Deep Learning for Natural Language Embeddings

2 May 2016
KTH

Roelof Pieters
@graphific

roelof@kth.se
www.csc.kth.se/~roelof/

ROYAL INSTITUTE OF TECHNOLOGY

AE ARTIFICIAL EXPERIENCE

GRAPH TECHNOLOGIES

Can we understand Language ?

Some of the challenges in Language Understanding:

1. Language is ambiguous:
   Every sentence has many possible interpretations.

2. Language is productive:
   We will always encounter new words or new constructions

3. Language is culturally specific

Can we understand Language ?

Some of the challenges in Language Understanding:

1. **Language is ambiguous:**
   Every sentence has many possible interpretations.

2. Language is productive:
   We will always encounter new words or new constructions

- **fruit flies like a banana**

  NN NN VB DT NN

  NN VB P DT NN

  NN NN P DT NN

  NN VB VB DT NN

- **the students went to class**

  DT NN VB P NN

- **plays well with others**

  VB ADV P NN

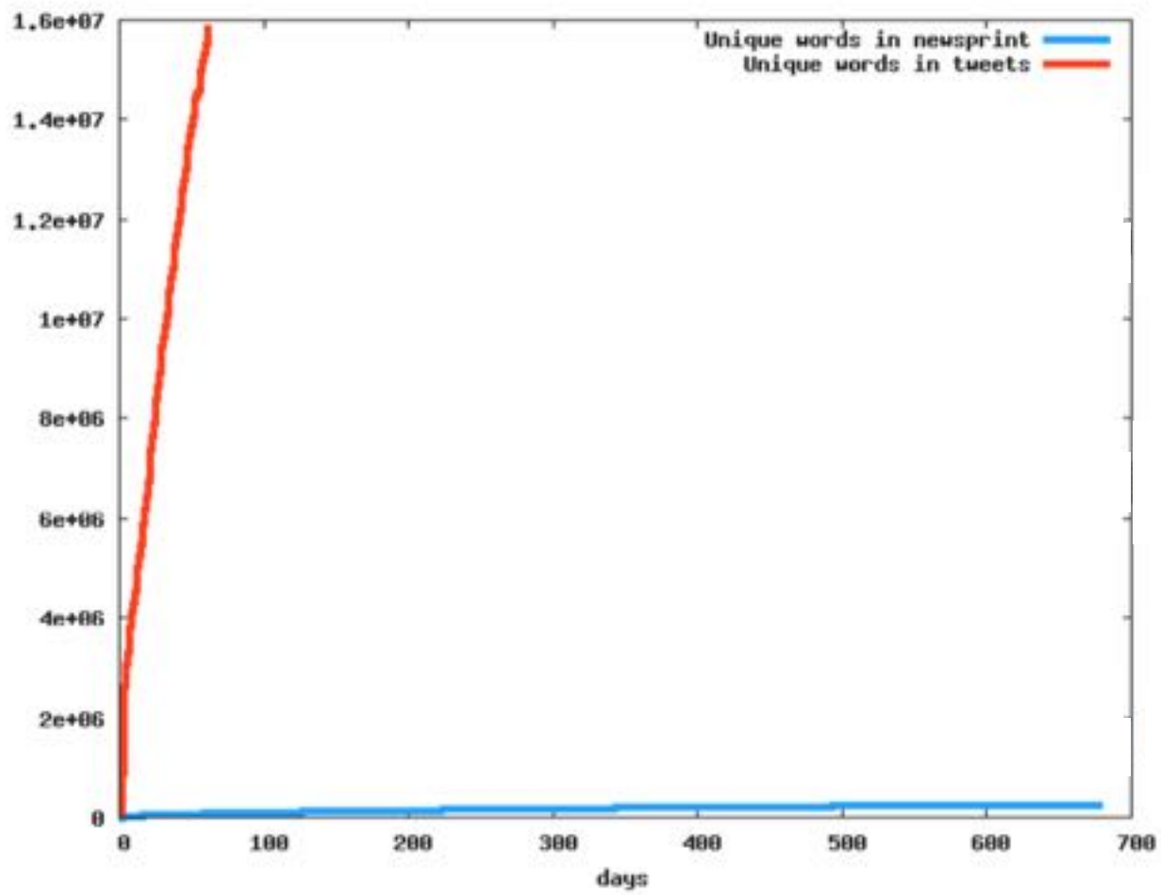
  NN NN P DT

Can we understand Language ?

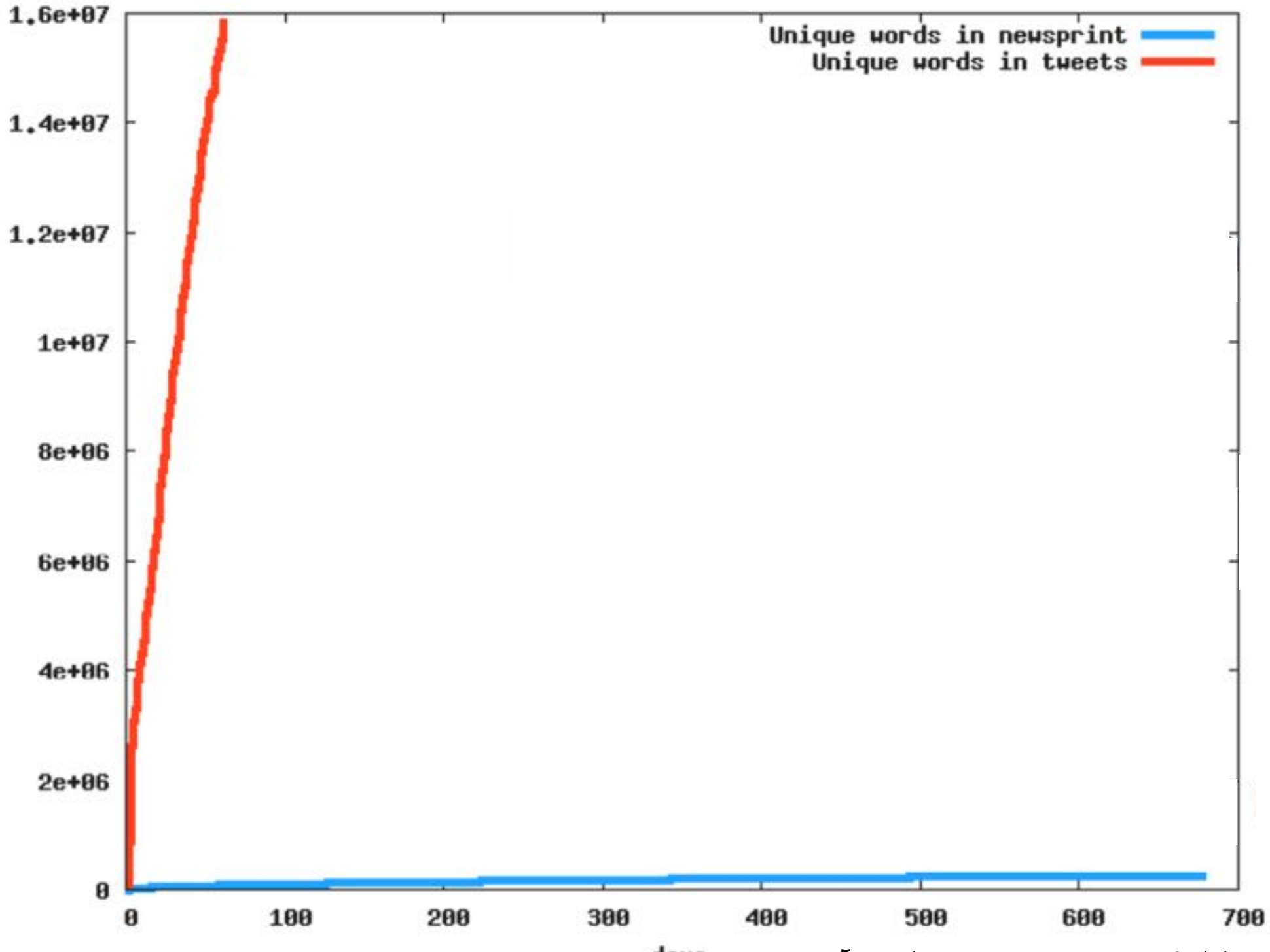
Some of the challenges in Language Understanding:

1. Language is ambiguous:
   Every sentence has many possible interpretations.

2. **Language is productive**:
   We will always encounter new words or new constructions



Digital Media Deluge: text



[Karlgren 2014, NLP Sthlm Meetup]

Digital Media Deluge: text

twitter

lol ?

loly	hahah	. . .
lolz	hahaha	
lole	hahahaha	
lola	hahahah	
loll	haha	
loli	hehehe	
lolo	hahahahaha	
lols	ummm	
lolu	o.o	
loil	hehe	
loel	xdd	
locl	heheh	
loal	uhhh	
loul	loool	
lool	lool	
l'ol	ahah	
luol	ahaha	
#lol	looool	
lo	lolz	
lolan	ahahah	
lol'd	ikr	
lolas	just kidding	
lolis	xddd	
lolab	ummmm	
	xd	
	:p	

[http://lexicon.gavagai.se/lookup/en/lol]

Can we understand Language ?

Some of the challenges in Language Understanding:

1. Language is ambiguous:
   Every sentence has many possible interpretations.

2. Language is productive:
   We will always encounter new words or new constructions

3. **Language is culturally specific**

Can we understand Language ?

NLP Data?

Name	Language	Size	Availability	Comments
Penn Treebank	US English	2 million + words	Available (distributed by LDC)	1 million WSJ, 1 million speech, surface syntax (1970s TG)
BLLIP WSJ corpus	US English	30 million words	Available (distributed by LDC)	WSJ newswire. Automatically parsed, not hand checked. Same structure as Penn Treebank, except for some additional coreference marking
ICE-GB	UK English	1 million words (83,394 sentences)	Available; c. 500 pounds	British part of ICE, the International Corpus of English project. Tagged and parsed for function. Half spoken material.
Bulgarian Treebank	Bulgarian	n/a	POS-tagged texts and dependencies analyses are available (some are free on the web, others via a license agreement)	An under construction Bulgarian HPSG treebank.
Penn Chinese Treebank	Chinese	100,000 words	Available (LDC)	Based on Xinhua news articles. 1980s-style GB syntax.
The Prague Dependency Treebank 1.0	Czech	500,000 words	Free on completion of license agreement (available through LDC).	Analyzed at the levels of parts of speech, syntactic functions (and, in the future, semantic roles) level in a dependency framework. Text from newspapers and weekly magazines.
Danish Dependency Treebank 1.0	Danish	100,000 words	Available free under the GPL.	Built on a portion of the Parole corpus.
Alpino Dependency Treebank	Dutch	150,000 words	Freely downloadable	Assorted subcorpora. By far the largest is the full cdbl (newspaper) part of the Eindhoven corpus.
NEGRA Corpus	German	20,000 sentences	Available free of charge to academics on completion of license agreement.	Saarland University Syntactically Annotated Corpus of German Newspaper Texts. Tagged, and with syntactic structures.
TIGER corpus	German	700,000 words	Available free of charge for research purposes on completion of license agreement.	German newspaper text (Frankfurter Rundschau). Semi-automatically parsed. They also have a good treebank search tool, TIGERSearch.
Icelandic Parsed Historical Corpus (IcePaHC)	Icelandic	1,000,000 words	Free download (LGPL)	Texts from 1150 through 2008!
TUT: Turin University Treebank	Italian	2,400 sentences	Free download.	Morhpological analysis and dependency analysis. Penn Treebank translation. Civil law and newspaper texts.
Floresta Sintá(c)tica	Portuguese	168,000 words hand-corrected; 1,000,000 words automatically parsed	Hand corrected part is free web download; automatically parsed part available through email contact	Text from CETEMPúblico corpus. Phrase structure and dependency representations. Available in several formats, including Penn Treebank format.
Talbanken05	Swedish	300,000 words	Free download	Resurrects and modernizes an early treebank from the 1970s.

some of the (many) treebank datasets

source: http://www-nlp.stanford.edu/links/statnlp.html#Treebanks

Penn Treebank

That's a lot of "manual" work:

```
( (S (`` ``)
    (S-TPC-2
      (NP-SBJ-1 (PRP We) )
      (VP (MD would)
        (VP (VB have)
          (S
            (NP-SBJ (-NONE- *-1) )
            (VP (TO to)
              (VP (VB wait)
                (SBAR-TMP (IN until)
                  (S
                    (NP-SBJ (PRP we) )
                    (VP (VBP have)
                      (VP (VBN collected)
                        (PP-CLR (IN on)
                          (NP (DT those)(NNS assets)))))))))))))
(, ,) (`` ``)
(NP-SBJ (PRP he) )
(VP (VBD said)
  (S (-NONE- *T*-2) ))
(. .) ))
```

Penn Treebank

With a lot of issues:

- **the students went to class**

  DT NN VB P NN

- **plays well with others**

  VB ADV P NN

  NN NN P DT

- **fruit flies like a banana**

  NN NN VB DT NN

  NN VB P DT NN

  NN NN P DT NN

  NN VB VB DT NN

Deep Learning: Why for NLP ?

Beat state of the art at:

- Language Modeling (Mikolov et al. 2011) [WSJ AR task]
- Speech Recognition (Dahl et al. 2012, Seide et al 2011; following Mohammed et al. 2011)
- Sentiment Classification (Socher et al. 2011)

and we already know for some longer time it works well for Computer Vision of course:

- MNIST hand-written digit recognition (Ciresan et al. 2010)
- Image Recognition (Krizhevsky et al. 2012) [ImageNet]

Deep Learning: Why for NLP ?

One Model rules them all ?

DL approaches have been successfully applied to:

- Automatic summarization
- Coreference resolution
- Discourse analysis
- Machine translation
- Morphological segmentation
- Named entity recognition (NER)
- Natural language generation
- Word sense disambiguation
- Relationship extraction
- Speech processing
- Part-of-speech tagging
- sentence boundary disambiguation
- Sentiment analysis
- Optical character recognition (OCR)
- Question answering
- Parsing
- Word segmentation
- Natural language understanding
- Information retrieval (IR)
- Speech recognition
- Topic segmentation and recognition
- Speech segmentation
- Information extraction (IE)

Semantics: Meaning

- What is the meaning of a word?
  (Lexical semantics)

- What is the meaning of a sentence?
  ([Compositional] semantics)

- What is the meaning of a longer piece of text?
  (Discourse semantics)

Language Model

- Language models define probability distributions over (natural language) strings or sentences

- Joint and Conditional Probability

$$P(X|Y) = \frac{P(X,Y)}{P(Y)}$$

Language Model

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, 'and what is the use of a book,' thought Alice 'without pictures or conversation?'

P(of) = 3/66
P(Alice) = 2/66
P(was) = 2/66
P(to) = 2/66
P(her) = 2/66
P(sister) = 2/66
P(,) = 4/66
P(') = 4/66

Language Model

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, 'and what is the use of a book,' thought Alice 'without pictures or conversation?'

$P(w_{i+1}=\text{of} \mid w_i=\text{tired}) = 1$
$P(w_{i+1}=\text{of} \mid w_i=\text{use}) = 1$
$P(w_{i+1}=\text{sister} \mid w_i=\text{her}) = 1$
$P(w_{i+1}=\text{beginning} \mid w_i=\text{was}) = 1/2$
$P(w_{i+1}=\text{reading} \mid w_i=\text{was}) = 1/2$

$P(w_{i+1}=\text{bank} \mid w_i=\text{the}) = 1/3$
$P(w_{i+1}=\text{book} \mid w_i=\text{the}) = 1/3$
$P(w_{i+1}=\text{use} \mid w_i=\text{the}) = 1/3$

$$P(X|Y) = \frac{P(X,Y)}{P(Y)}$$

Word senses

What is the meaning of words?

- Most words have many different senses:
  dog = animal or sausage?

How are the meanings of different words related?

- - Specific relations between senses:
  Animal is more general than *dog*.

- - Semantic fields:
  money is related to *bank*

Word senses

Polysemy:

- A lexeme is polysemous if it has different related senses

- bank = financial institution or building

Homonyms:

- Two lexemes are homonyms if their senses are unrelated, but they happen to have the same spelling and pronunciation

- bank = (financial) bank or (river) bank

Word senses: relations

Symmetric relations:

- Synonyms: couch/sofa
  Two lemmas with the same sense

- Antonyms: cold/hot, rise/fall, in/out
  Two lemmas with the opposite sense

Hierarchical relations:

- Hypernyms and hyponyms: pet/dog
  The hyponym (***dog***) is more specific than the hypernym (***pet***)

- Holonyms and meronyms: car/wheel
  The meronym (***wheel***) is a part of the holonym (***car***)

Distributional representations

"You shall know a word by the company it keeps"
(J. R. Firth 1957)

One of the most successful ideas of modern statistical NLP!

