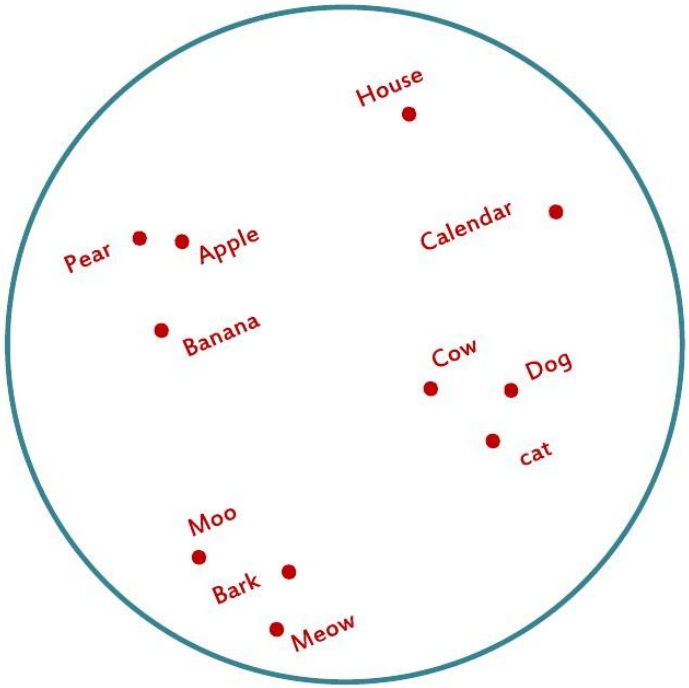
⇒ Test dictionary: 1,500 pairs (Nearest neighbor, P@1)

Method	EN-IT	EN-DE	EN-FI	EN-ES
Mikolov et al. (2013)	34.93†	35.00†	25.91†	27.73†
Faruqui and Dyer (2014)	38.40*	37.13*	27.60*	26.80*
Shigeto et al. (2015)	41.53†	43.07†	31.04†	33.73†
Dinu et al. (2015)	37.7	38.93*	29.14*	30.40*
Lazaridou et al. (2015)	40.2	-	-	-
Xing et al. (2015)	36.87†	41.27†	28.23†	31.20†
Artetxe et al. (2016)	39.27	41.87*	30.62*	31.40*
Zhang et al. (2016)	36.73†	40.80†	28.16†	31.07†
Smith et al. (2017)	43.1	43.33†	29.42†	35.13†
Our method (AAAI18)	**45.27**	**44.13**	**32.94**	**36.60**

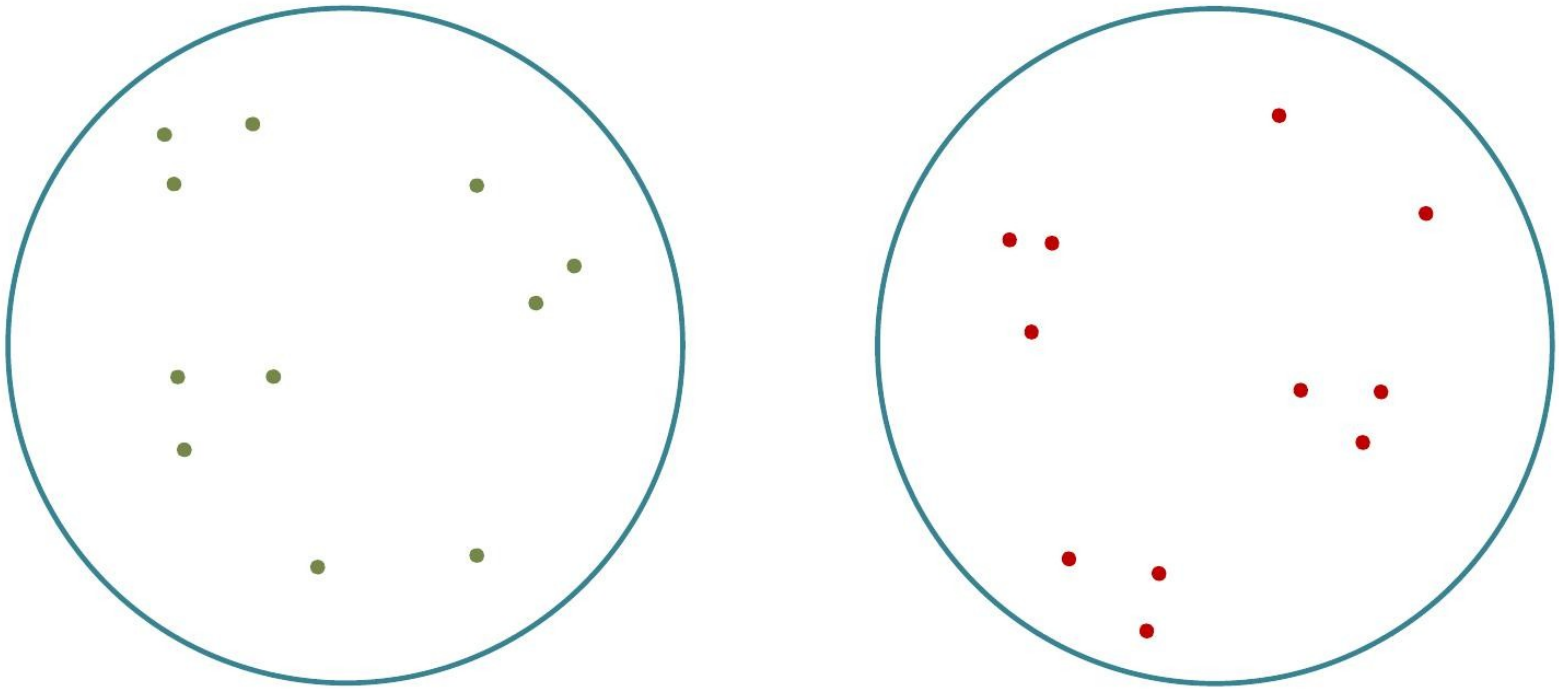
Why does it work?

Why does it work?