Hundreds scour Clark Fork **River** **banks** in annual cleanup
The Missoulian - 22 hours ago

The Value of Digital Experience + Oracle **Banks** on ABM
CMSWire - 4 hours ago

Bitcoin and **Banks** Can Work Together, Interview With Priv…
CoinTelegraph - 8 hours ago

these words represent banking

Distributional hypothesis

He filled the wampimuk, passed it around and we all drunk some

We found a little, hairy wampimuk sleeping behind the tree

(McDonald & Ramscar 2001)

Word Representation

- NLP treats words mainly (rule-based/statistical approaches at least) as atomic symbols:

Love Candy Store

- or in vector space:

[o o o o o **1** o o o o o o o o o o o o ...]

- also known as “one hot” representation.

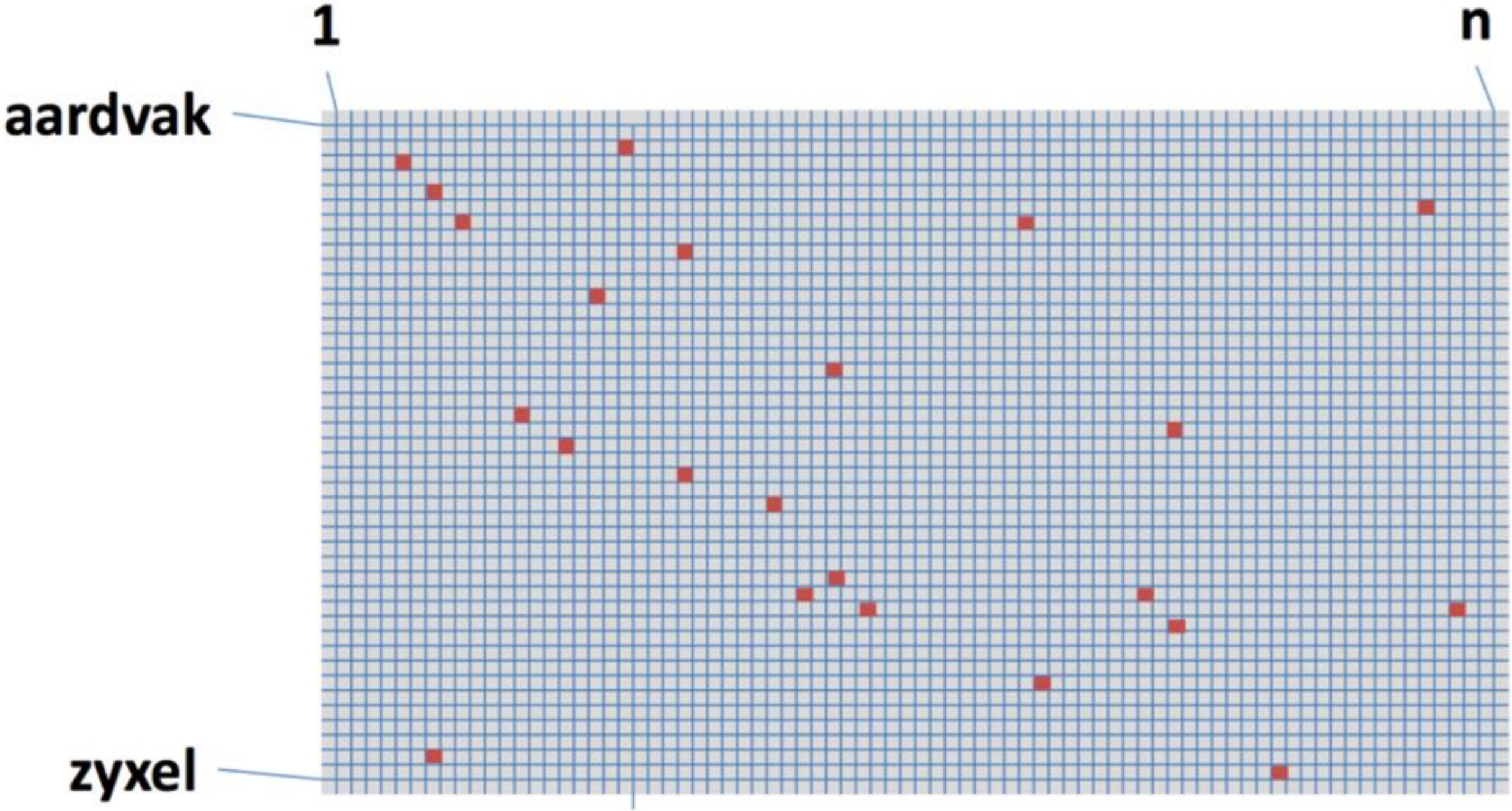
- Its problem ?

Candy [o o o o o **1** o o o o o o o o o o o o ...] AND

Store [o o o o o o o o o o o o o o o o **1** o ...] = **0** !

Word Representation

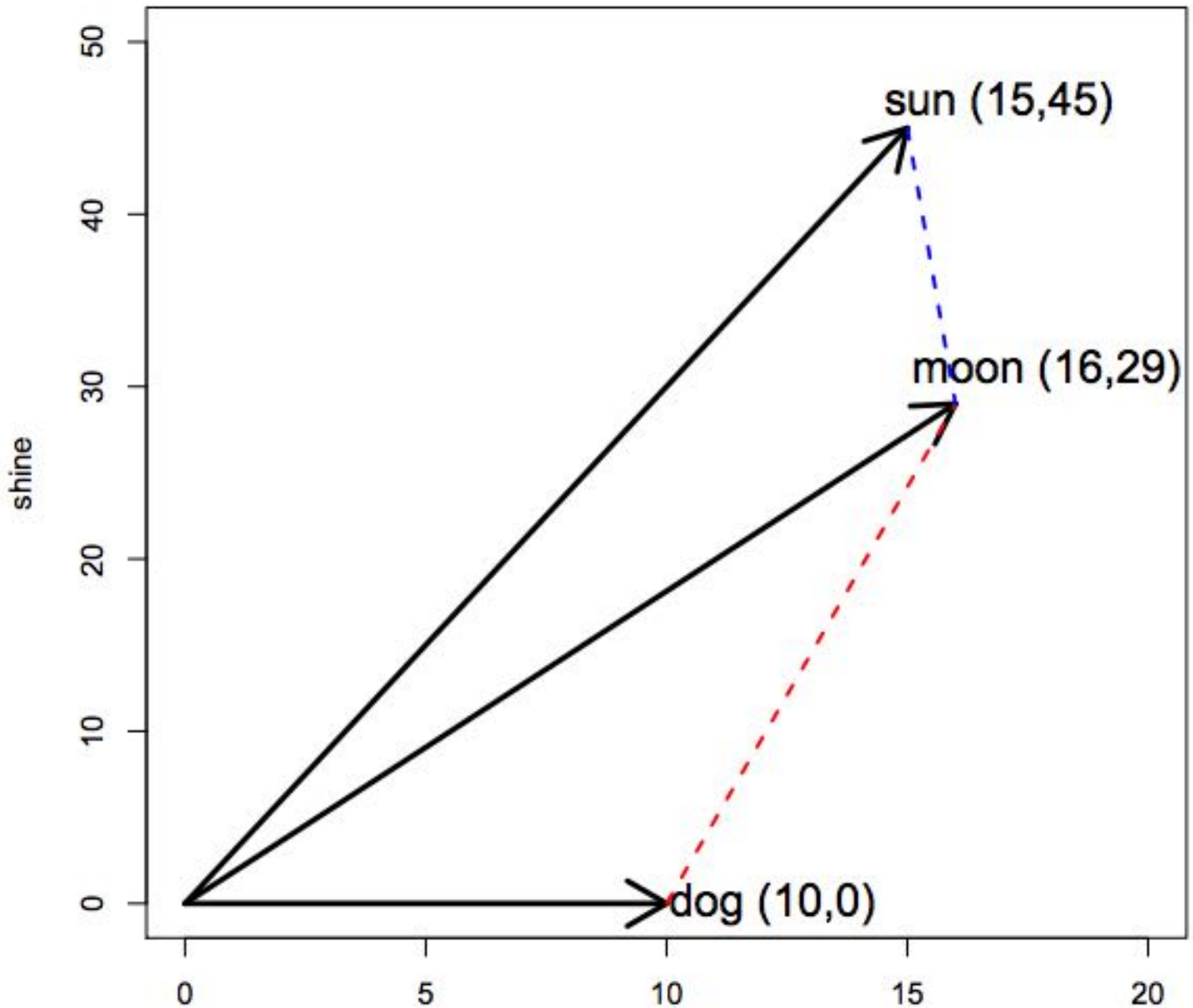
Term-document matrix = Sparse!



Distributional semantics

Distributional meaning as co-occurrence vector:

	planet	night	full
moon	10	22	43
sun	14	10	4
dog	0	4	2



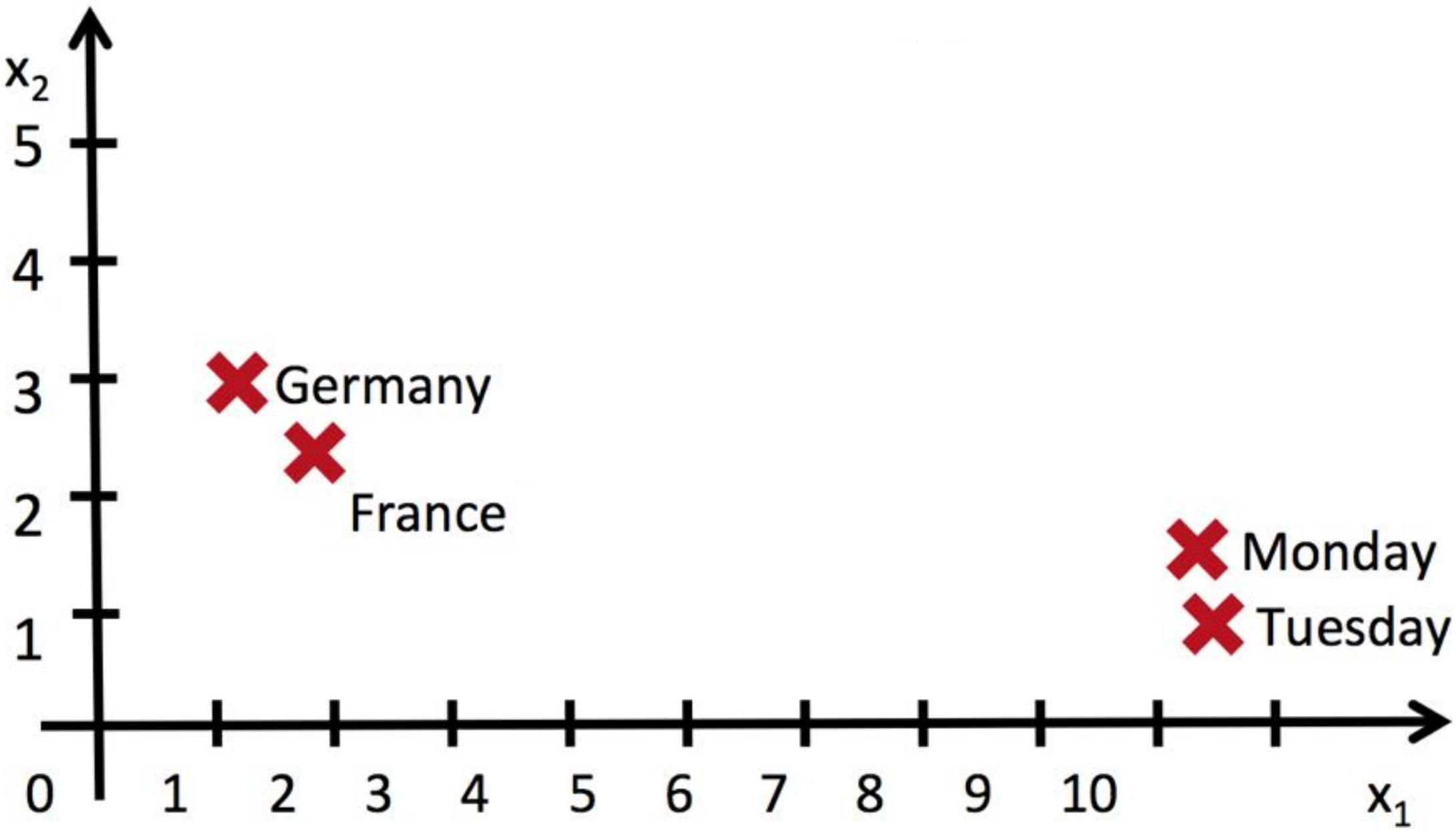
Deep Distributional representations

- Taking it further:
  - Continuous word embeddings
  - Combine vector space semantics with the prediction of probabilistic models
  - Words are represented as a **dense** vector:

Candy = [0.286, 0.792, −0.177, −0.107, 0.109, −0.542, 0.349, 0.271]

Vector Space Model

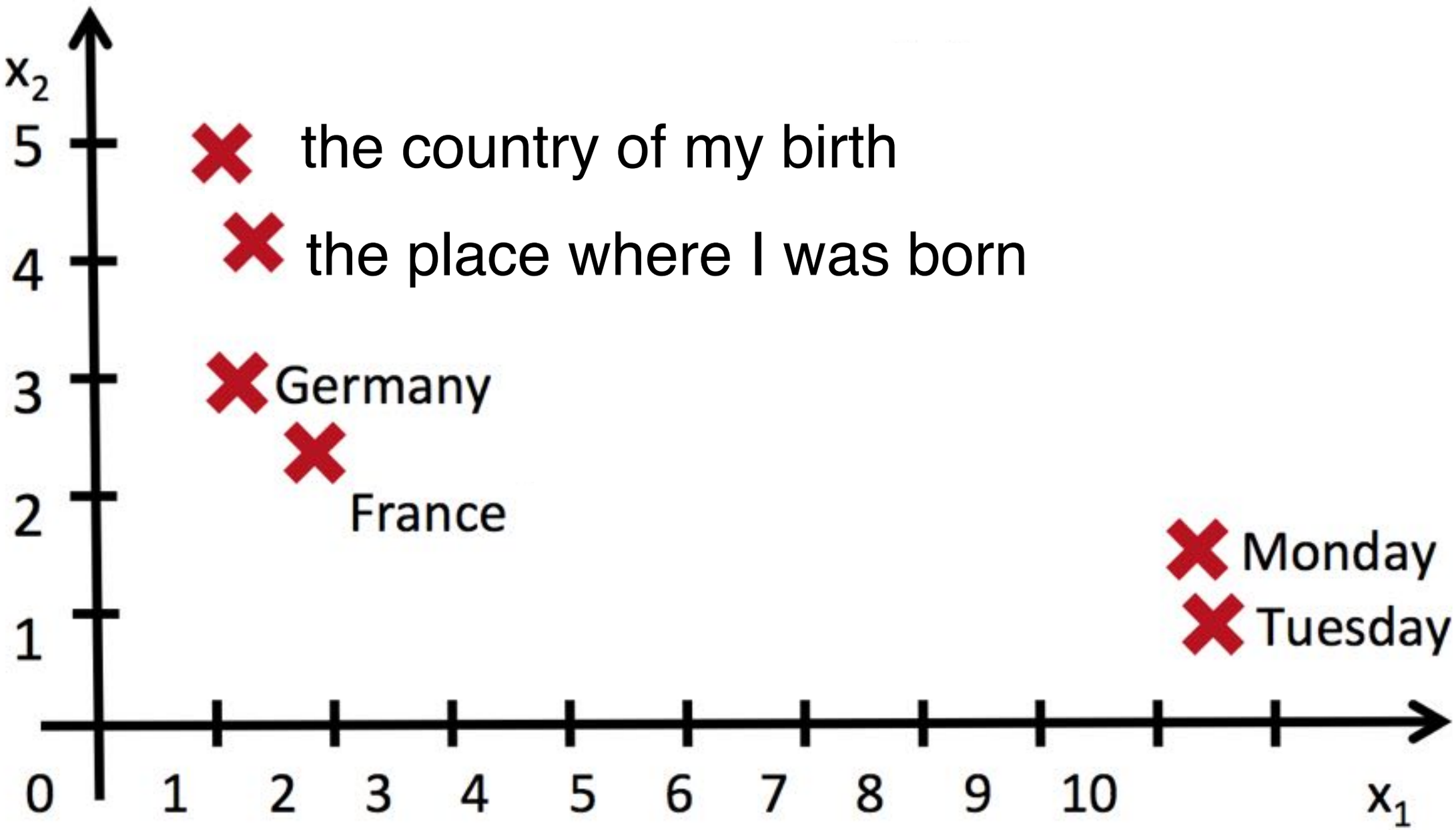
In a perfect world:



adapted rom Bengio, “Representation Learning and Deep Learning”, July, 2012, UCLA

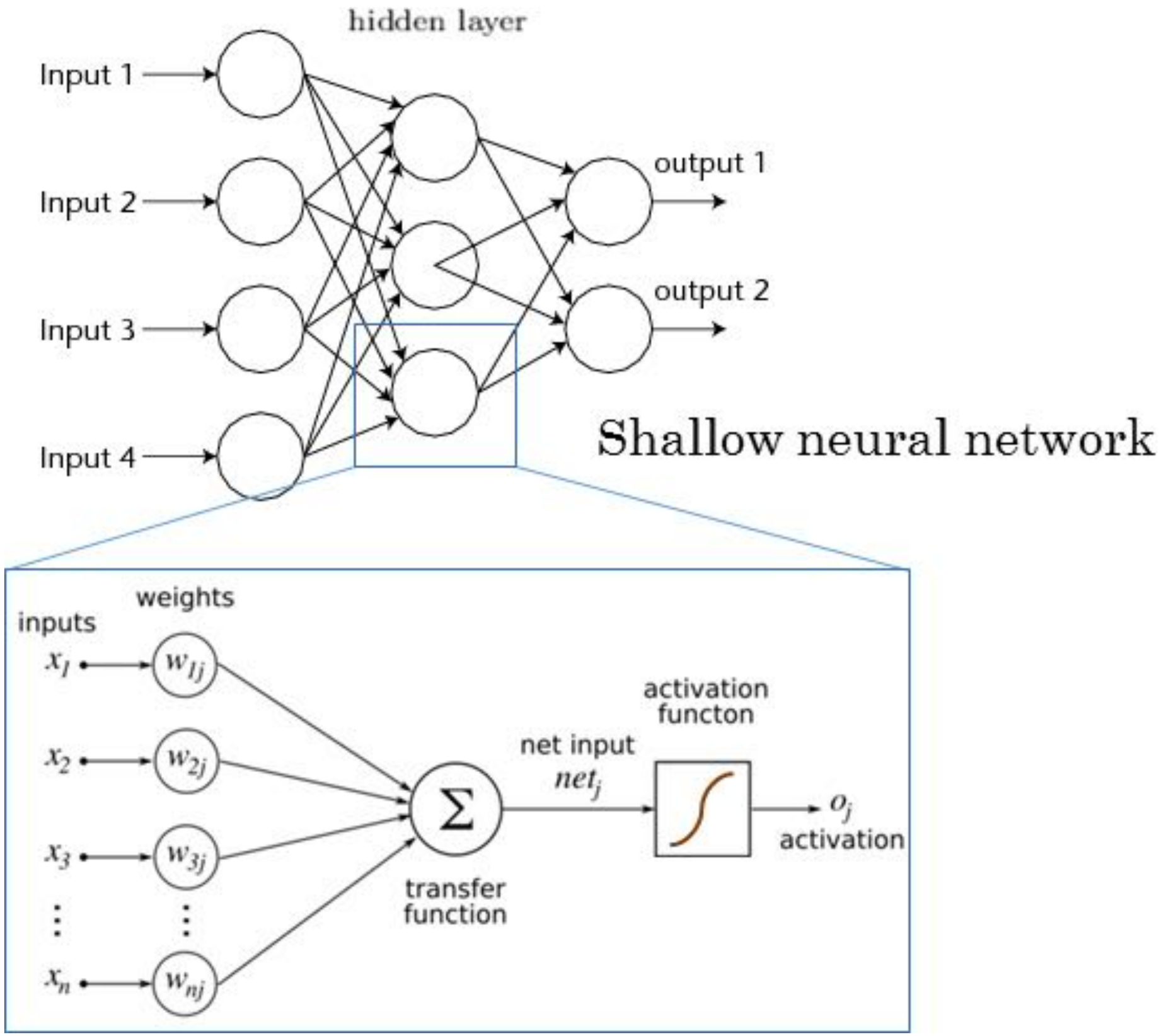
Vector Space Model

In a perfect world:



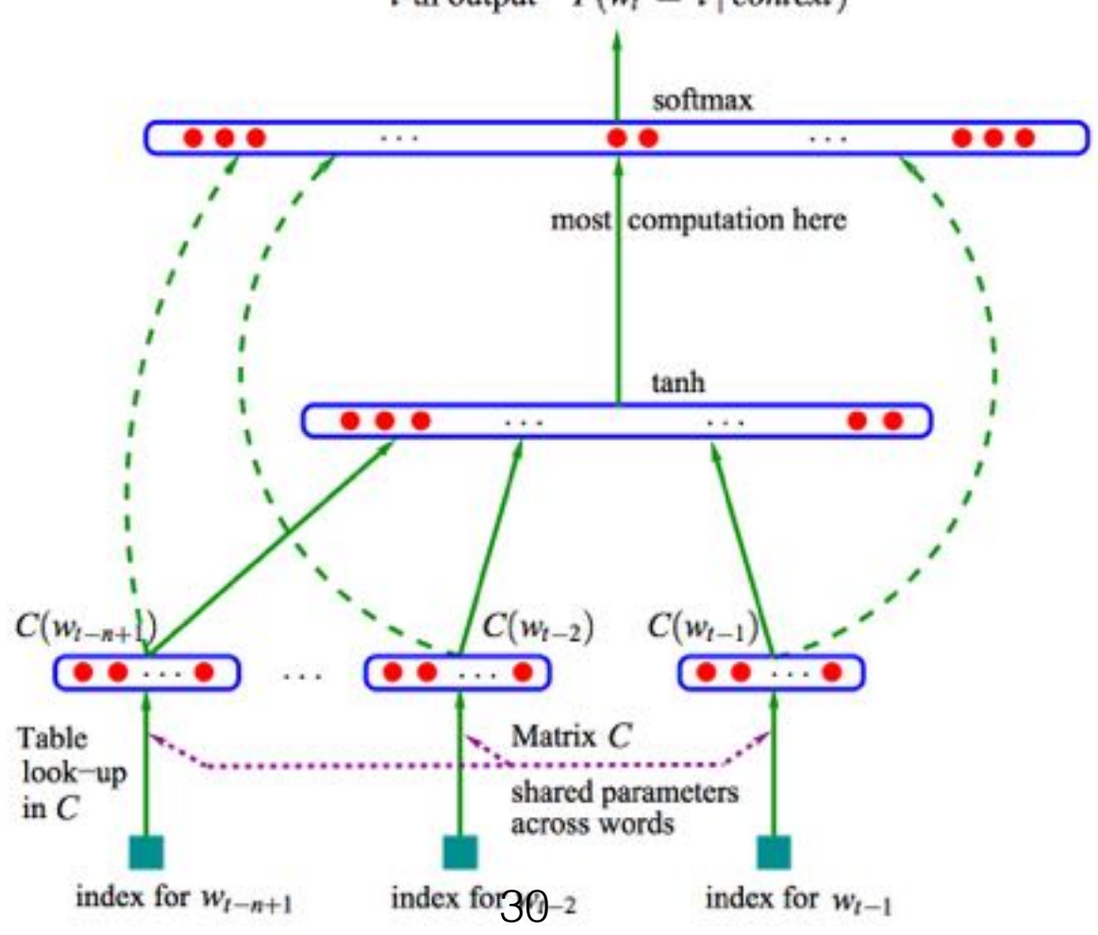
adapted rom Bengio, "Representation Learning and Deep Learning", July, 2012, UCLA

How?


hidden layer
Input 1
Input 2
Input 3
Input 4
output 1
output 2
Shallow neural network
weights
inputs
x_1
x_2
x_3
x_n
w_{1j}
w_{2j}
w_{3j}
w_{nj}
Σ
transfer function
net input
net_j
activation functon
o_j
activation

How?

- Representation of words as continuous vectors has a long history (Hinton et al. 1986; Rumelhart et al. 1986; Elman 1990)
- First neural network language model: NNLM (Bengio et al. 2001; Bengio et al. 2003) based on earlier ideas of distributed representations for symbols (Hinton 1986)



Vector Space Model

In a perfect world:

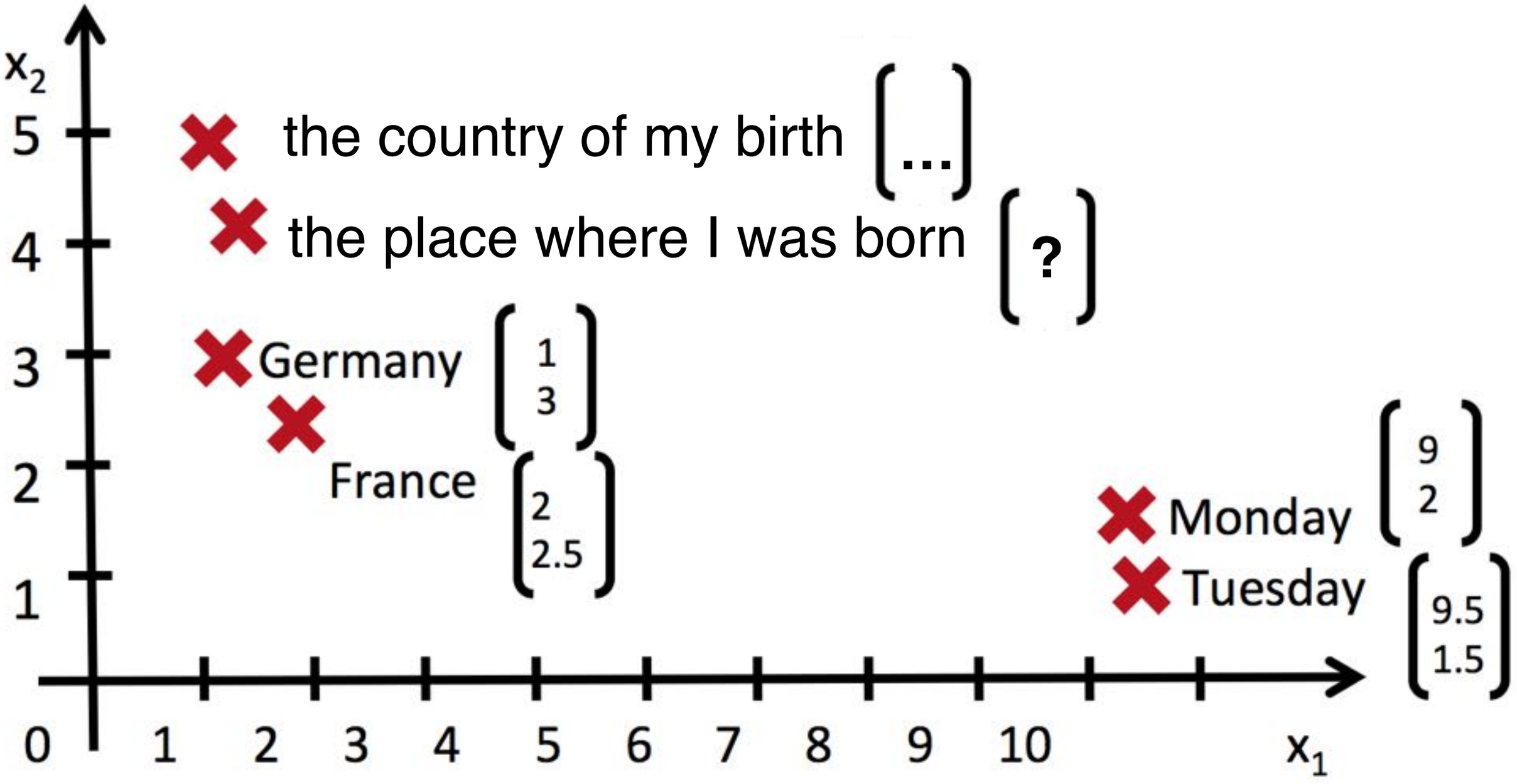


Figure (edited) from Bengio, "Representation Learning and Deep Learning", July, 2012, UCLA

How? Language Compositionality

Principle of compositionality:

the “meaning (**vector**) of a complex expression (**sentence**) is determined by:

- the meanings of its constituent expressions (**words**) and
- the rules (**grammar**) used to combine them”

— Gottlob Frege
(1848 - 1925)

Neural Networks for NLP

- Can theoretically (given enough units) approximate "any" function
- and fit to "any" kind of data
- Efficient for NLP: hidden layers can be used as word lookup tables
- Dense distributed word vectors + efficient NN training algorithms:
  - Can scale to billions of words !

Compositionality

- How do we handle the compositionality of language in our models?

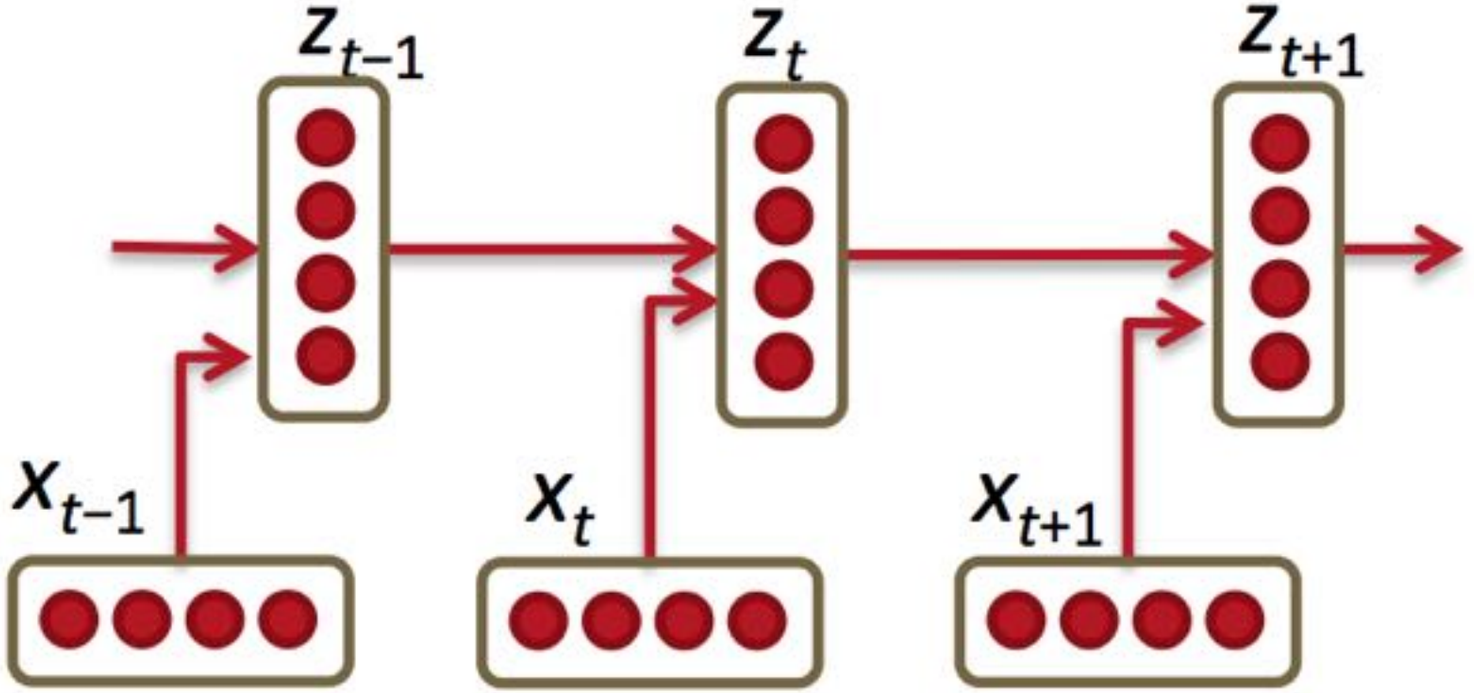
Compositionality

- How do we handle the compositionality of language in our models?
- **Recursion** :

  the same operator (same parameters) is applied repeatedly on different components