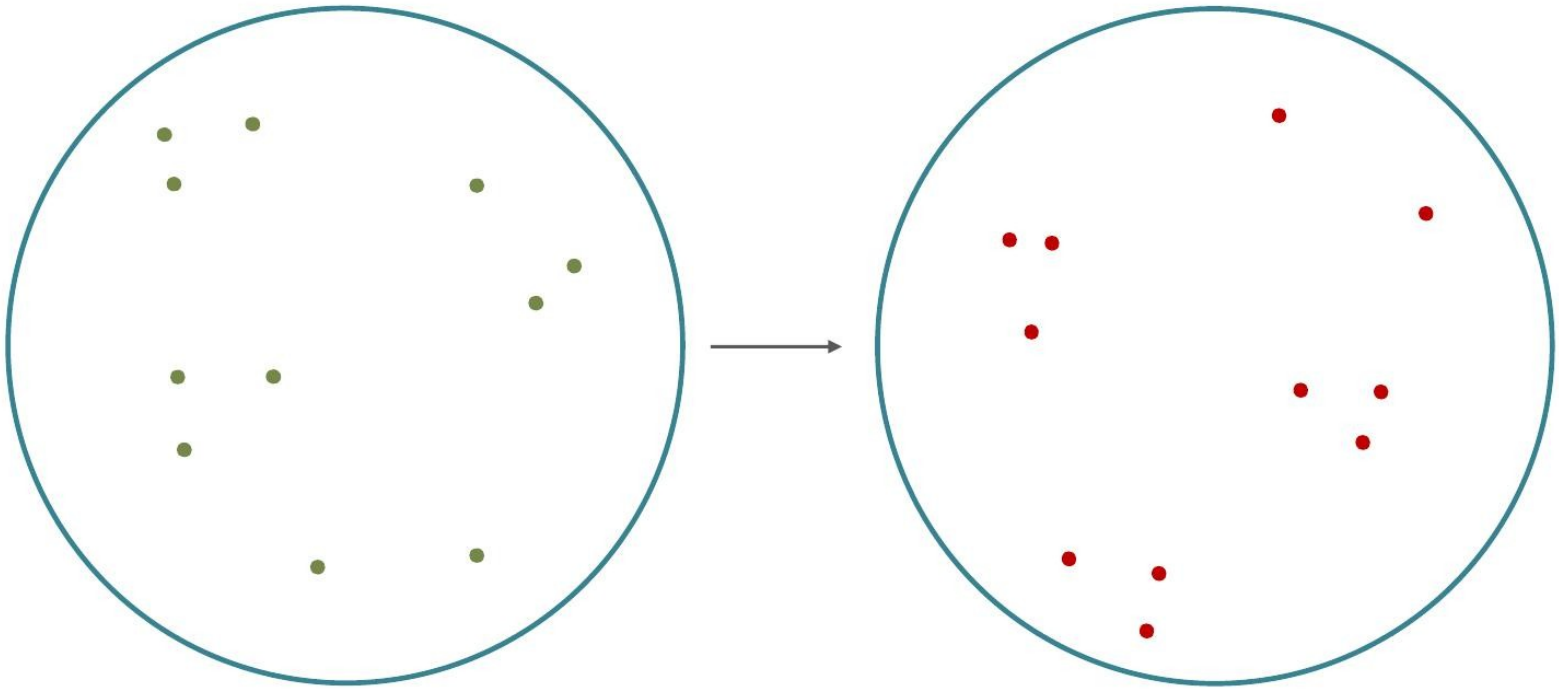




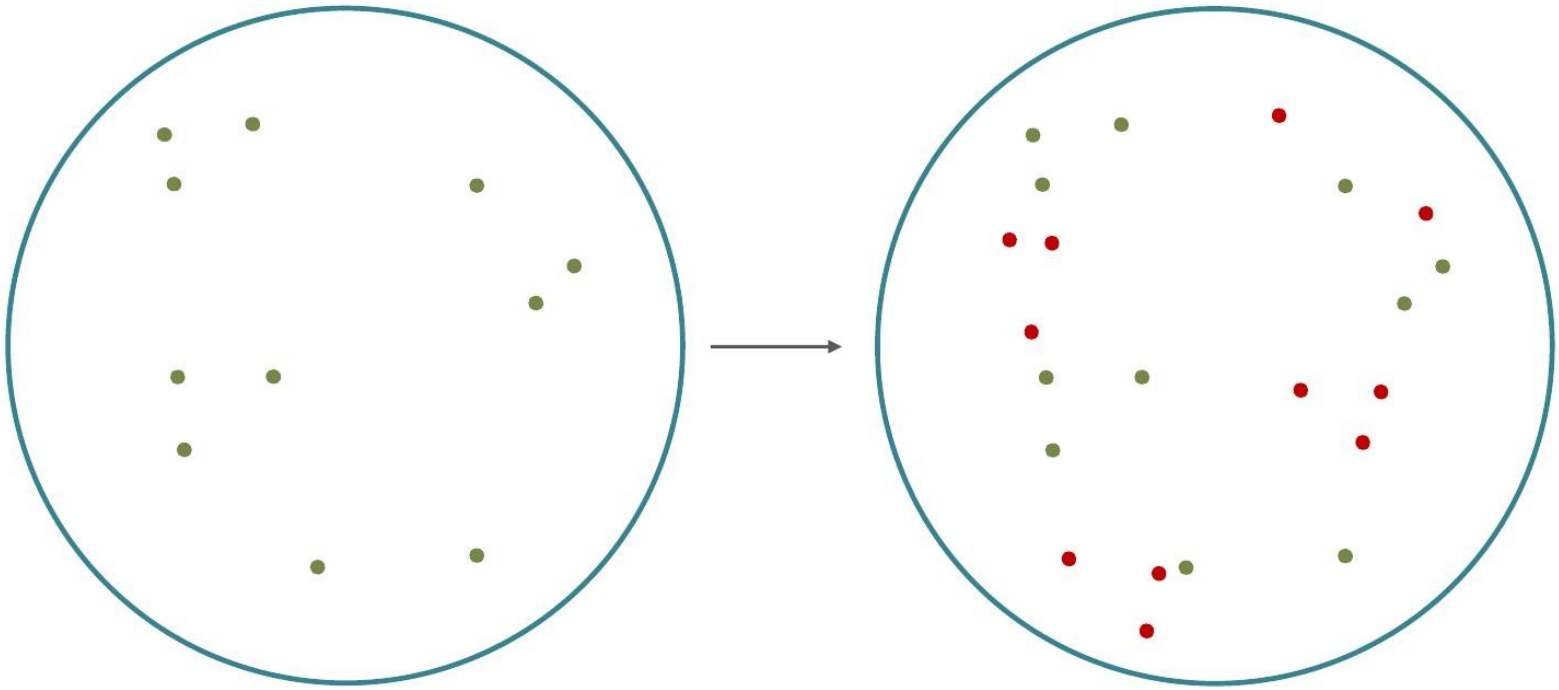
Why does it work?



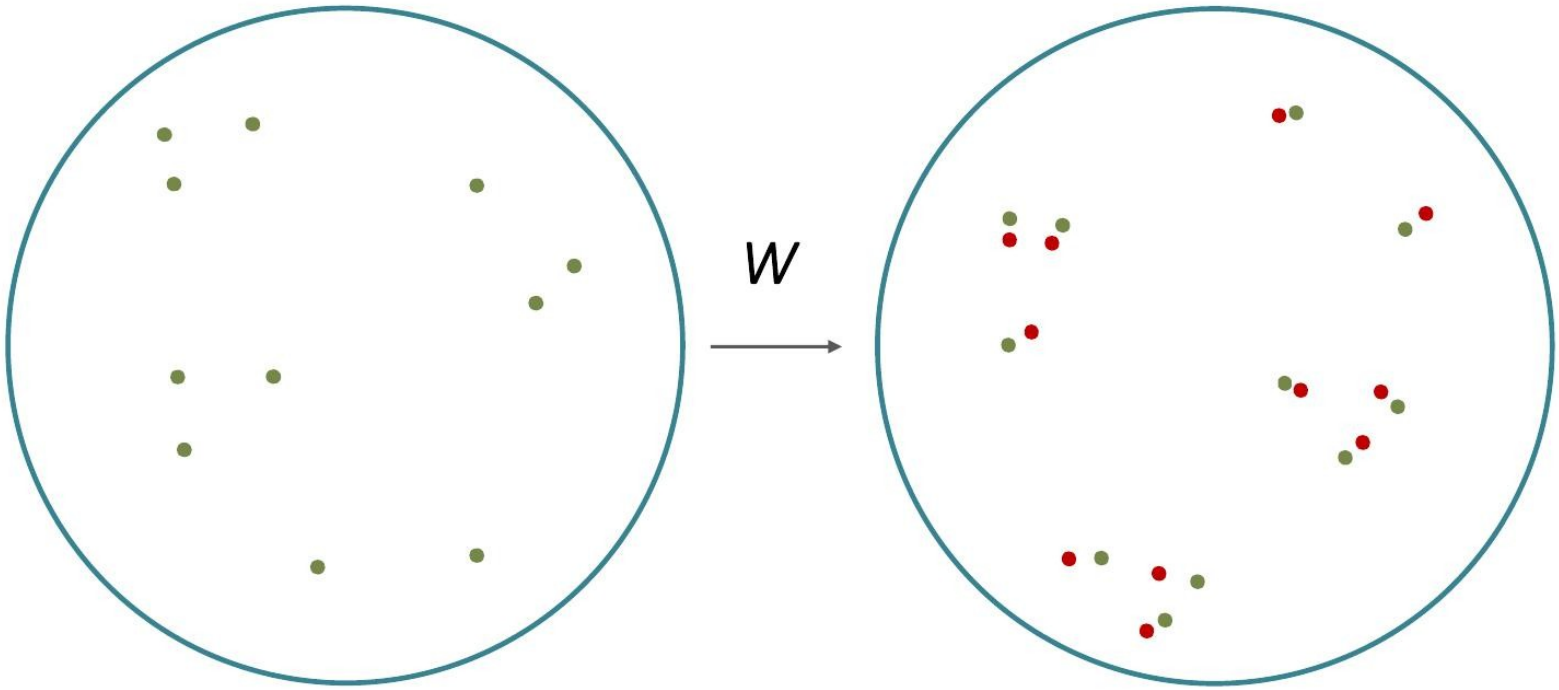
Why does it work?



Why does it work?

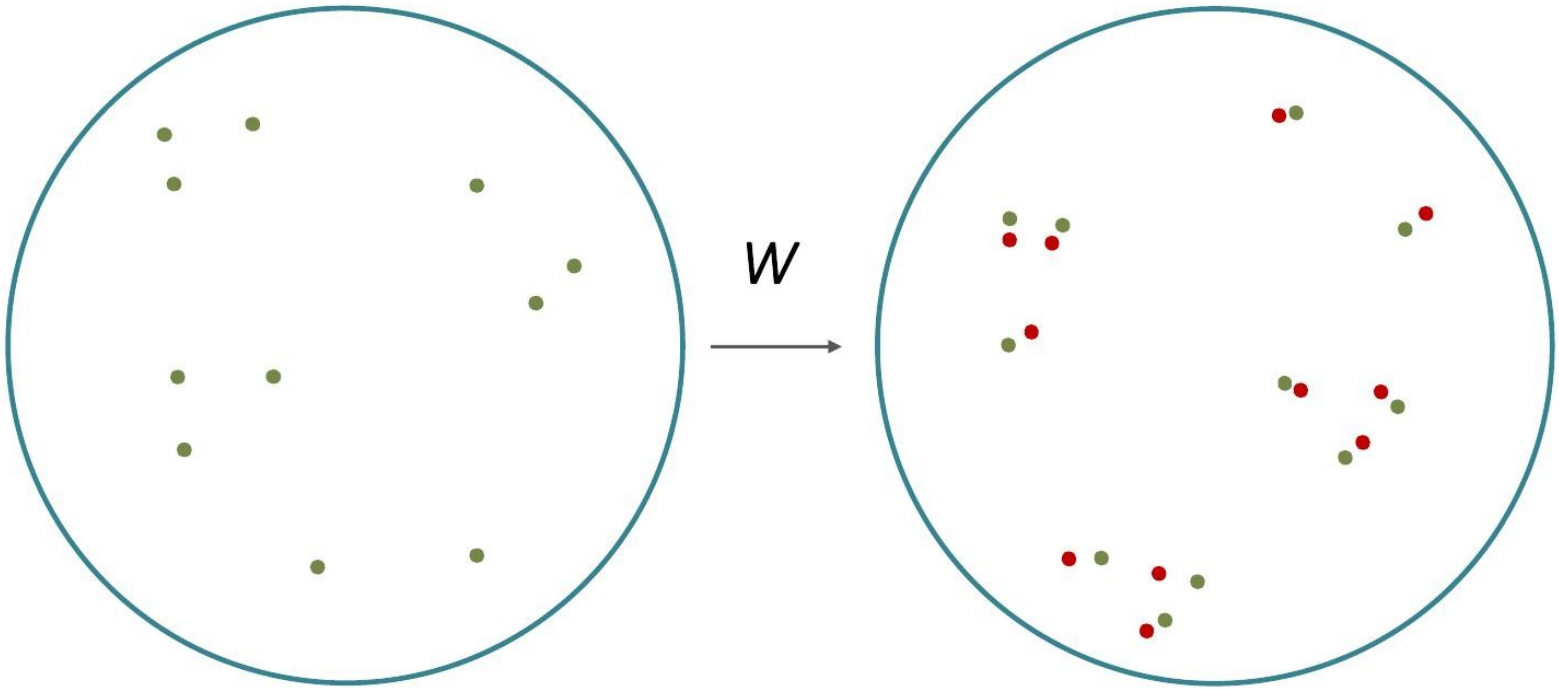


Why does it work?



Why does it work?

Languages are (to a large extent)
isometric in word embedding space (!)

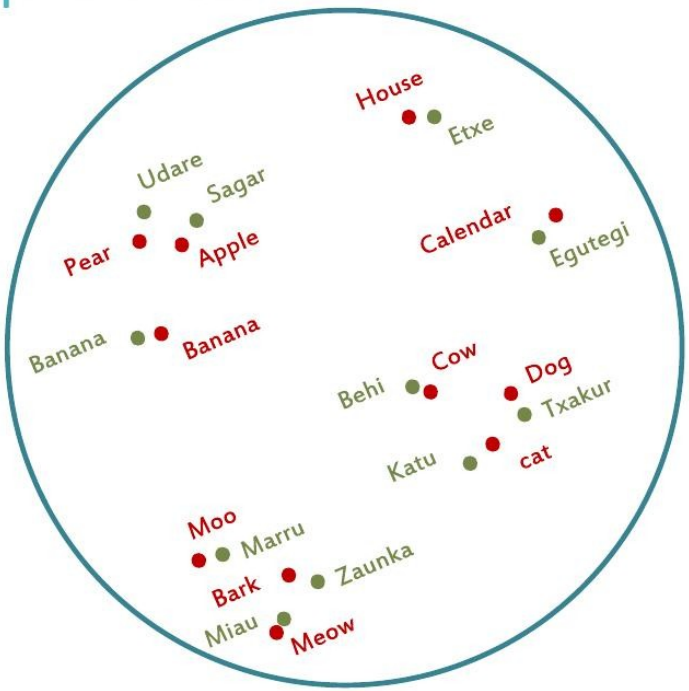


Outline

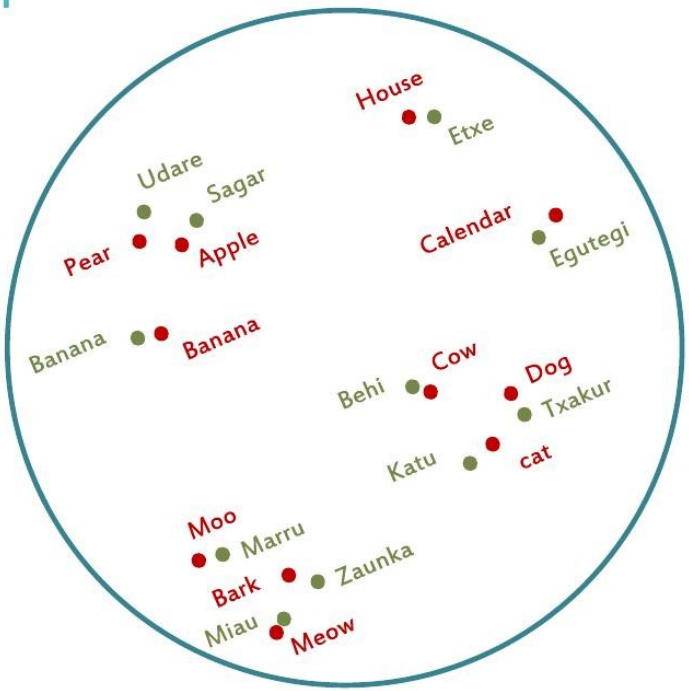
- Bilingual embedding mappings
  - *Introduction to vector space models (embeddings)*
  - *Bilingual embedding mappings (AAAI18)*
  - *Reduced supervision*
    - Self-learning, semi-supervised (ACL17)
    - Self-learning, fully unsupervised (ACL18)
  - *Conclusions*
- Unsupervised neural machine translation
  - *Introduction to NMT*
  - *From bilingual embeddings to uNMT (ICLR18)*
  - *Unsupervised statistical MT (EMNLP18)*
  - *Conclusions*

Reducing supervision

Reducing supervision



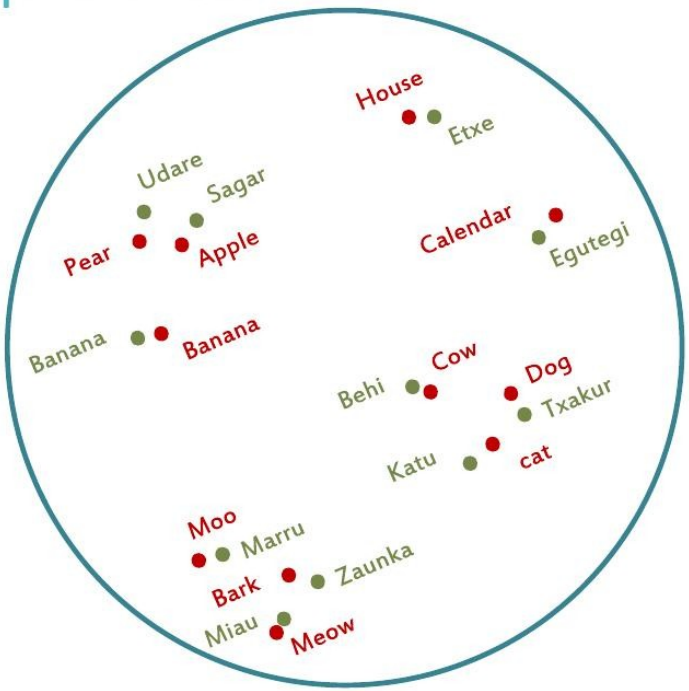
Reducing supervision



Previous work

bilingual signal for training

Reducing supervision

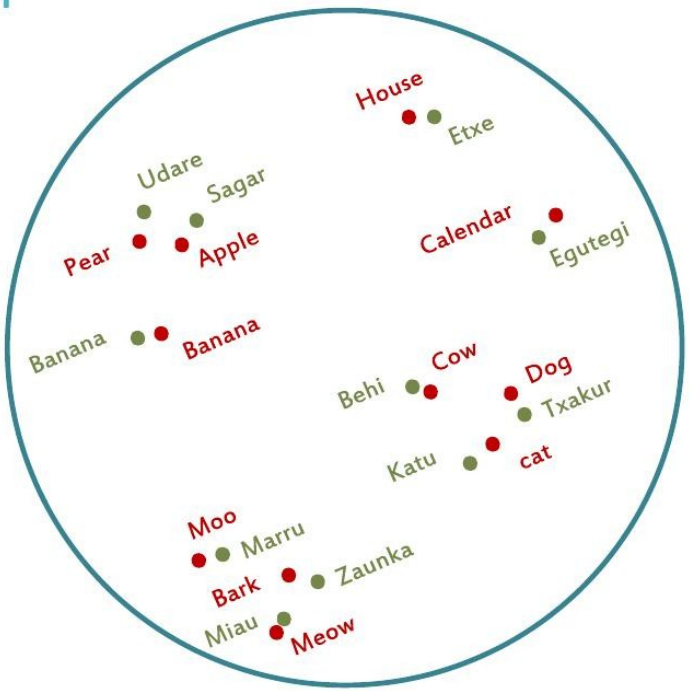


bilingual signal for training

Previous work

- parallel corpora
- comparable corpora
- (big) dictionaries

Reducing supervision

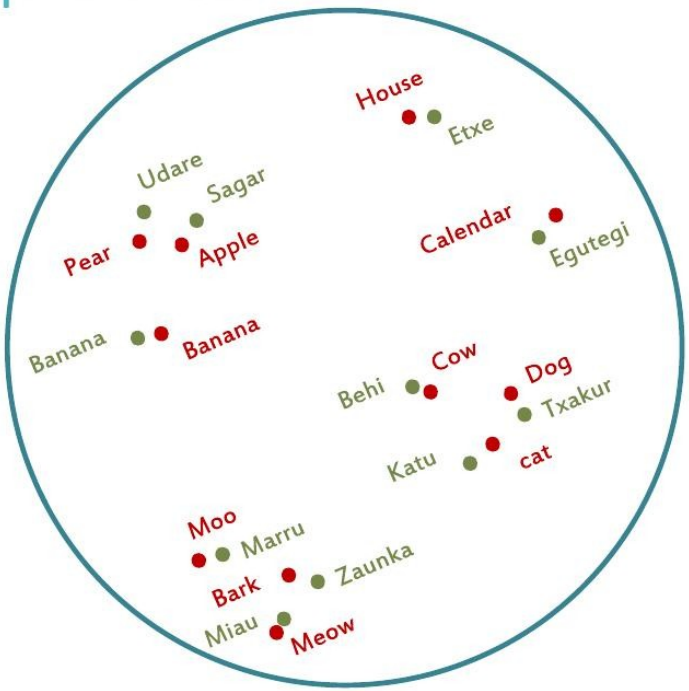


bilingual signal for training

Previous work

- parallel corpora
- comparable corpora
- (big) dictionaries

Reducing supervision



bilingual signal for training

Previous work	Our work
- parallel corpora	- 25 word dictionary
- comparable corpora	- numerals (1, 2, 3…)
- (big) dictionaries	- nothing

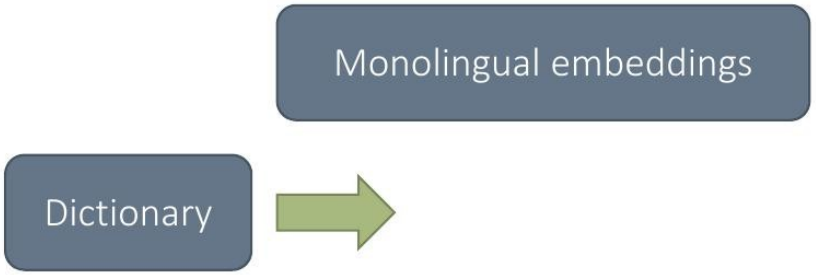
Self-learning

Self-learning

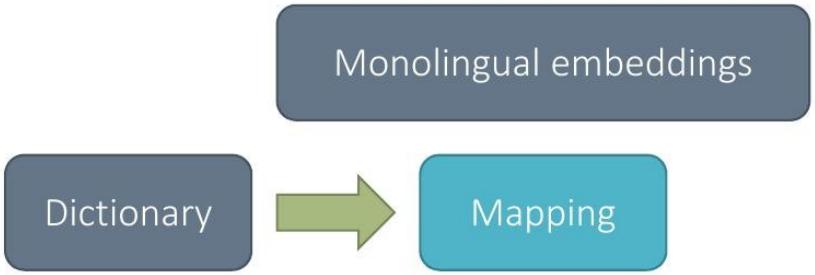
Monolingual embeddings

Self-learning

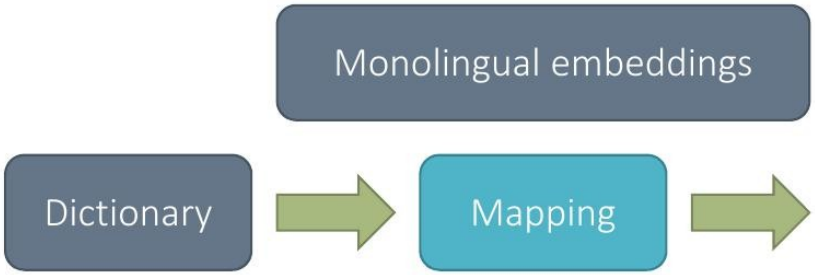
Self-learning



Self-learning

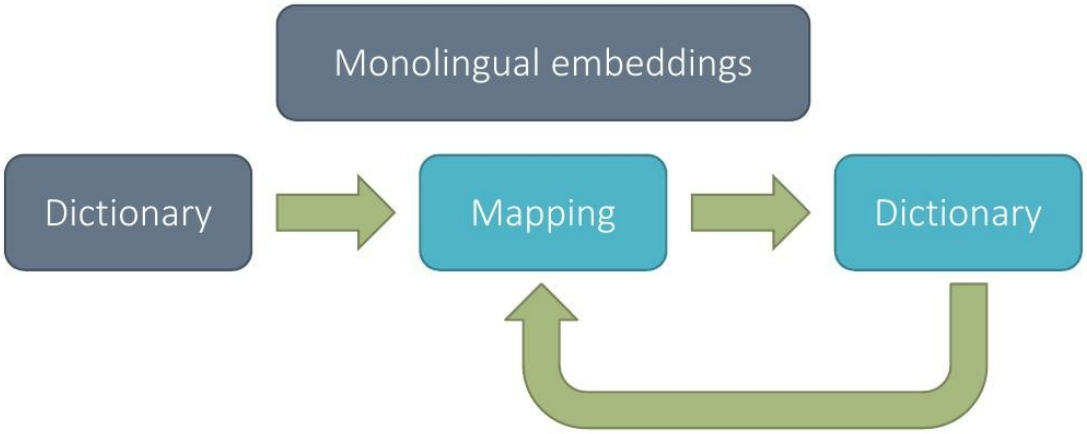


Self-learning



Self-learning

Self-learning



Unsupervised experiments (ACL18)

Unsupervised experiments (ACL18)

- Dataset by Dinu et al. (2015) extended German, Finnish, Spanish
  ⇒ *Monolingual embeddings (CBOW + negative sampling)*
  ⇒ *Seed dictionary: 5,000 word pairs / 25 word pairs / none*
  ⇒ *Test dictionary: 1,500 word pairs*

Supervision	Method	EN-IT	EN-DE	EN-FI	EN-ES
5k dict.	Mikolov et al. (2013)	34.93†	35.00†	25.91†	27.73†
	Artetxe et al. (2016)	39.27	41.87*	30.62*	31.40*
	Smith et al. (2017)	43.1	43.33†	29.42†	35.13†
	Our method (AAAI18)	45.27	44.13	**32.94**	36.60
25 dict.	Our method (ACL17)	37.27	39.60	28.16	-
None	Zhang et al. (2017)	0.00	0.00	0.01	0.01
	Conneau et al. (2018)	13.55	42.15	0.38	21.23
	Our method (ACL18)	**48.13**	**48.19**	32.63	**37.33**

Conclusions

- Simple self-learning method to train bilingual embedding mappings
- Unsupervised matches results of supervised methods!
- Implicit optimization objective independent from seed dictionary
- High quality dictionaries:
  Manual analysis shows that real accuracy > 60%
  High frequency words up to 80%
- Full reproducibility (including datasets):
  `https://github.com/artetxem/vecmap`

- Shows that languages share “semantic” structure to a large degree

References: cross-lingual mappings

- Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. **Generalizing and Improving Bilingual Word Embedding Mappings with a Multi-Step Framework of Linear Transformations**. In *AAAI-2018.*

- Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. **Learning bilingual word embeddings with (almost) no bilingual data**. In *ACL-2017*.

- Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. **A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings**. In *ACL-2018*.

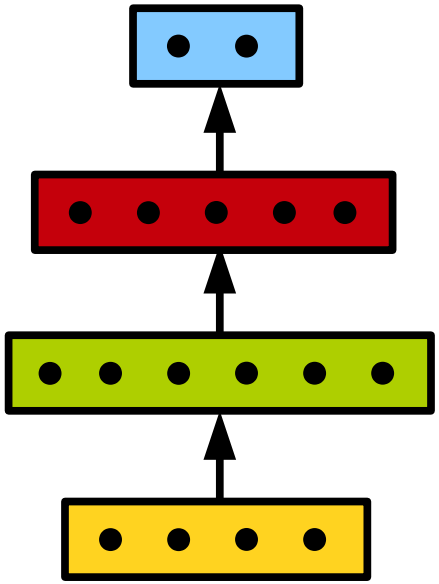
Plan for this session

- Representation learning
  - Word embeddings
  - *Amazing abilities with word embeddings (Artetxe et al. 2018)*
- **From words to sequences: Recurrent Neural Networks (RNN)**

From words to sequences

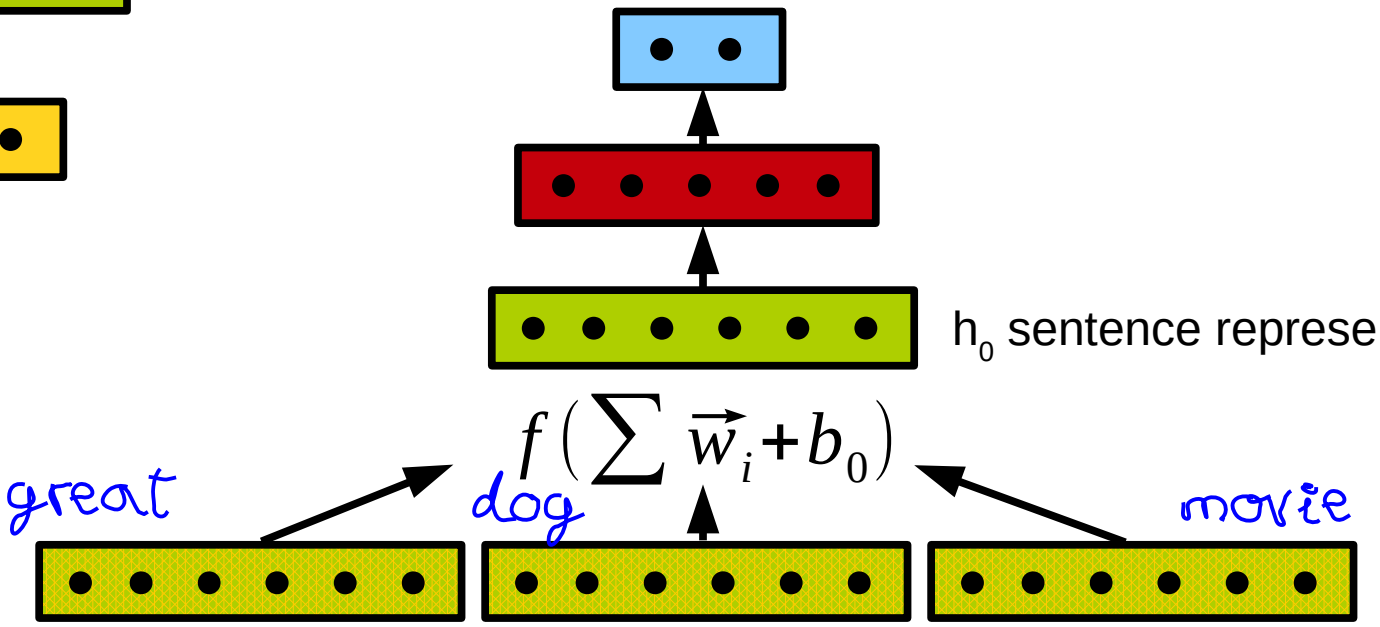
- Representation for words:
  one vector for each word (word embeddings)
- Representation for sequences of words:
  one vector for each sequence (?!)
  - Is it *possible* to represent a sentence in one vector at all?
  - Let's go back to MLP

From words to sequences



MLP: What is h_0 with respect to the words in the input ?