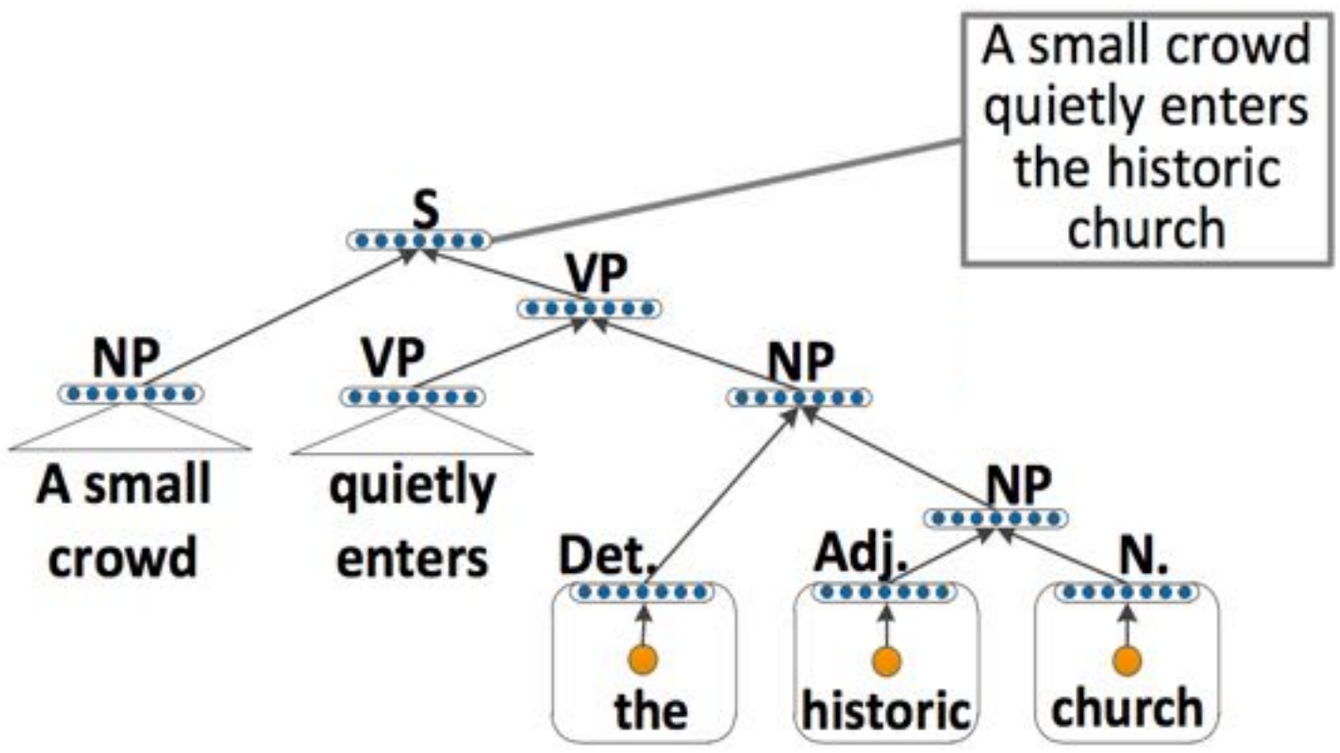
RNN 1: Recurrent Neural Networks

- How do we handle the compositionality of language in our models?
- Option 1: **Recurrent** Neural Networks (RNN)



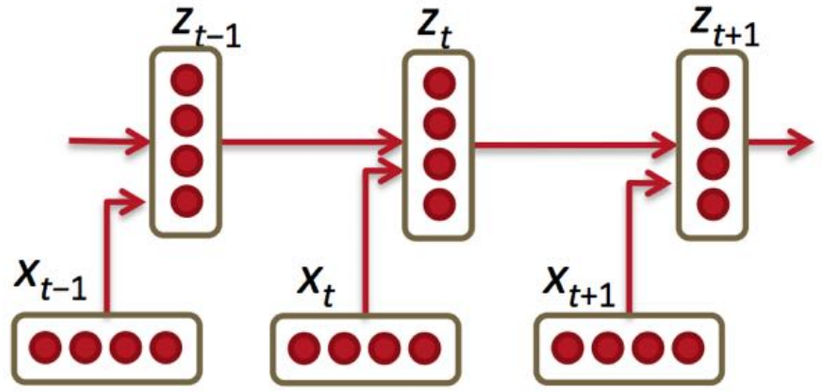
RNN 2: Recursive Neural Networks

- How do we handle the compositionality of language in our models?
- Option 2: **Recursive** Neural Networks (also sometimes called RNN)



Recurrent Neural Networks

- achieved SOTA in 2011 on Language Modeling (WSJ AR task) (Mikolov et al., INTERSPEECH 2011):



"Comparison to other LMs shows that RNN LMs are state of the art by a large margin. Improvements inrease with more training data."

Model	Eval WER[%]
Baseline - KN5	17.2
Discriminative LM [14]	16.9
All RNN	**14.4**

- and again at ASRU 2011:

> *"[RNN LM trained on a] single core on 400M words in a few days, with 1% absolute improvement in WER on state of the art setup"*

Mikolov, T., Karafiat, M., Burget, L., Cernock, J.H., Khudanpur, S. (2011)
Recurrent neural network based language model

Recurrent Neural Networks

input $$x(t) = w(t) + s(t-1)$$

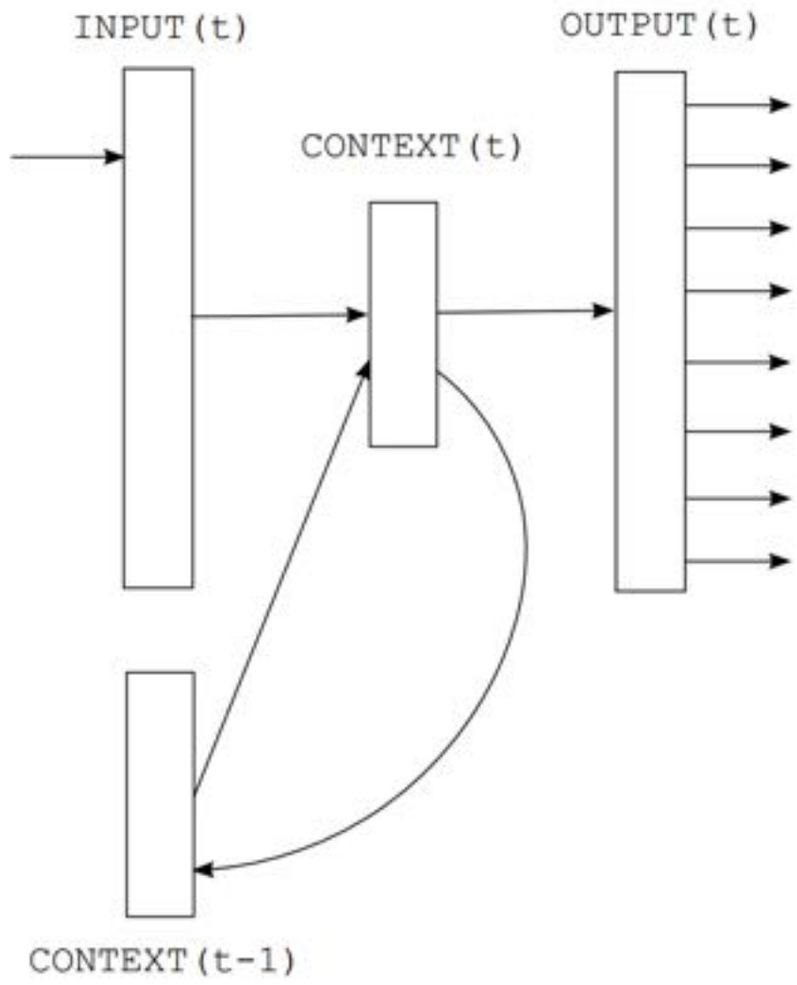
hidden layer(s) $$s_j(t) = f\left(\sum_i x_i(t) u_{ji}\right)$$

output layer $$y_k(t) = g\left(\sum_j s_j(t) v_{kj}\right)$$

+ sigmoid activation function

+ softmax function:

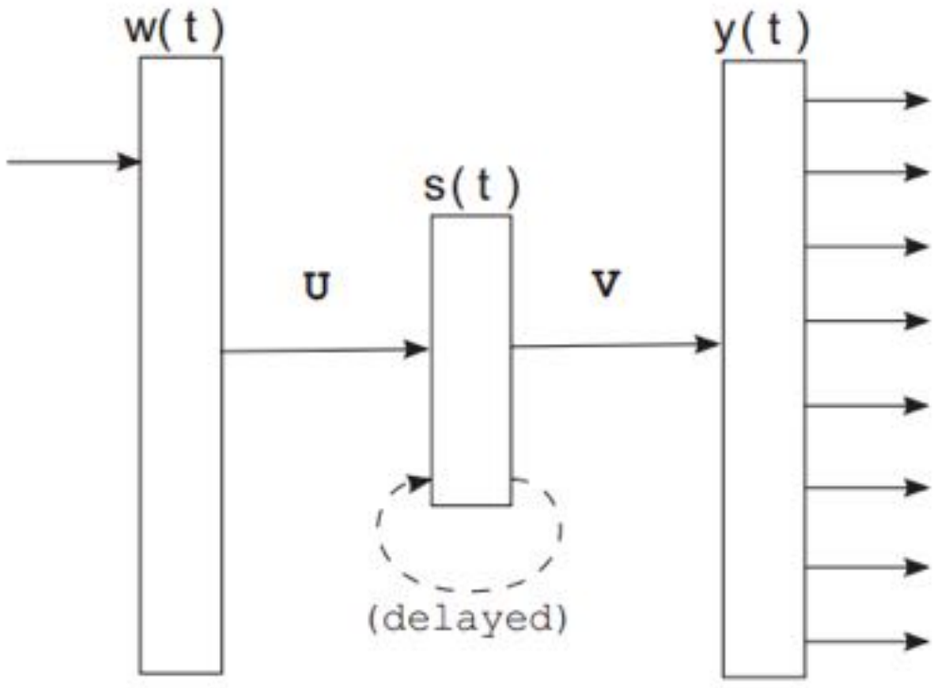
$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$$



(simple recurrent neural network for LM)

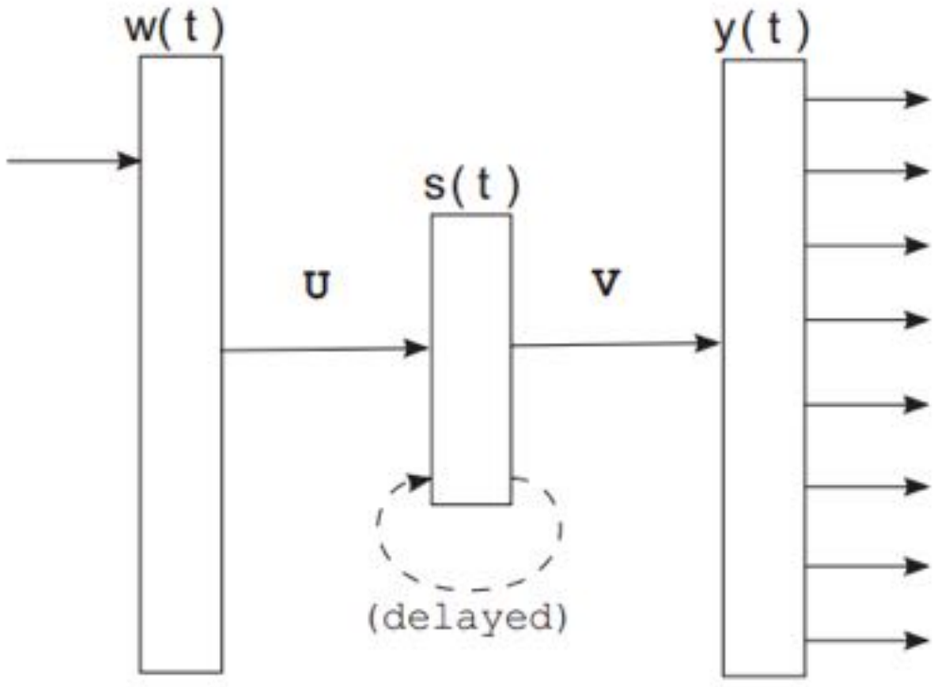
Mikolov, T., Karafiat, M., Burget, L., Cernock, J.H., Khudanpur, S. (2011)
Recurrent neural network based language model

Recurrent Neural Networks



backpropagation through time

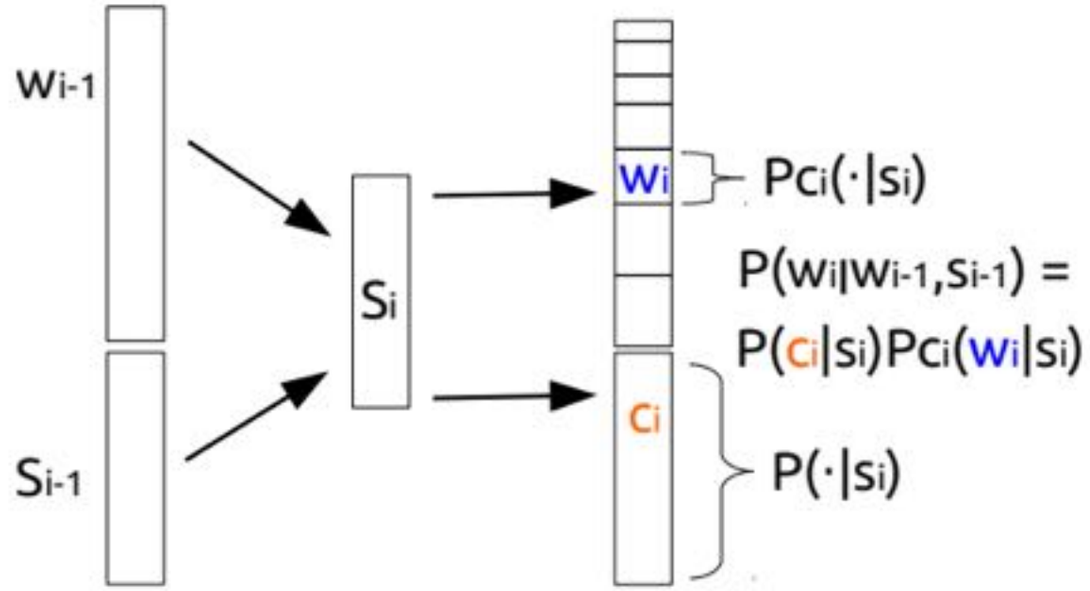
Recurrent Neural Networks



backpropagation through time

$$P_{rnn}(w_i|s_i) =$$

$$P_{rnn}(c_i|s_i)P^{c_i}_{rnn}(w_i|s_i)$$

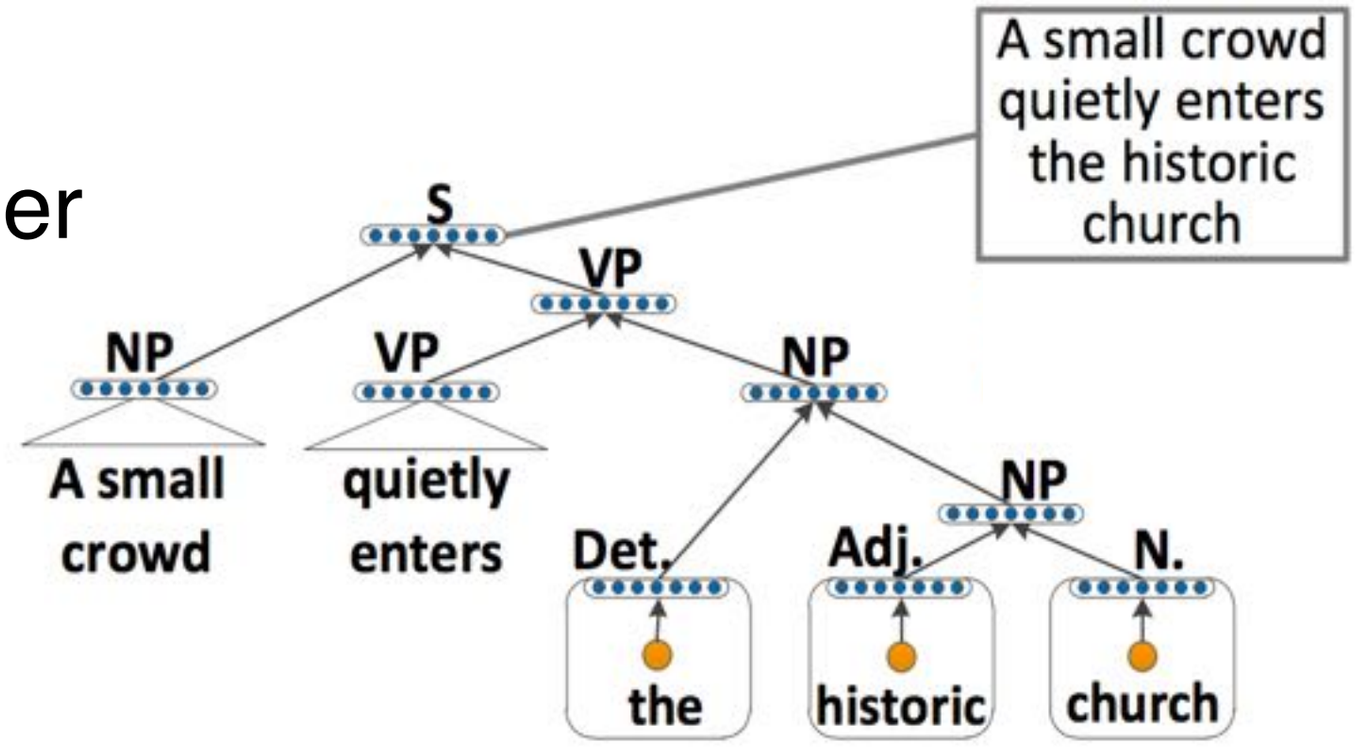


class based recurrent NN

[code (Mikolov's RNNLM Toolkit) and more info: http://rnnlm.org/]