- Add vectors of words in context (1's in x), plus bias, apply non-linearity

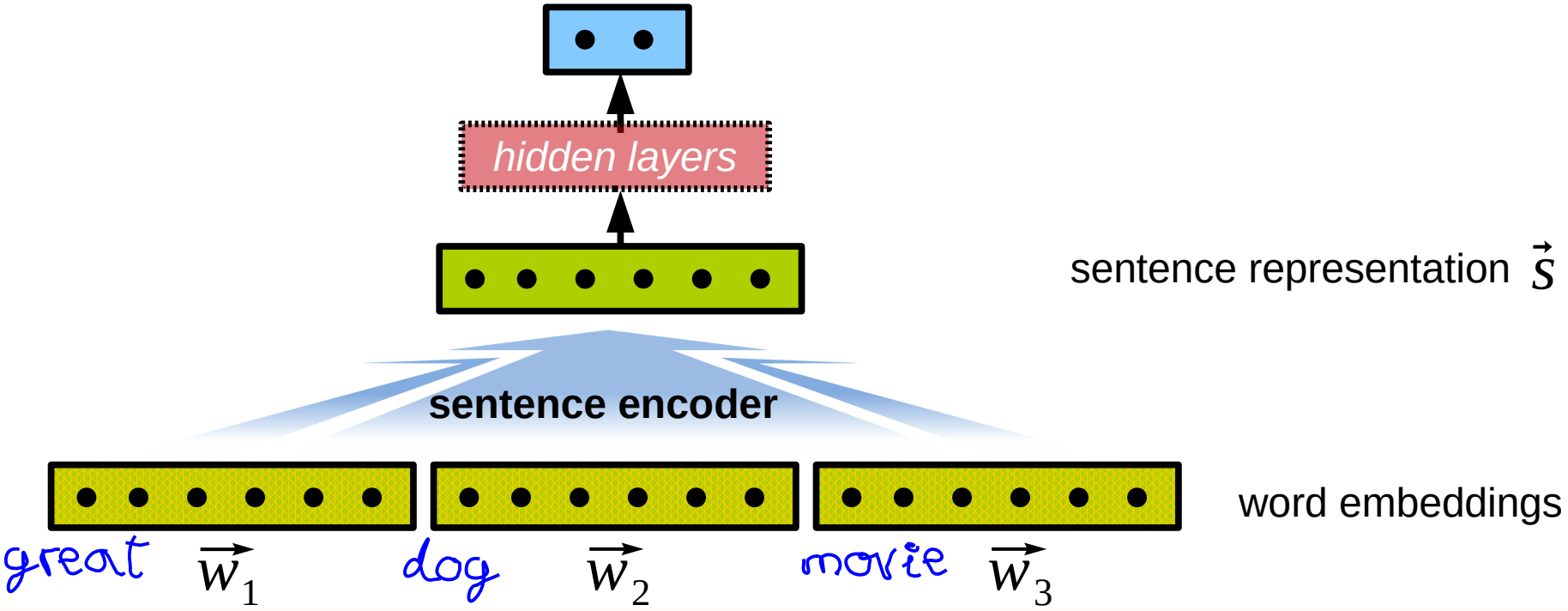


Sentence encoder

A function:
input a sequence of word embeddings
output a sentence representation

$\vec{w}_i \in \mathbb{R}^D$

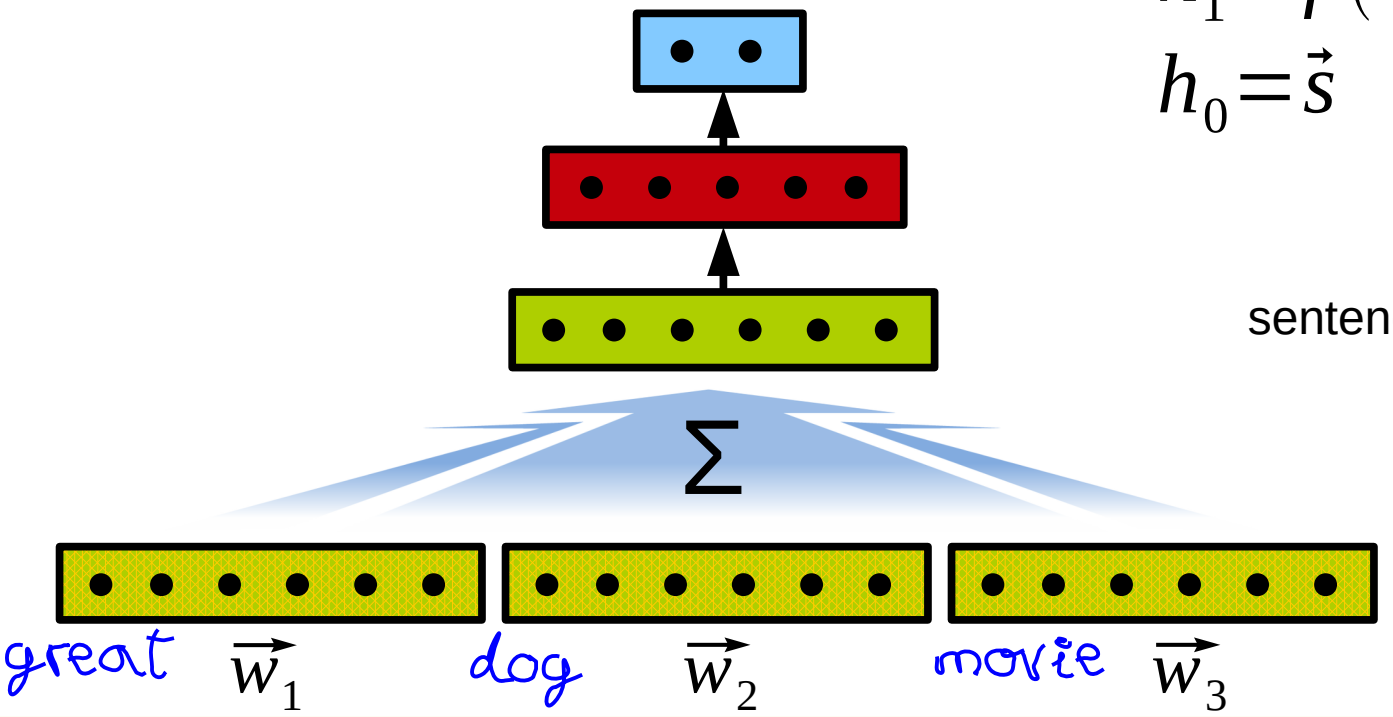
$\vec{s} \in \mathbb{R}^{D'}$



Sentence encoder

Baseline 1: Continuous bag of words

$$h_1 = f\left(W_1 h_0 + b_1\right)$$
$$h_0 = \vec{s}$$



sentence representation $\vec{s}$

word embeddings

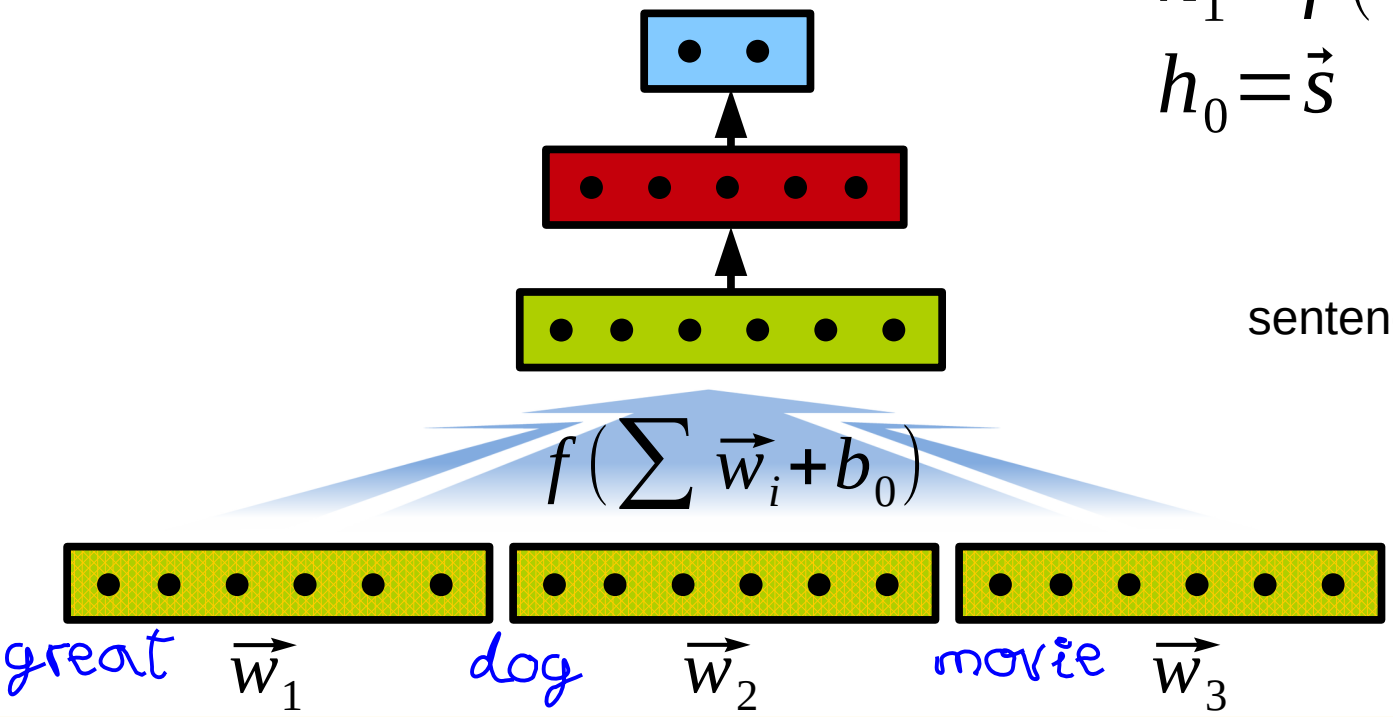
Sentence encoder

Baseline 1': MLP

It is implicitly encoding sequences as bag of words

$$h_1 = f(W_1 h_0 + b_1)$$

$$h_0 = \vec{s}$$



$f(\sum \vec{w}_i + b_0)$

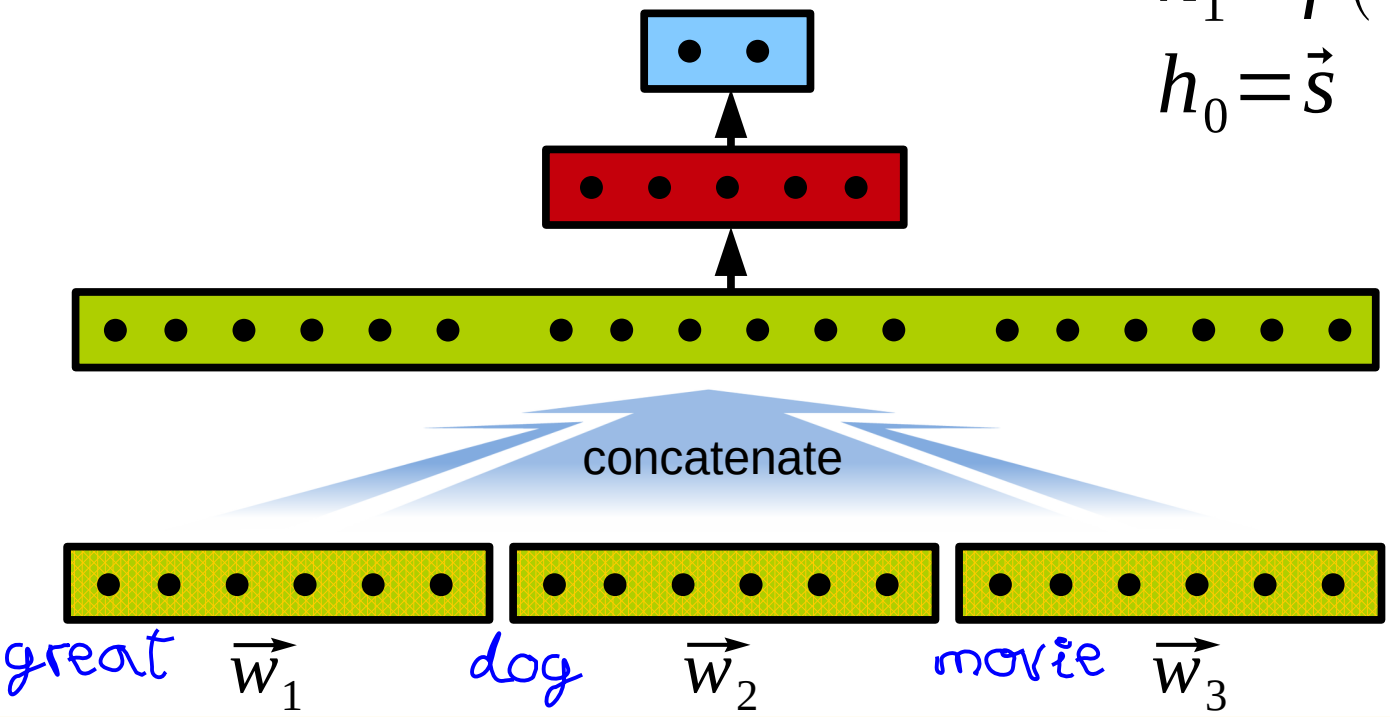
$\vec{w}_1$ $\vec{w}_2$ $\vec{w}_3$

sentence representation $\vec{s}$

Sentence encoder

Baseline 2: Add word order with concatenation

$$h_1 = f(W_1 h_0 + b_1)$$
$$h_0 = \vec{s}$$



Sentence encoder

Baseline 2: Add word order with concatenation

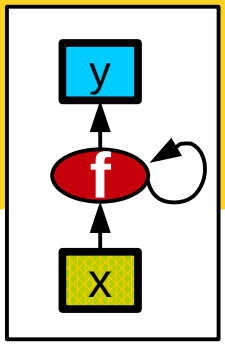
$$h_1 = f\left(W_1 h_0 + b_1\right)$$
$$h_0 = \vec{s}$$

It is a masterful performance but, for me, not enough to make me a fan of the film.

It is not only lovely to look at (the exquisite northern Italian countryside made me want to hop on a plane that moment), but is made even better by the subtle performance of Timothée Chalamet. "

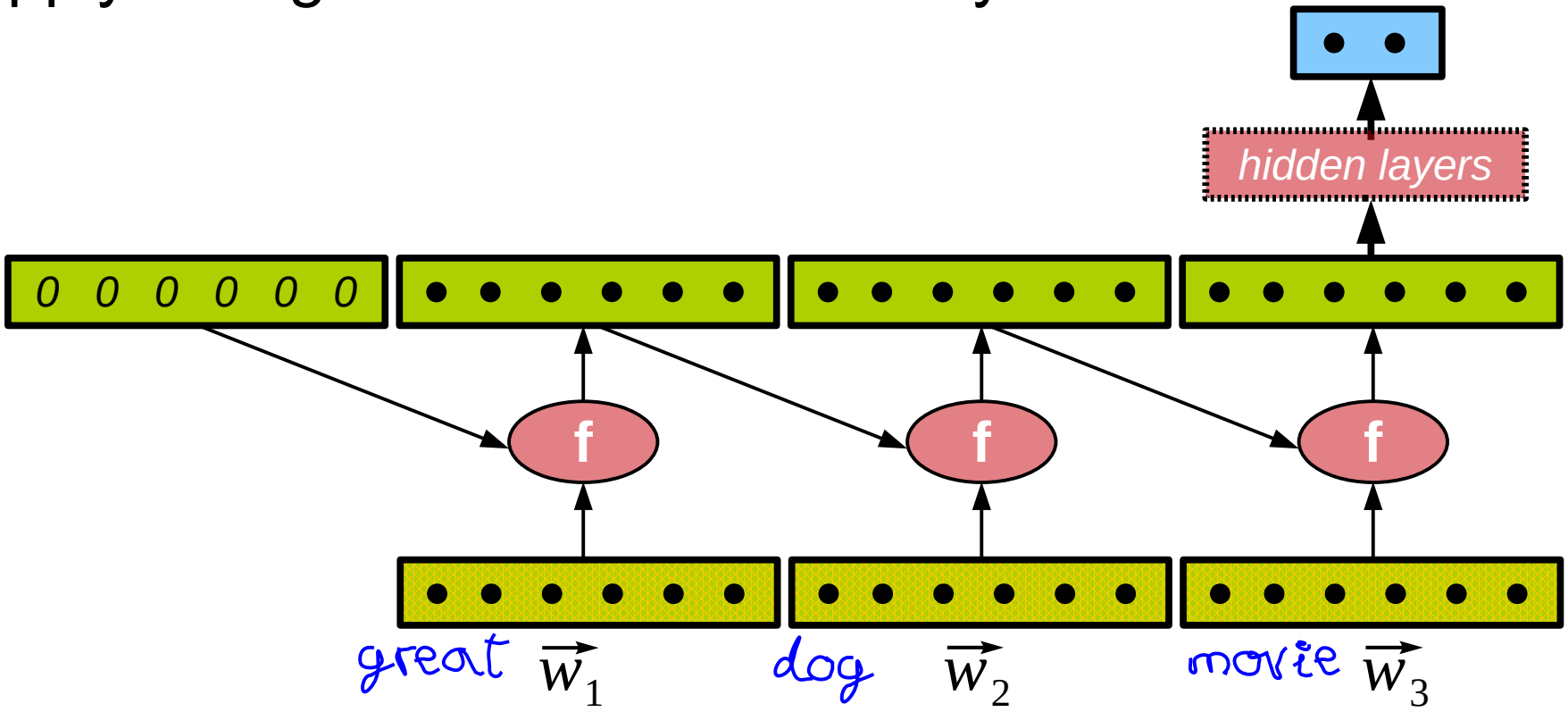
Fantastic movie

Sentence encoder

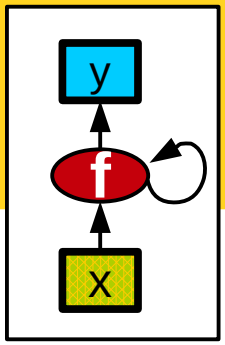


Recurrent Neural Network (RNN)