Recursive Neural Network

- Recursive Neural Network for LM (Socher et al. 2011; Socher 2014)

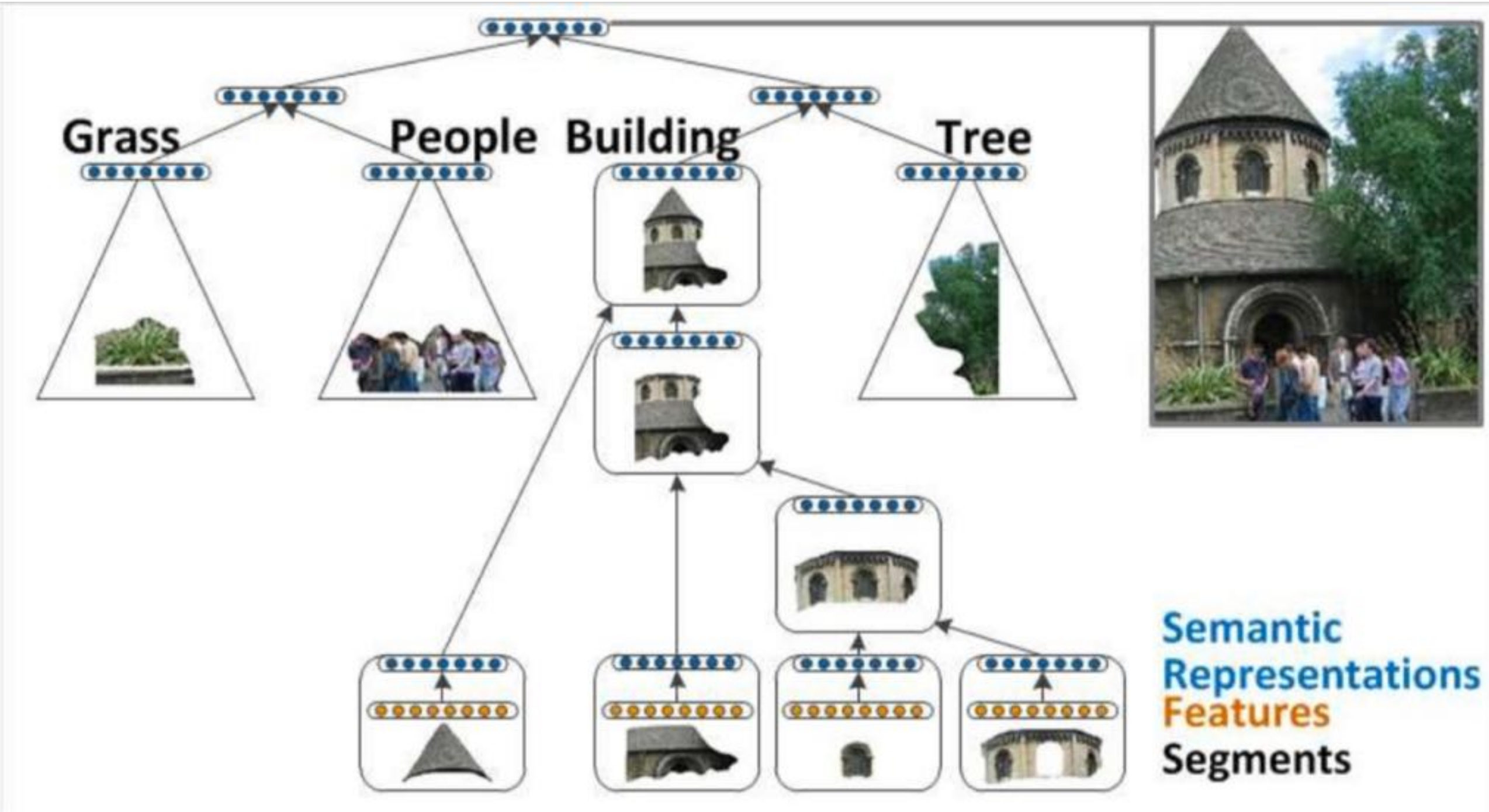


- achieved SOTA on new Stanford Sentiment Treebank dataset (but comparing it to many other models):

Model	Fine-grained		Positive/Negative	
	All	Root	All	Root
NB	67.2	41.0	82.6	81.8
SVM	64.3	40.7	84.6	79.4
BiNB	71.0	41.9	82.7	83.1
VecAvg	73.3	32.7	85.1	80.1
RNN	79.0	43.2	86.1	82.4
MV-RNN	78.7	44.4	86.8	82.9
RNTN	**80.7**	**45.7**	**87.6**	**85.4**

Socher, R., Perelygin,, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C. (2013)
Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank
info & code: http://nlp.stanford.edu/sentiment/

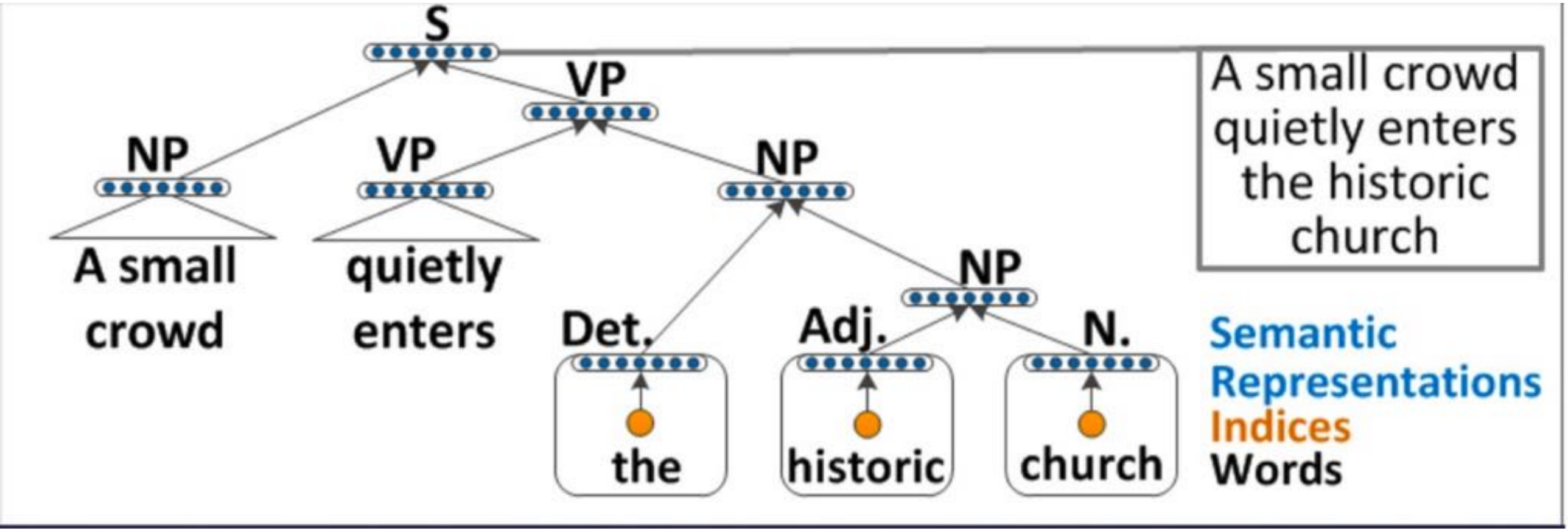
Recursive Neural Tensor Network



Socher, R., Liu, C.C., NG, A.Y., Manning, C.D. (2011)
Parsing Natural Scenes and Natural Language with Recursive Neural Networks

code & info: http://www.socher.org/index.php/Main/ParsingNaturalScenesAndNaturalLanguageWithRecursiveNeuralNetworks

Recursive Neural Tensor Network



Recursive Neural Network

- RNN (Socher et al. 2011a)

Model	Fine-grained		Positive/Negative	
	All	Root	All	Root
NB	67.2	41.0	82.6	81.8
SVM	64.3	40.7	84.6	79.4
BiNB	71.0	41.9	82.7	83.1
VecAvg	73.3	32.7	85.1	80.1
RNN	79.0	43.2	86.1	82.4
MV-RNN	78.7	44.4	86.8	82.9
RNTN	**80.7**	**45.7**	**87.6**	**85.4**

Socher, R., Perelygin,, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C. (2013)
Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank
info & code: http://nlp.stanford.edu/sentiment/

Recursive Neural Network

- RNN (Socher et al. 2011a)
- Matrix-Vector RNN (MV-RNN) (Socher et al., 2012)

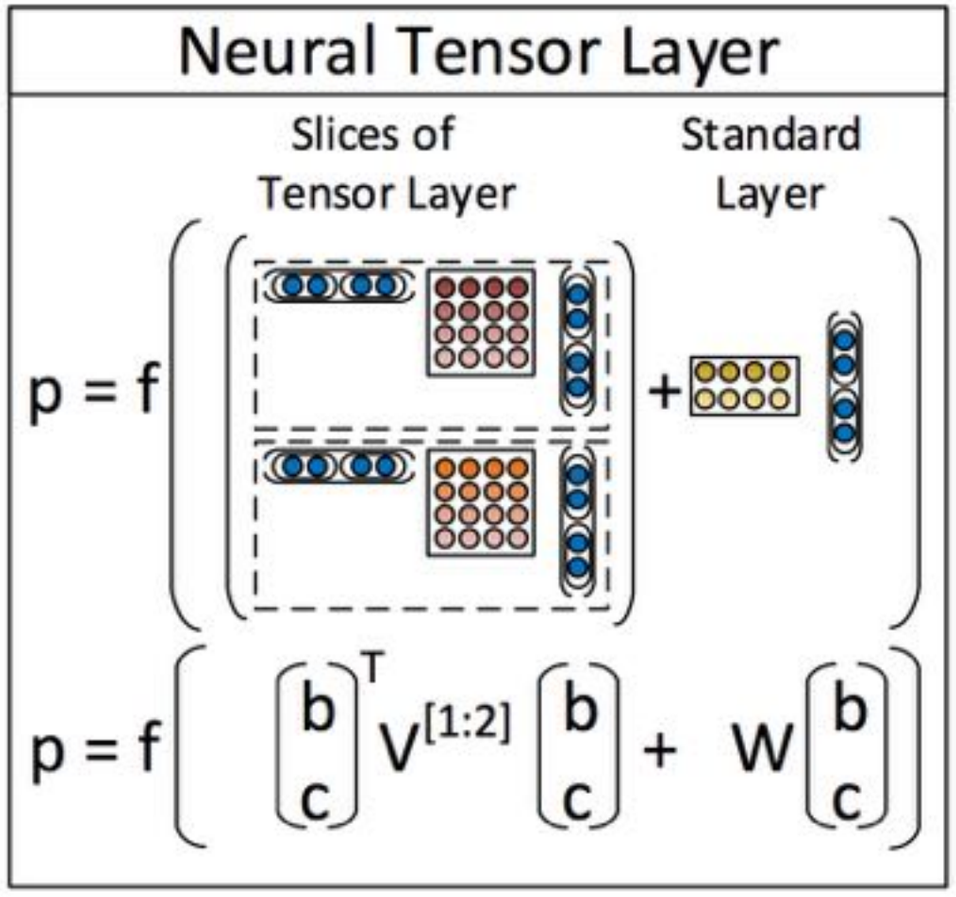
Model	Fine-grained		Positive/Negative	
	All	Root	All	Root
NB	67.2	41.0	82.6	81.8
SVM	64.3	40.7	84.6	79.4
BiNB	71.0	41.9	82.7	83.1
VecAvg	73.3	32.7	85.1	80.1
RNN	79.0	43.2	86.1	82.4
MV-RNN	78.7	44.4	86.8	82.9
RNTN	**80.7**	**45.7**	**87.6**	**85.4**

Socher, R., Perelygin,, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C. (2013)
Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank
info & code: http://nlp.stanford.edu/sentiment/

Recursive Neural Network

- RNN (Socher et al. 2011a)
- Matrix-Vector RNN (MV-RNN) (Socher et al., 2012)
- Recursive Neural Tensor Network (RNTN) (Socher et al. 2013)

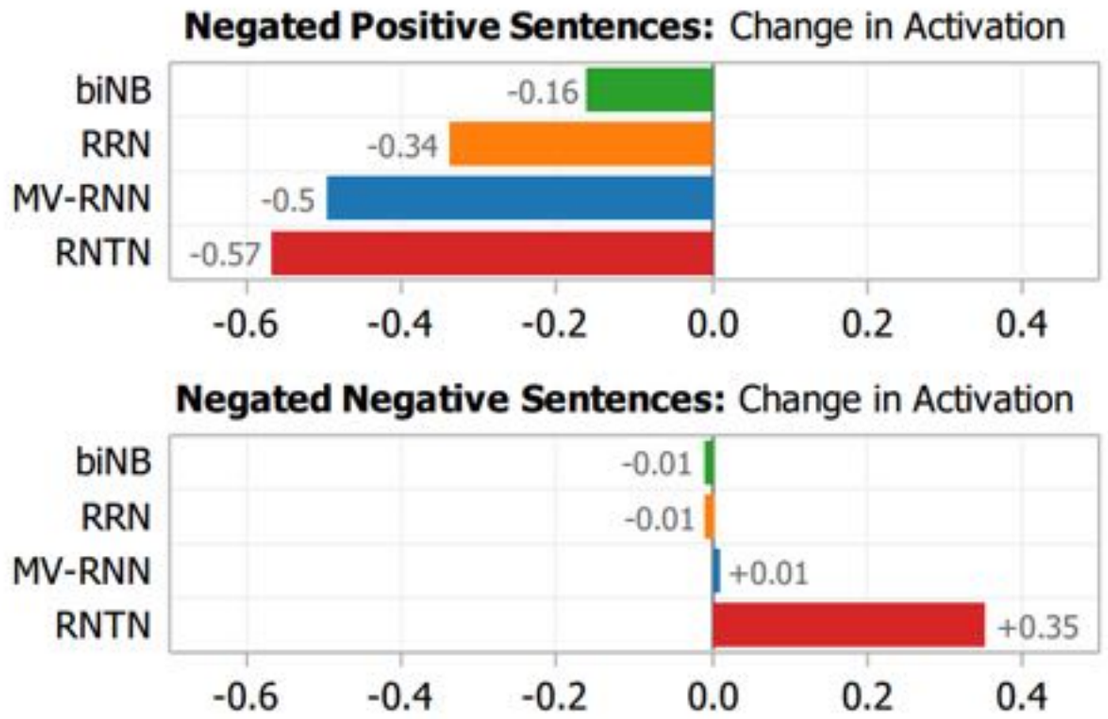
Model	Fine-grained		Positive/Negative	
	All	Root	All	Root
NB	67.2	41.0	82.6	81.8
SVM	64.3	40.7	84.6	79.4
BiNB	71.0	41.9	82.7	83.1
VecAvg	73.3	32.7	85.1	80.1
RNN	79.0	43.2	86.1	82.4
MV-RNN	78.7	44.4	86.8	82.9
RNTN	**80.7**	**45.7**	**87.6**	**85.4**



Recursive Neural Network

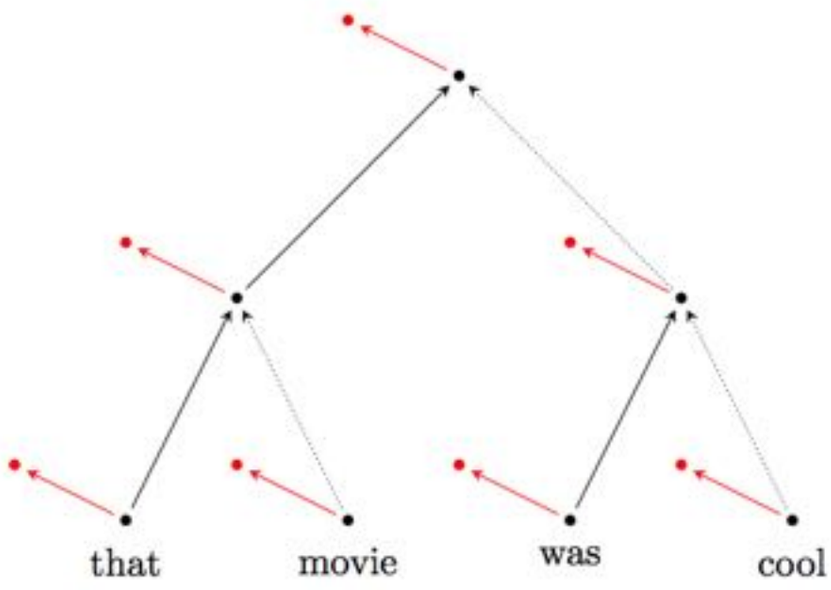
- negation detection:

Model	Accuracy	
	Negated Positive	Negated Negative
biNB	19.0	27.3
RNN	33.3	45.5
MV-RNN	52.4	54.6
RNTN	**71.4**	**81.8**

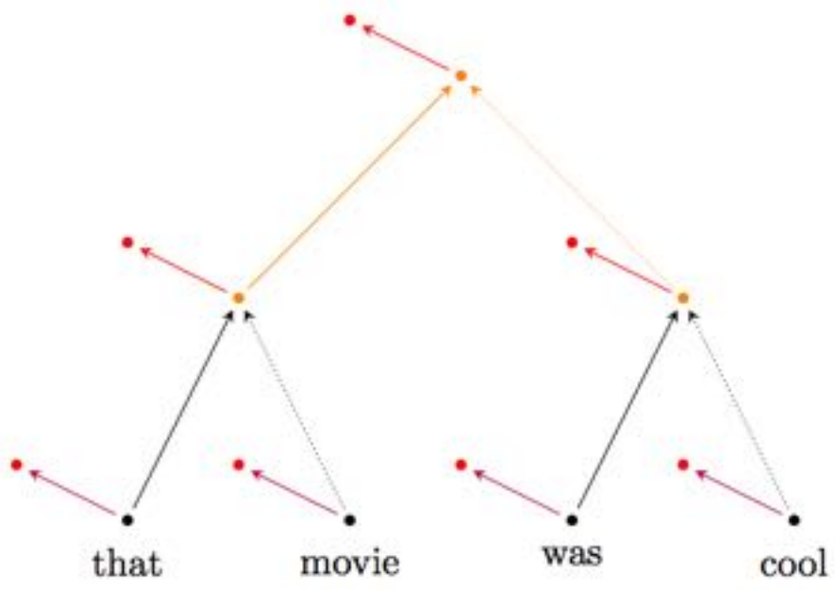


Socher, R., Perelygin,, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C. (2013)
Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank
info & code: http://nlp.stanford.edu/sentiment/