Apply a single function f recursively



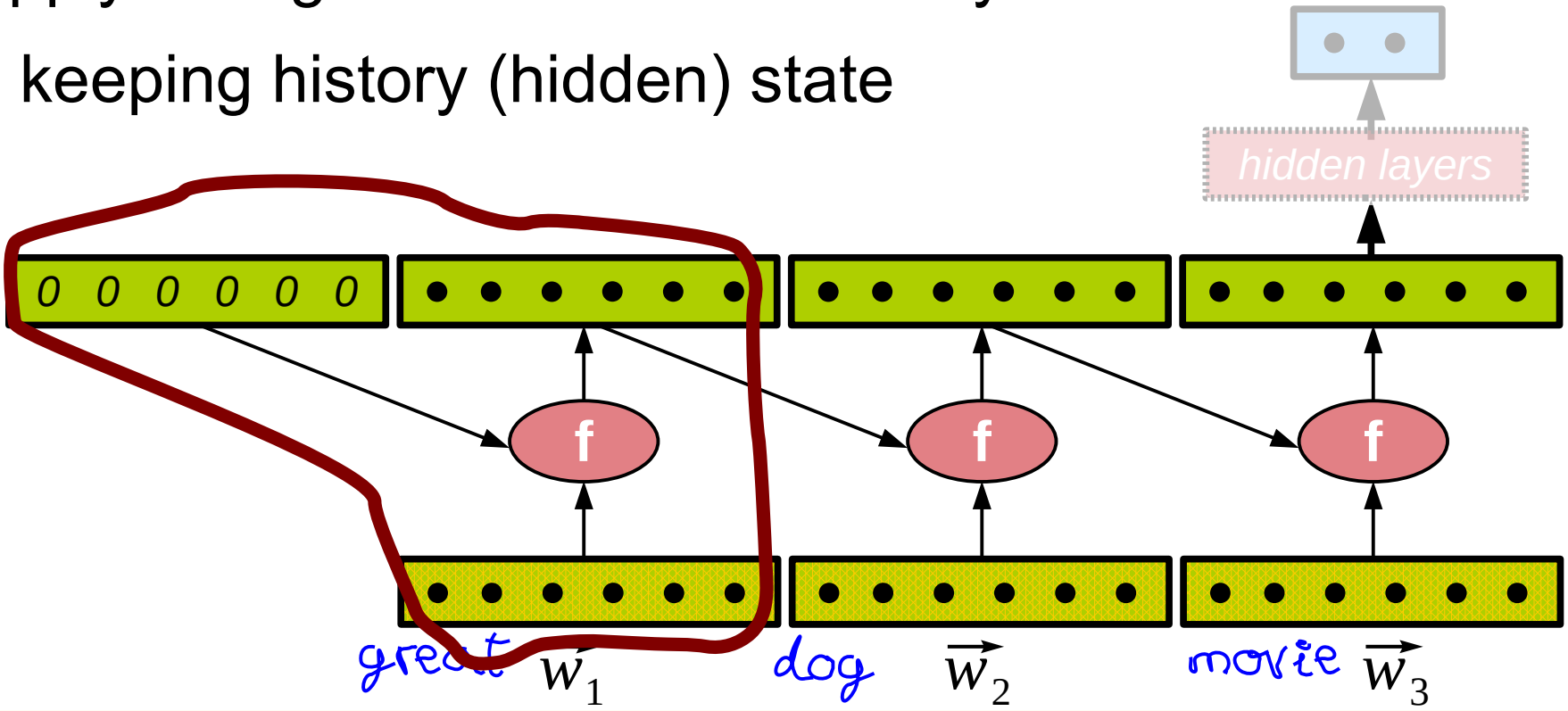
Sentence encoder: RNN



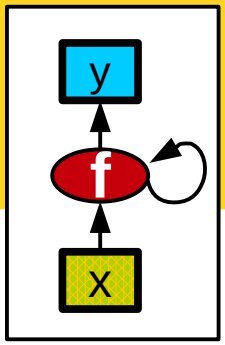
Recurrent Neural Network (RNN)

Apply a single function f recursively

keeping history (hidden) state



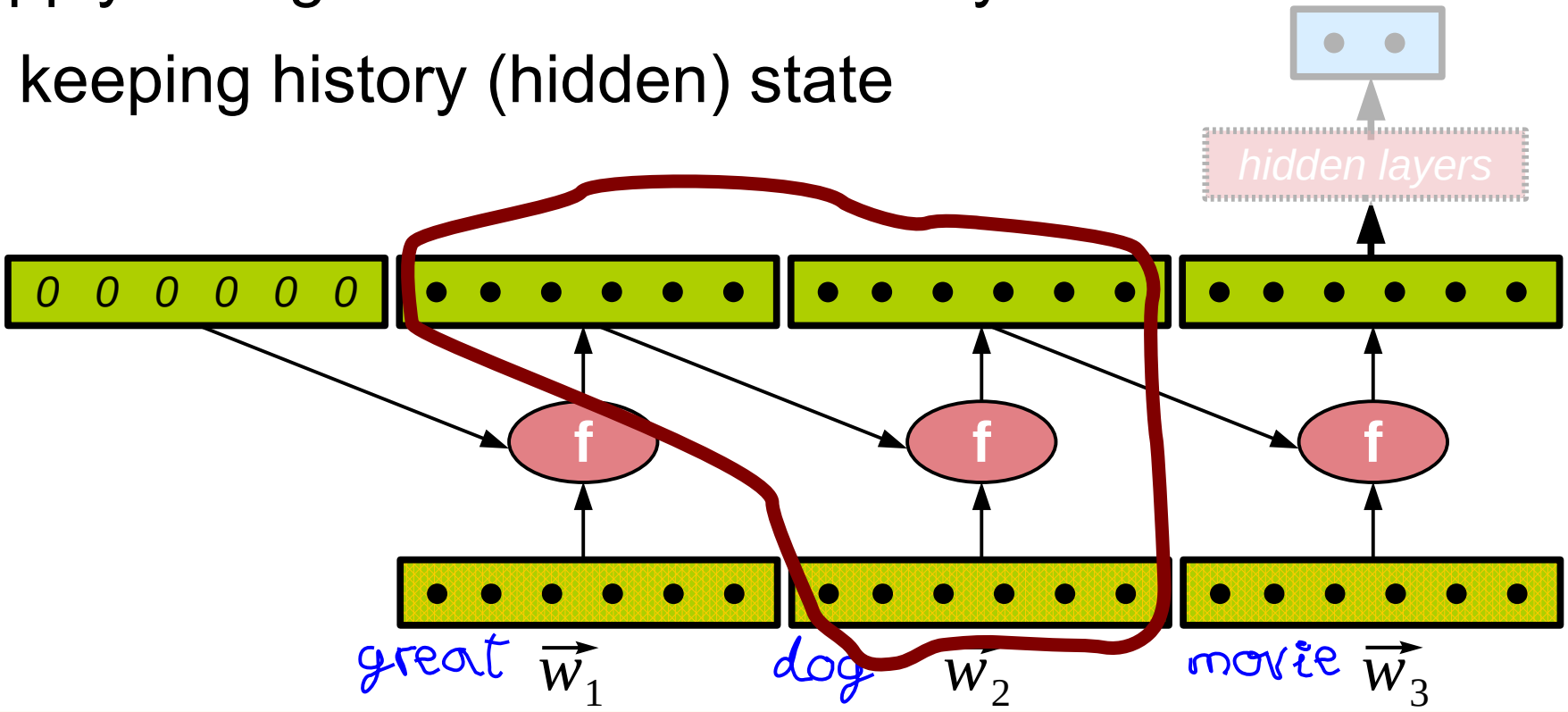
Sentence encoder: RNN



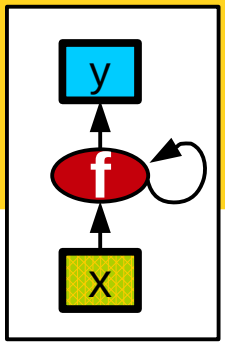
Recurrent Neural Network (RNN)