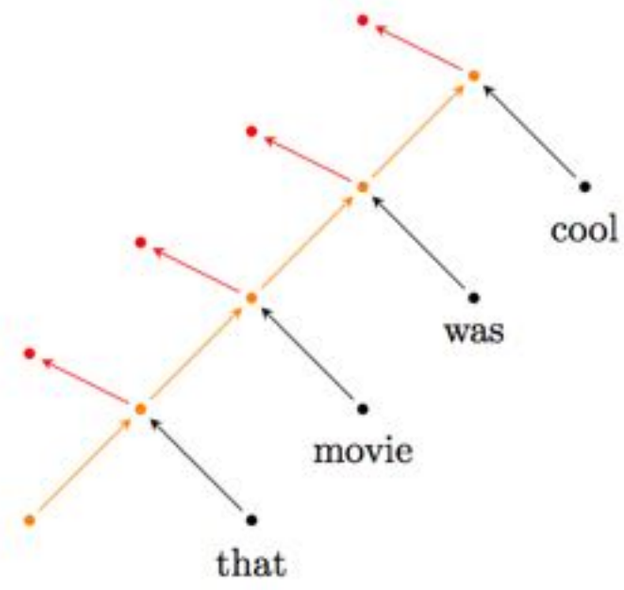
Recurrent vs Recursive



recursive net



untied recursive net



recurrent net

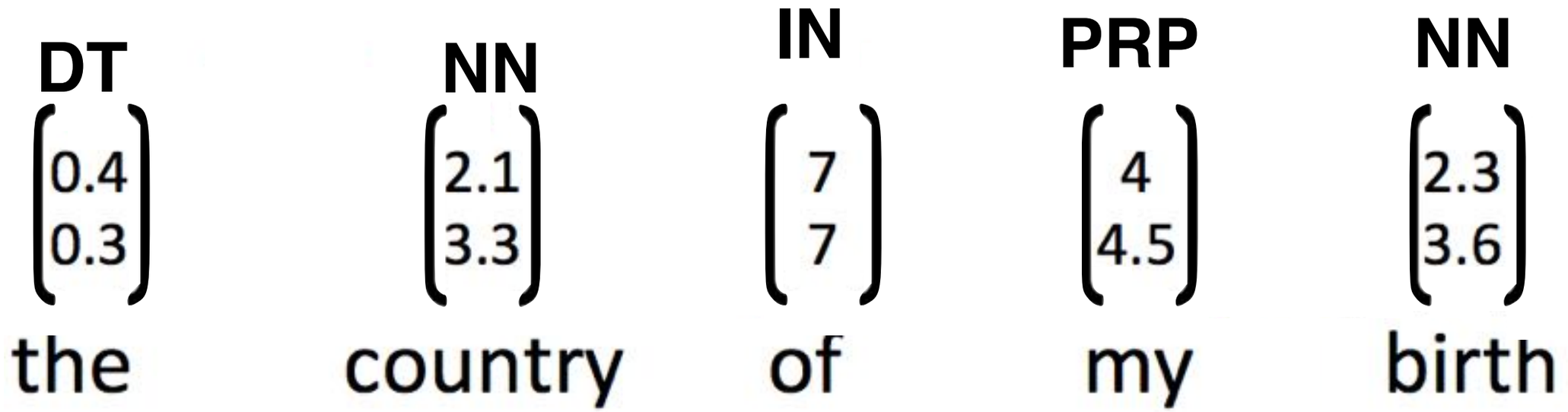
Recurrent NN for Vector Space

the country of my birth

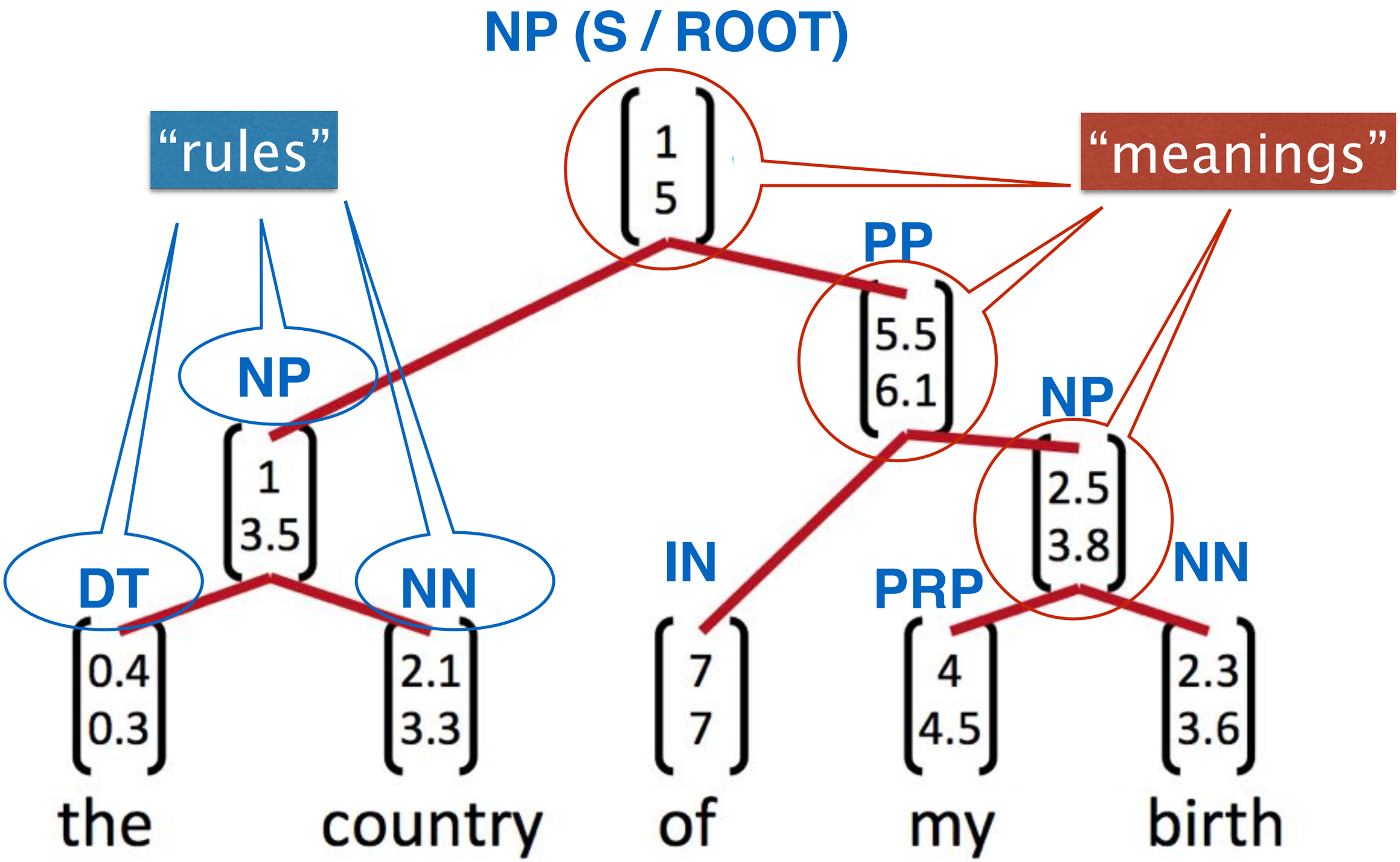
Recurrent NN for Vector Space

DT	NN	IN	PRP	NN
the	country	of	my	birth

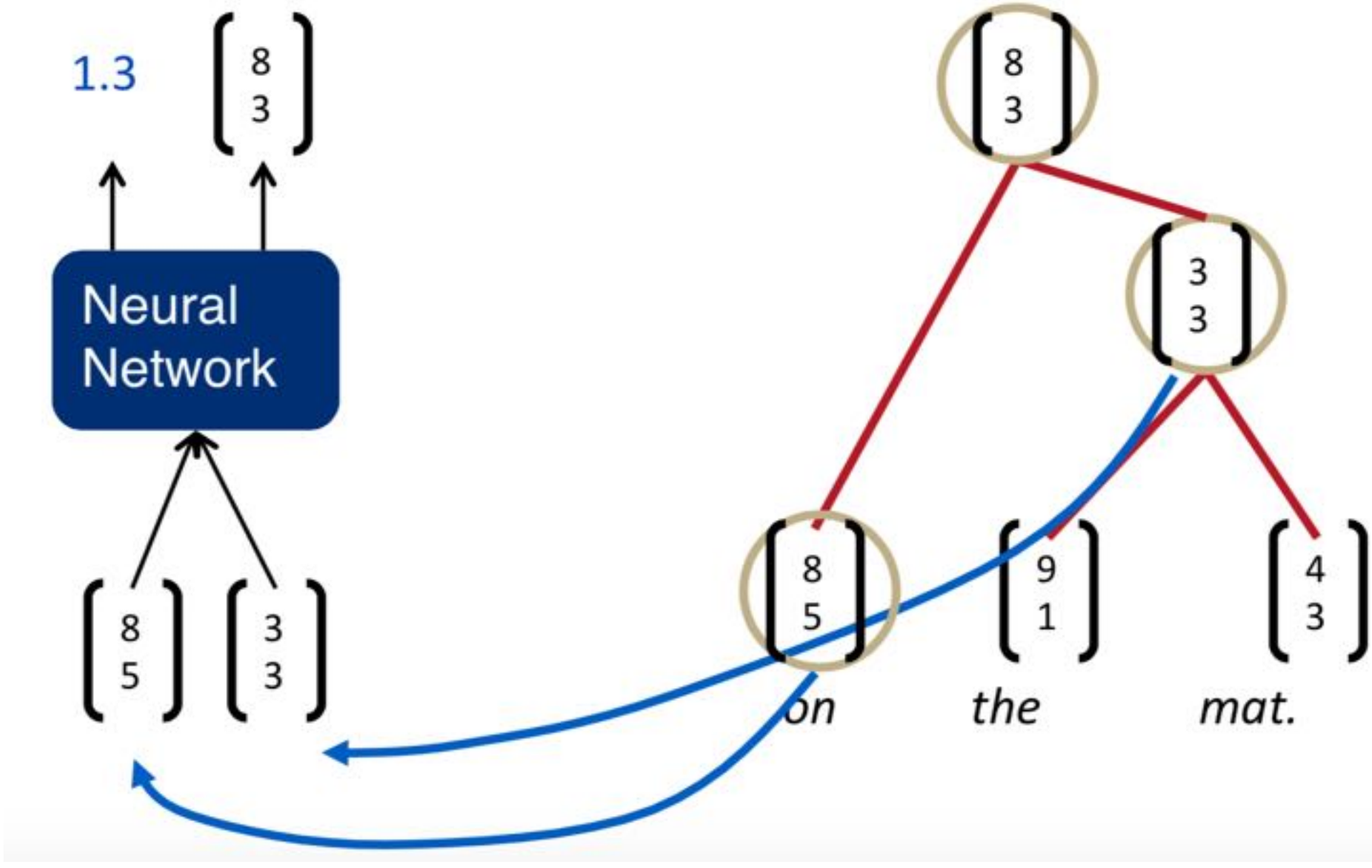
Recurrent NN for Vector Space



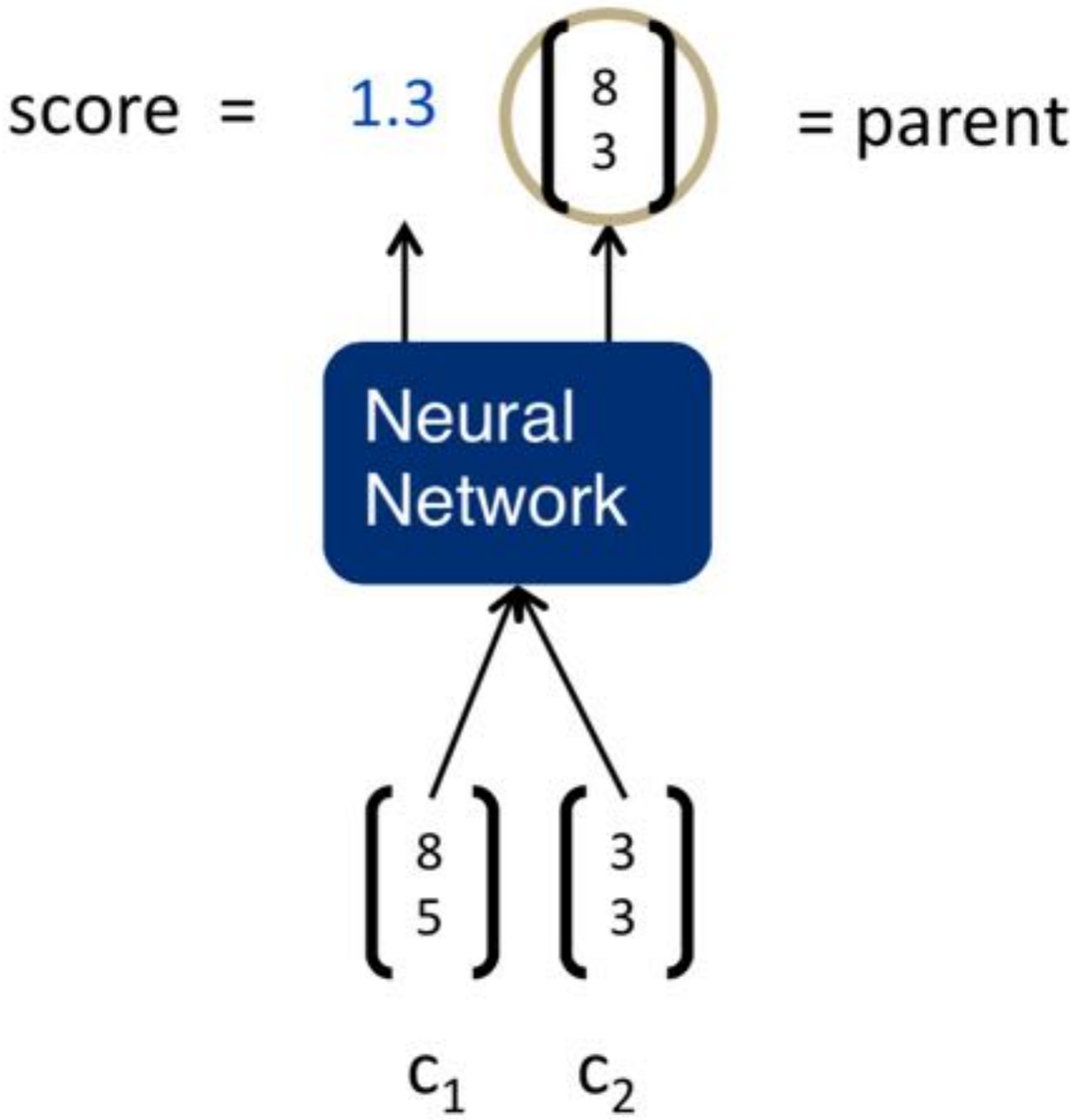
Recurrent NN for Vector Space



Recurrent NN for Vector Space



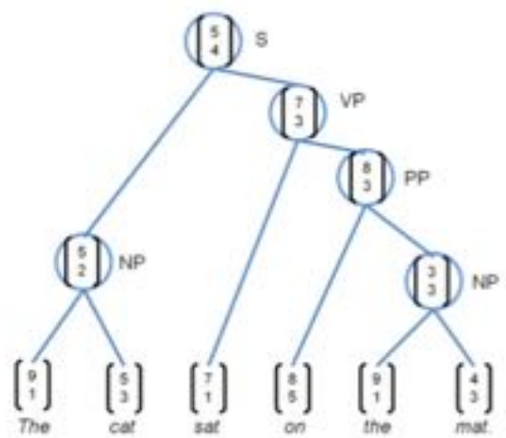
Recurrent NN for Vector Space



$$\text{score} = U^T p$$

$$p = \tanh\left(W \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + b\right),$$

Same W parameters at all nodes of the tree

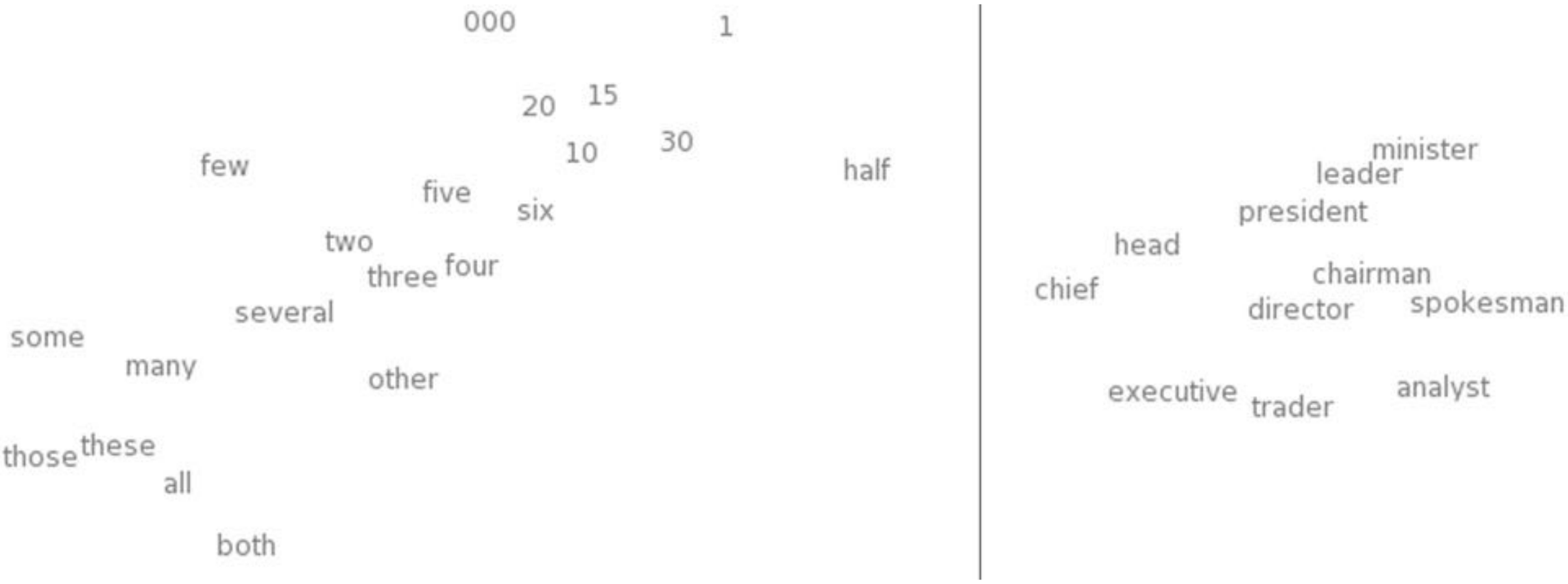


Word Embeddings: Turian (2010)

Turian, J., Ratinov, L., Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning

code & info: http://metaoptimize.com/projects/wordreprs/

Word Embeddings: Turian (2010)



Turian, J., Ratinov, L., Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning

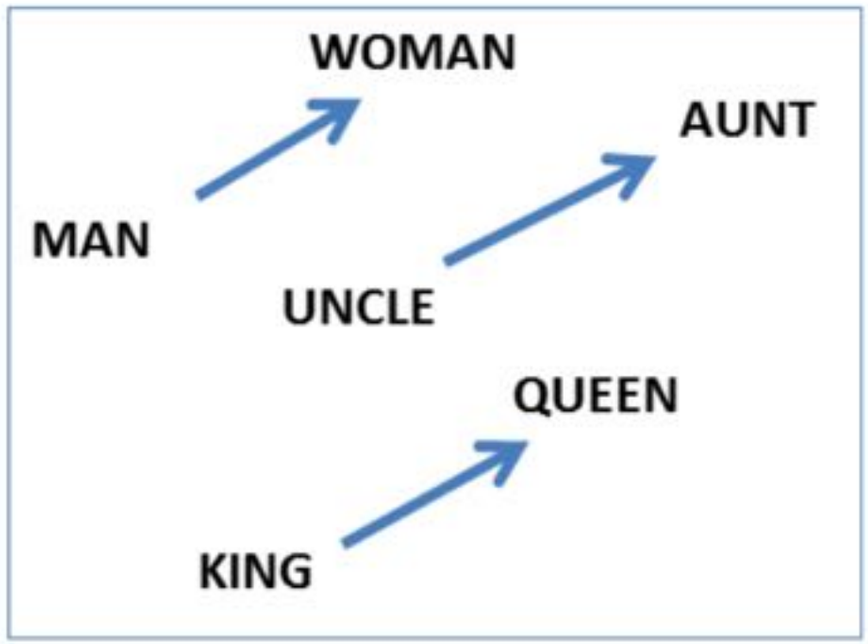
code & info: http://metaoptimize.com/projects/wordreprs/

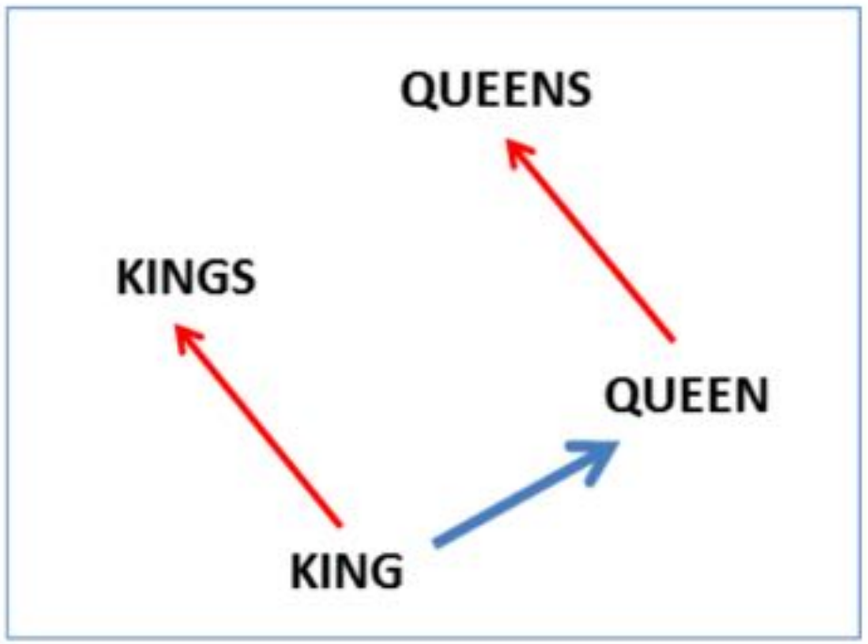
Word Embeddings: Collobert & Weston (2011)

FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
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

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P. (2011) .
Natural Language Processing (almost) from Scratch

Word2Vec & Linguistic Regularities: Mikolov (2013)





Rela

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

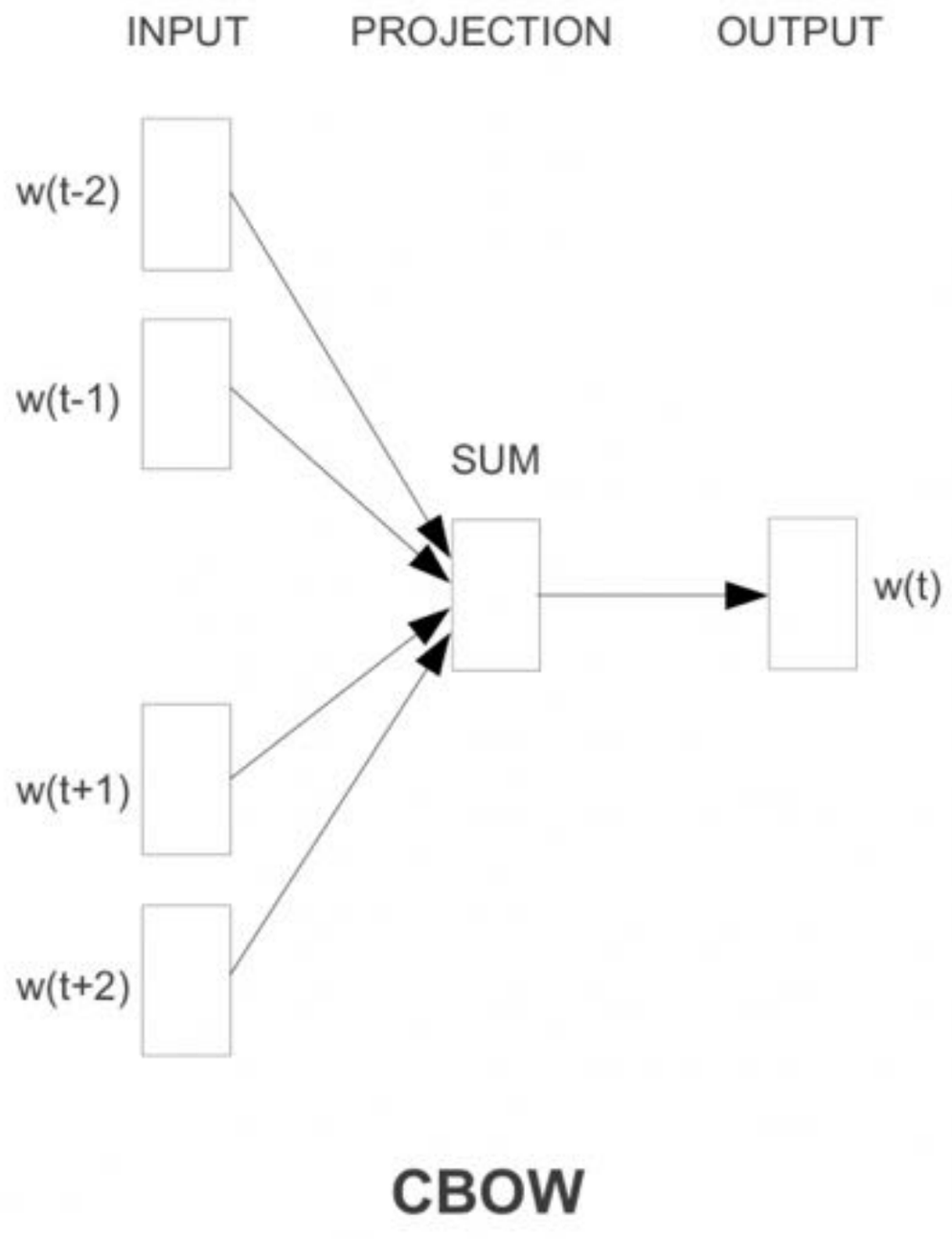
Mikolov, T., Yih, W., & Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations

code & info: https://code.google.com/p/word2vec/

Word2Vec & Linguistic Regularities: Mikolov (2013)

Continuous BoW (CBOW) Model

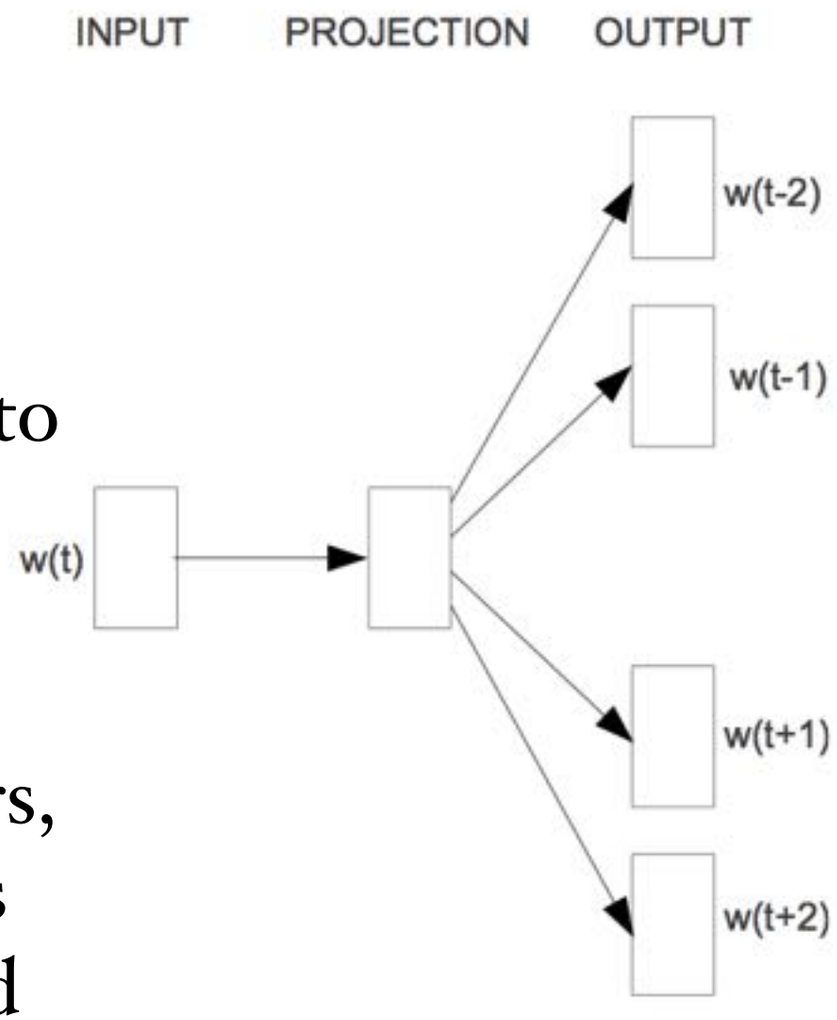
- Similar to the feedforward NNLM, but
- Non-linear hidden layer removed
- Projection layer shared for all words – Not just the projection matrix
- Thus, all words get projected into the same position – Their vectors are just averaged
- Called CBOW (continuous BoW) because the order of the words is lost
- Another modification is to use words from past and from future (window centered on current word)