Apply a single function f recursively

keeping history (hidden) state



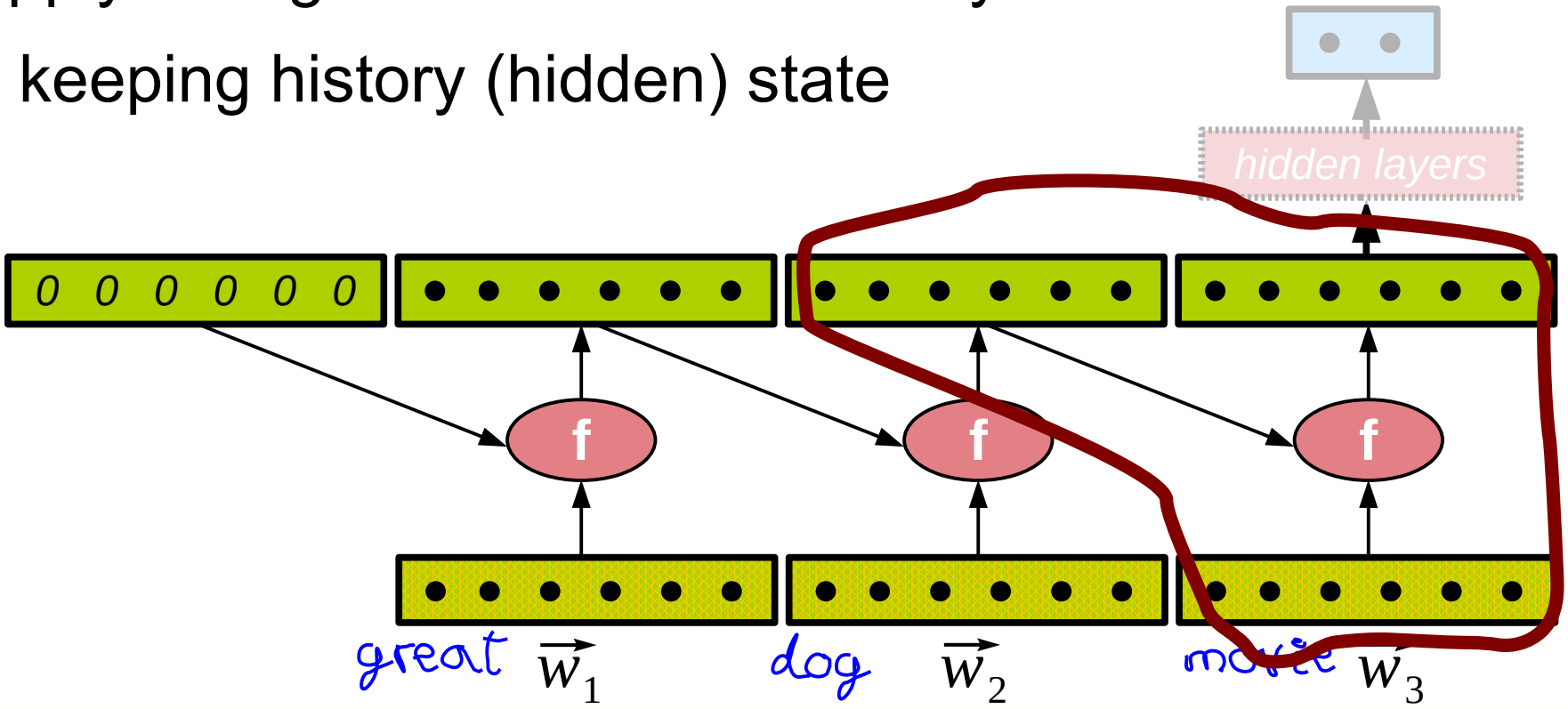
Sentence encoder: RNN



Recurrent Neural Network (RNN)

Apply a single function f recursively

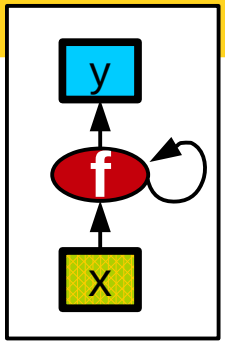
keeping history (hidden) state

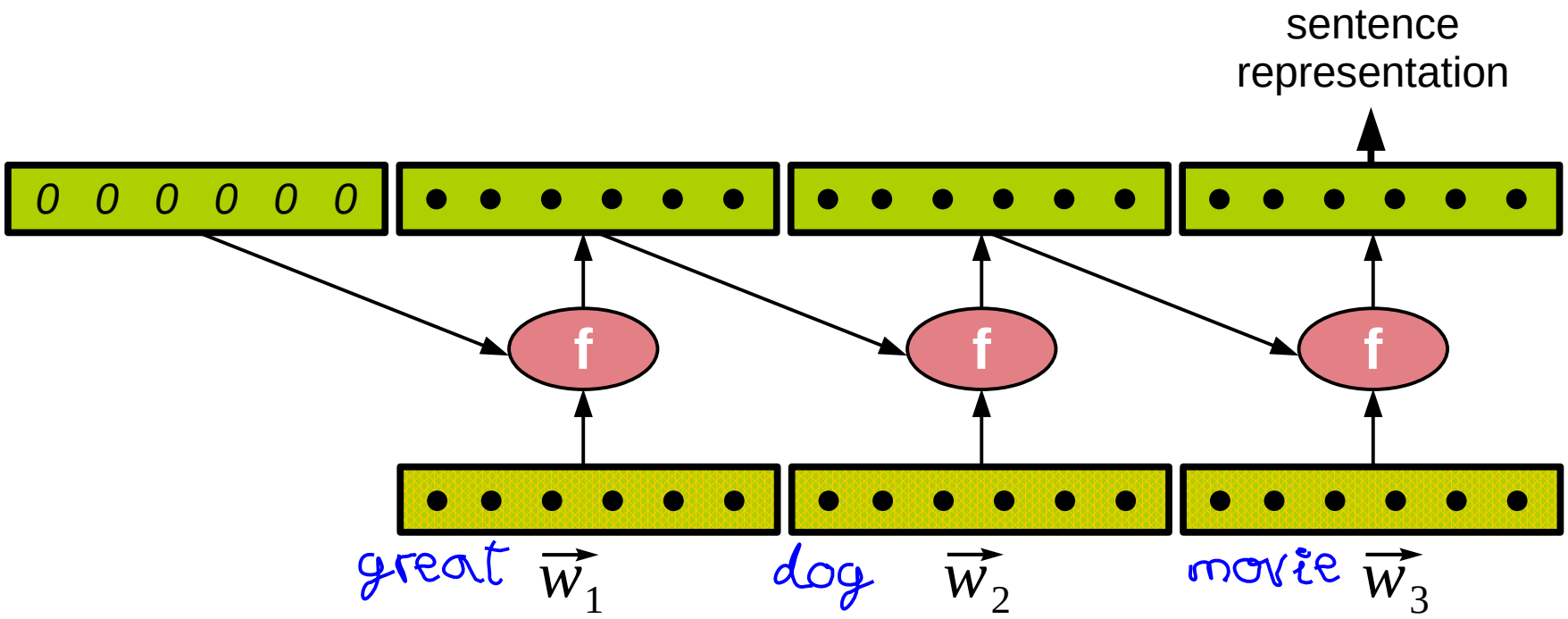


Sentence encoder: Basic RNN

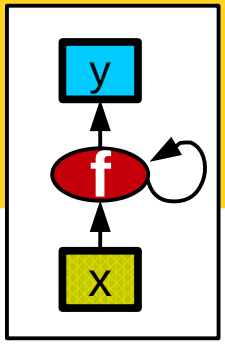
Apply a single function **f** recursively.

$$\vec{h}_t = f(\vec{h}_{t-1}, \vec{w}_t) = \tanh(W . [\vec{h}_{t-1}, \vec{w}_t] + \vec{b})$$

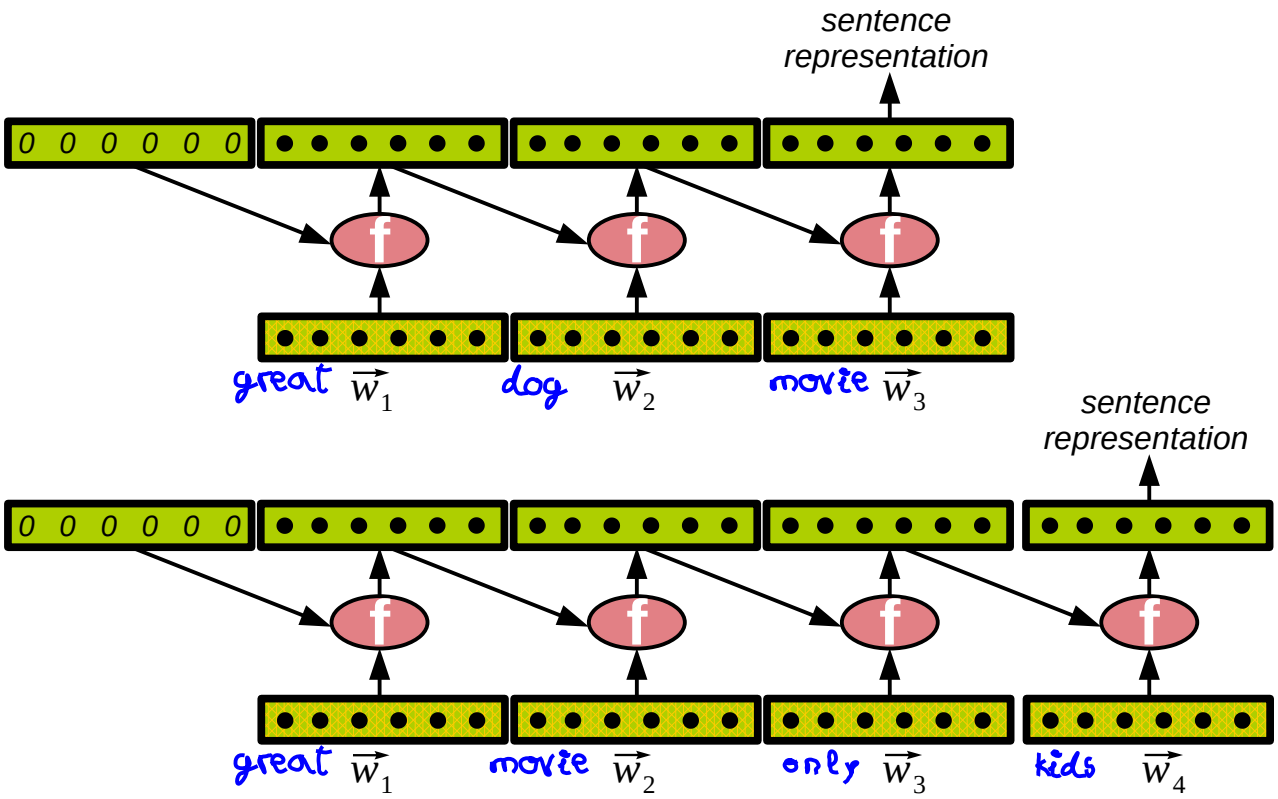




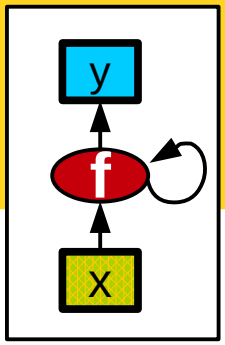
RNN: Unrolling



Varying sequence length:
unroll (different graph for each input)

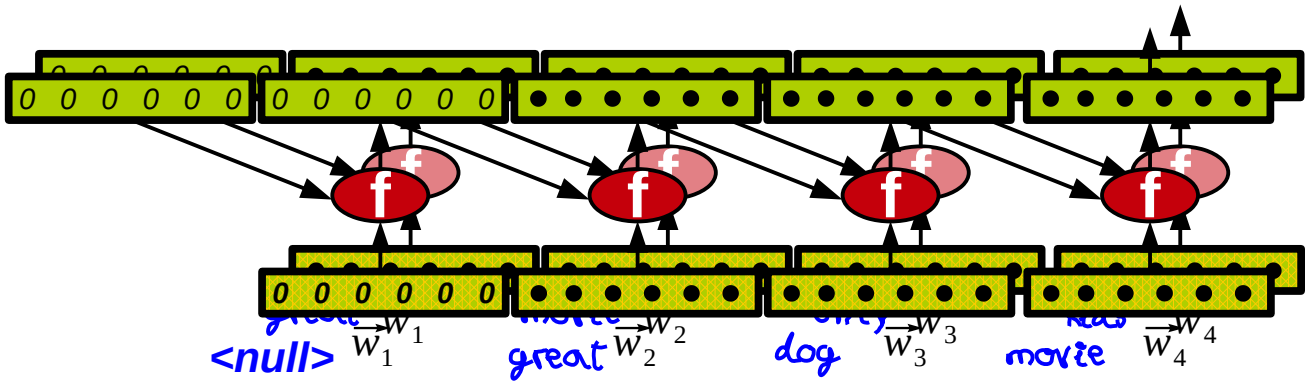


RNN: Batching

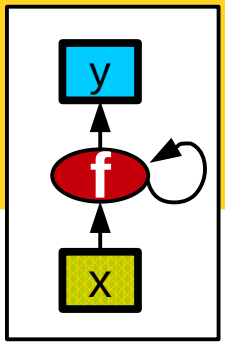


In order to do mini-batch:

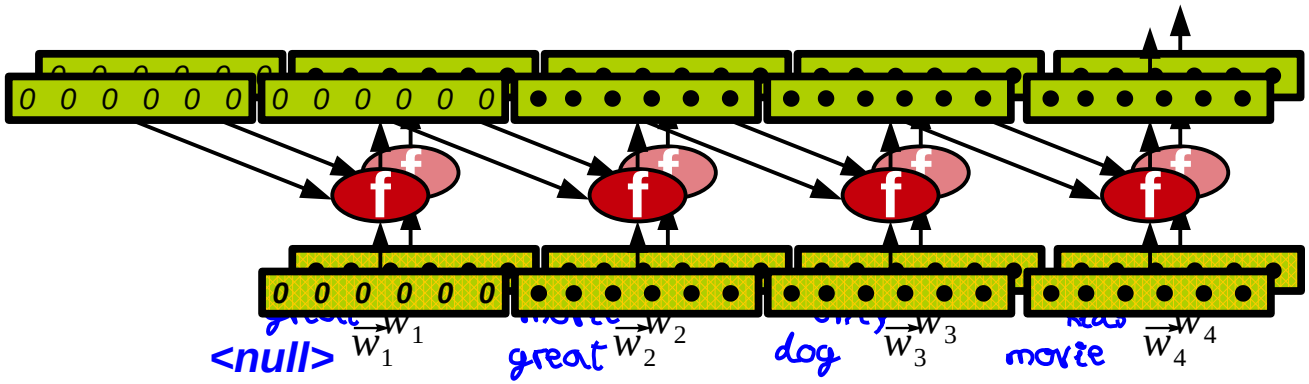
- Choose a single unrolling constant N
- Pad first words with zeros (shifting right)
- More sophisticated: shuffle examples by sentence length (set N to max. length in mini-batch)



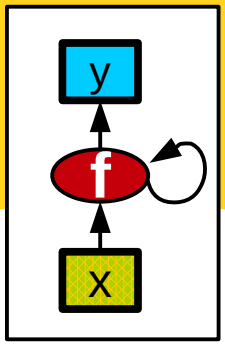
RNN: implementation



- Define the unrolled graph (with N cells)
- Weight sharing:
  - There is a single W, shared in the unrolled graph
  - Apply gradients repeatedly

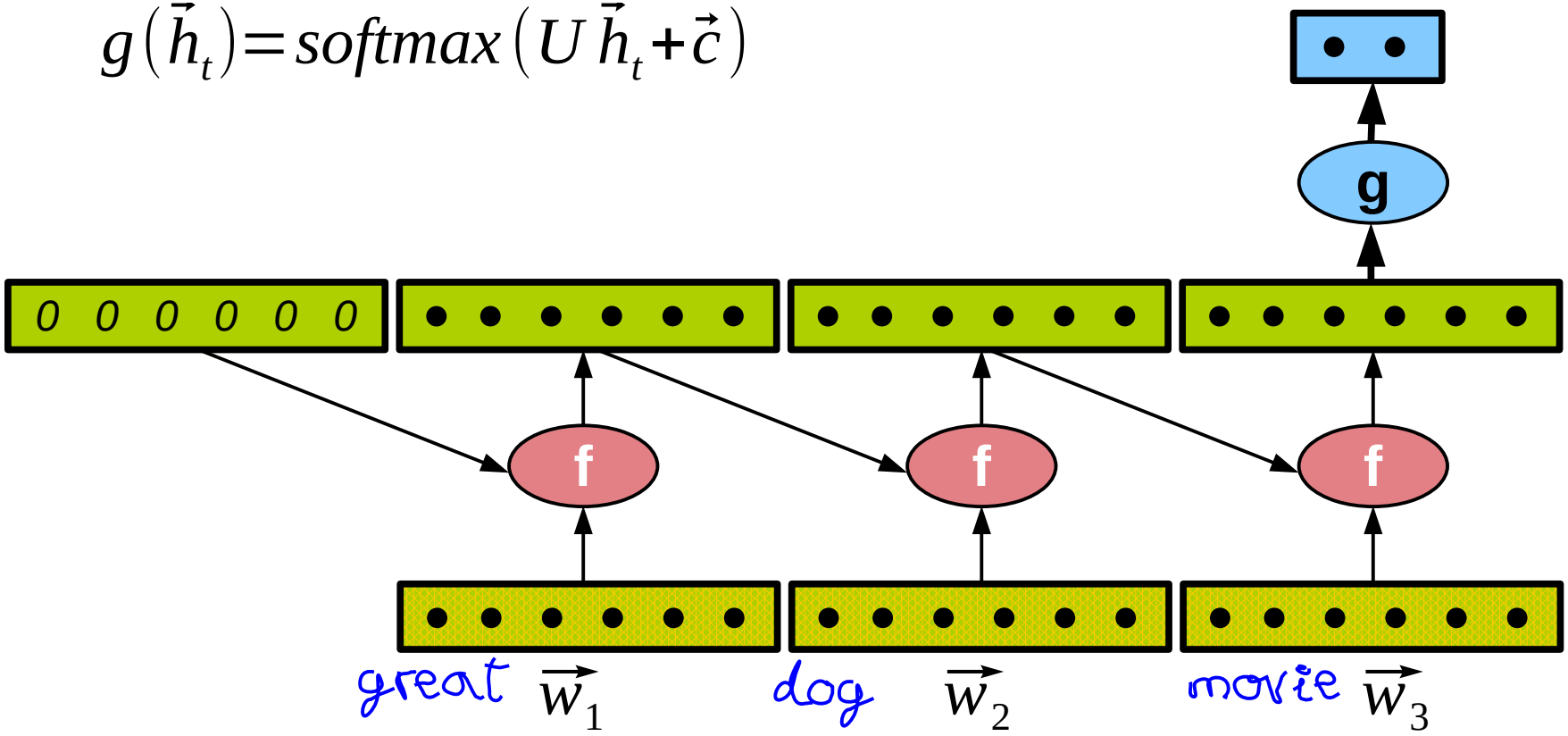


RNN: classifier

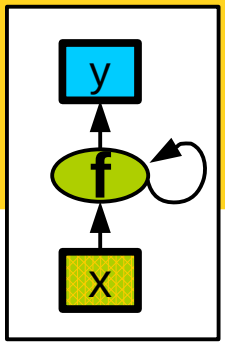


- Last hidden state is input to classifier

$$g(\vec{h}_t) = softmax(U\vec{h}_t + \vec{c})$$

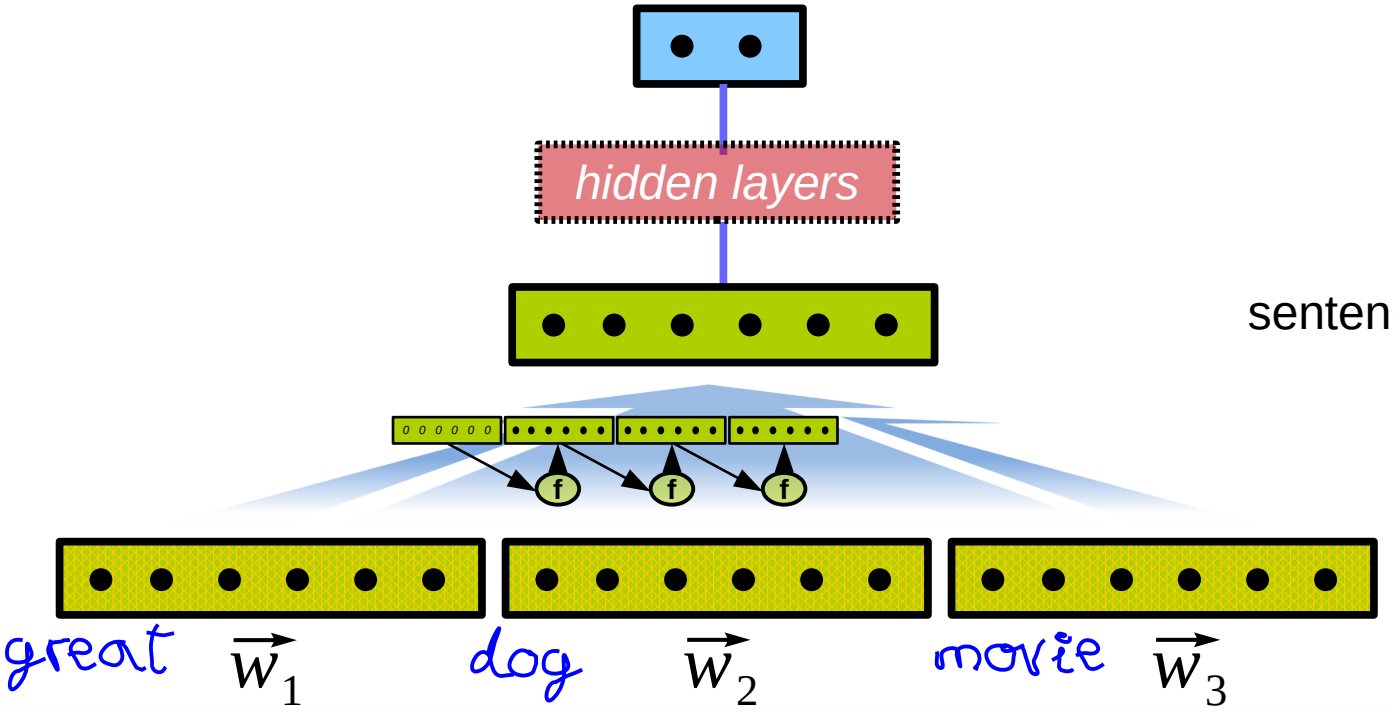


RNN as sentence encoder



Sentence encoder 3: RNN

$$\vec{h}_t = \tanh\left(W . [\vec{h}_{t-1}, \vec{w}_t] + \vec{b}\right)$$

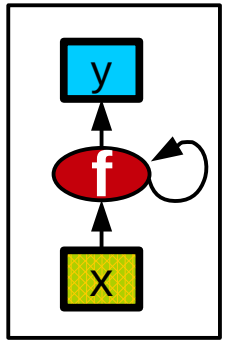


sentence representation $\vec{s}$

word embeddings $\vec{w}_1$ $\vec{w}_2$ $\vec{w}_3$

Embeddings and RNN in tf

- Optimized embedding access:
  - Vocabulary strings into integers (index)
  - OOV word
  - tf.nn.embedding_lookup
- Tensorflow has ready-to-use RNN
- In the lab, you will program the RNN from scratch using tensor operations



THANKS!

Acknowledgements:

- Overall slides: Sam Bowman (NYU), Chris Manning and Richard Socher (Stanford), Marek Rei and Ekaterina Kochmar (Cambridge), Mikel Artetxe and Gorka Labaka (UPV/EHU)
- All source url's listed in the slides.