Word2Vec & Linguistic Regularities: Mikolov (2013)

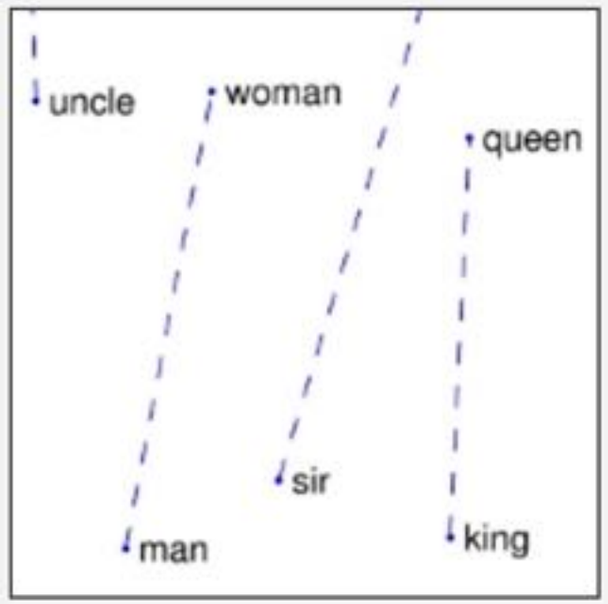
Continuous Skip-gram Model

- Similar to CBOW, but instead of predicting the current word based on the context
- Tries to maximize classification of a word based on another word in the same sentence
- Thus, uses each current word as an input to a log-linear classifier
- Predicts words within a certain window
- Observations – Larger window size => better quality of the resulting word vectors, higher training time – More distant words are usually less related to the current word than those close to it – Give less weight to the distant words by sampling less from those words in the training examples

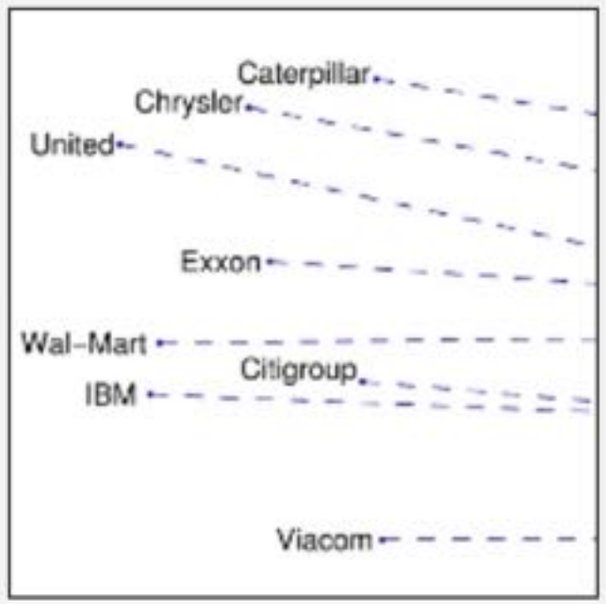


Skip-gram

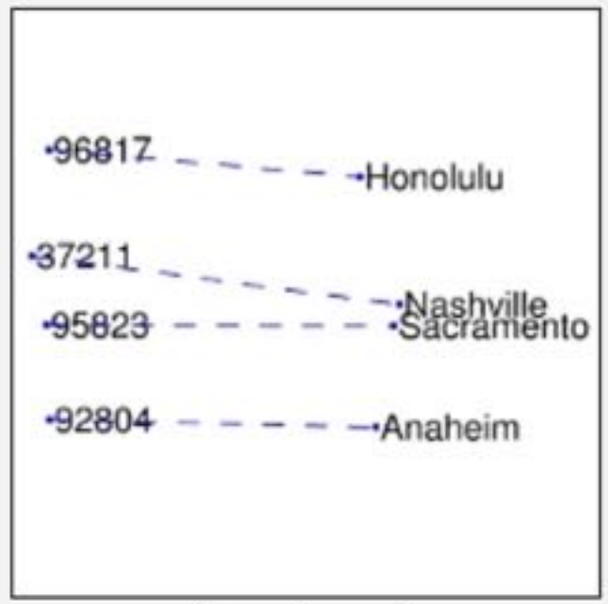
GloVe: Global Vectors for Word Representation (Pennington 2015)



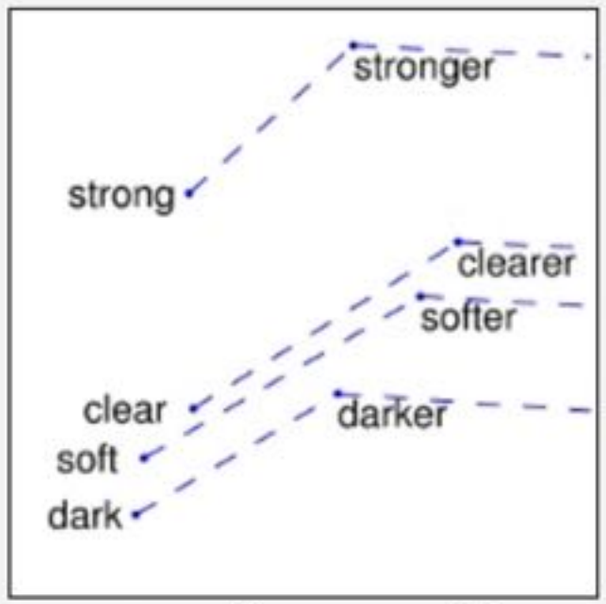
man - woman



company - ceo



city - zip code



comparative - superlative

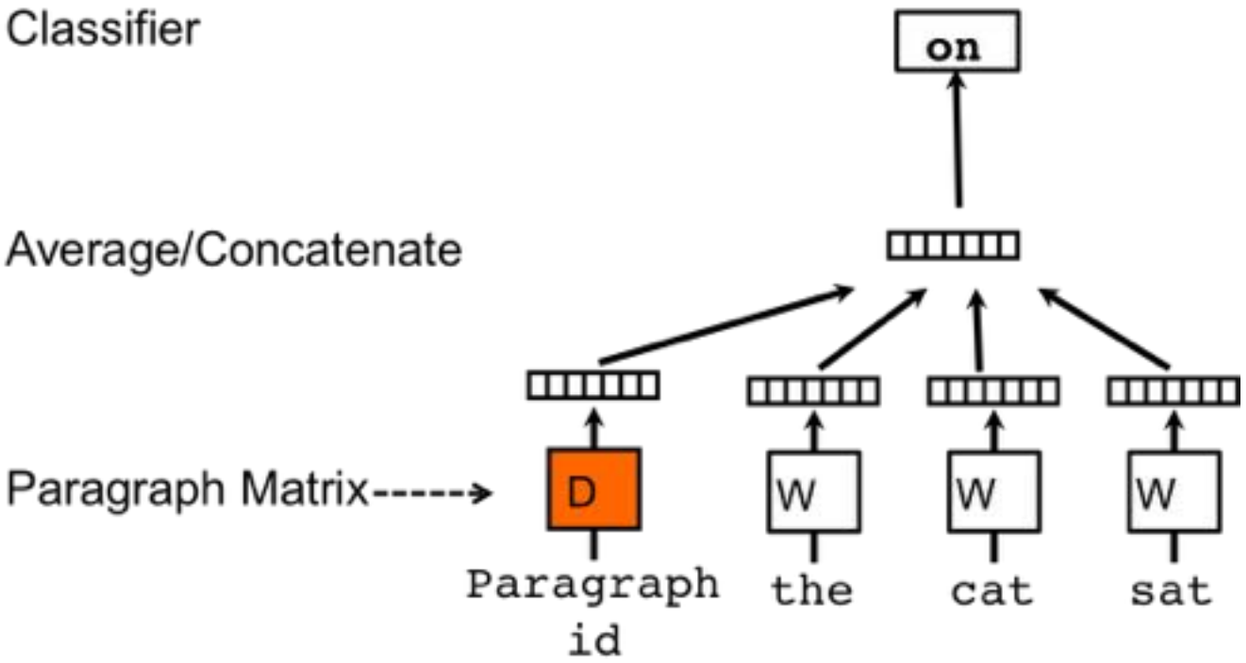
Glove == Word2Vec?

Omer Levy & Yoav Goldberg (2014) Neural Word Embedding as Implicit Matrix Factorization

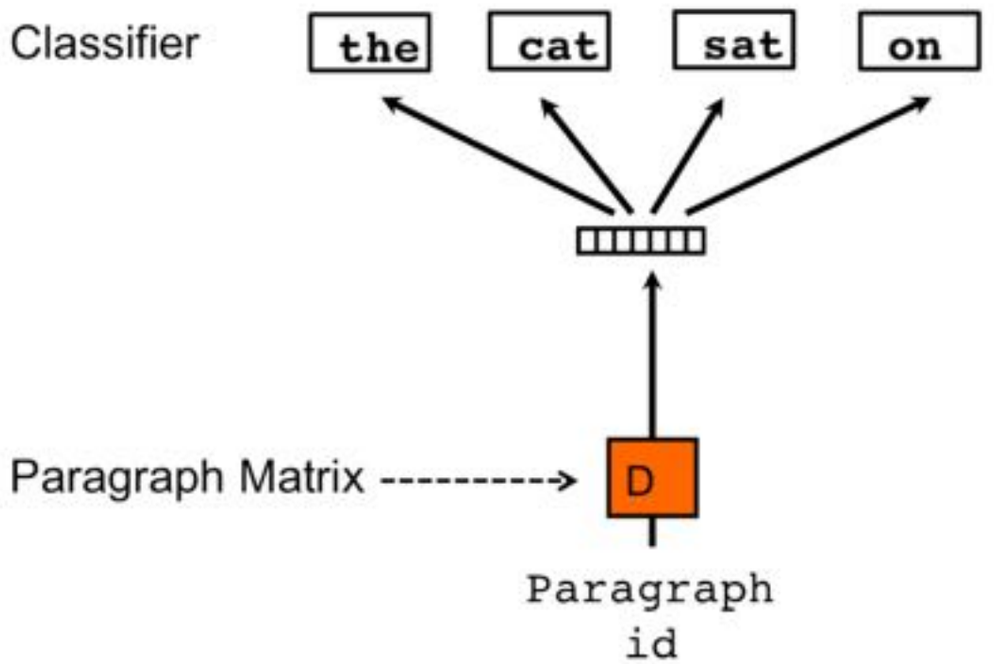
Paragraph Vectors: Dai et al. (2014)

Dai, A., Olah,. C., Le, Q., Corrado, G. (2014) Document Embedding with Paragraph Vectors

Paragraph Vectors: Dai et al. (2014)



Paragraph vector with distributed memory (PV-DM)



Distributed bag of words version (PVDBOW)

Dai, A., Olah,. C., Le, Q., Corrado, G. (2014) Document Embedding with Paragraph Vectors

Paragraph Vectors: Dai et al. (2014)

(a) Wikipedia nearest neighbours to "Lady Gaga" using Paragraph Vectors. All articles are relevant.

Article	Cosine Similarity
Christina Aguilera	0.674
Beyonce	0.645
Madonna (entertainer)	0.643
Artpop	0.640
Britney Spears	0.640
Cyndi Lauper	0.632
Rihanna	0.631
Pink (singer)	0.628
Born This Way	0.627
The Monster Ball Tour	0.620

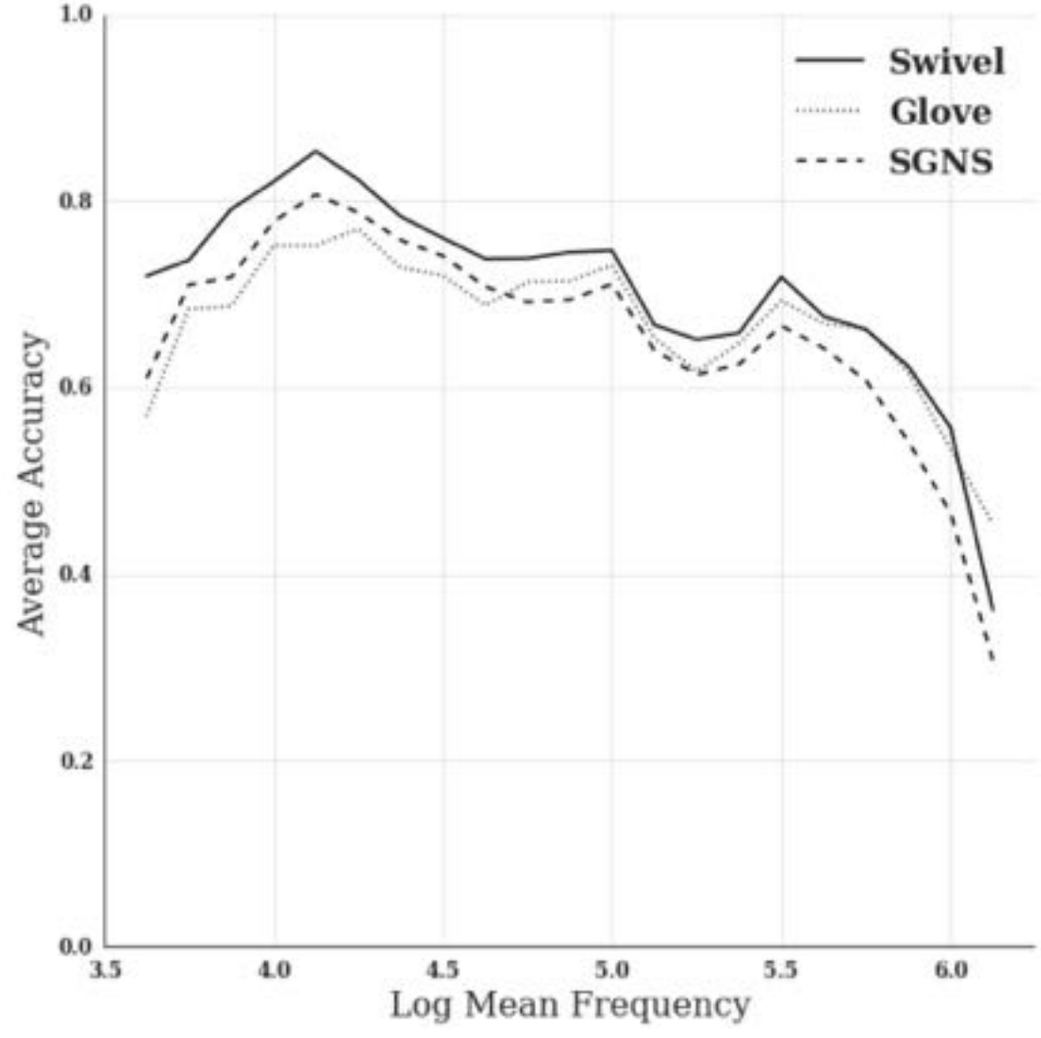
(b) Wikipedia nearest neighbours to pv("Lady Gaga") - wv("American") + wv("Japanese") using Paragraph Vectors. Note that Ayumi Hamasaki is one of the most famous singers, and one of the best selling artists in Japan. She also has an album called "Poker Face" in 1998.

Article	Cosine Similarity
Ayumi Hamasaki	0.539
Shoko Nakagawa	0.531
Izumi Sakai	0.512
Urbangarde	0.505
Ringo Sheena	0.503
Toshiaki Kasuga	0.492
Chihiro Onitsuka	0.487
Namie Amuro	0.485
Yakuza (video game)	0.485
Nozomi Sasaki (model)	0.485

Dai, A., Olah,. C., Le, Q., Corrado, G. (2014) Document Embedding with Paragraph Vectors

Swivel: Improving Embeddings by Noticing What's Missing (Shazeer et al 2016)

Method	WordSim Similarity	WordSim Relatedness	Bruni et al. MEN	Radinsky et al. M. Turk	Luong et al. Rare Words	Hill et al. SimLex	Google	MSR
SGNS	0.737	0.592	0.743	0.686	0.467	0.397	0.692	0.592
GloVe	0.651	0.541	0.738	0.627	0.386	0.360	0.716	0.578
Swivel	**0.748**	**0.616**	**0.762**	**0.720**	**0.483**	**0.403**	**0.739**	**0.622**
CBOW	0.700	0.527	0.708	0.664	0.439	0.358	0.667	0.570

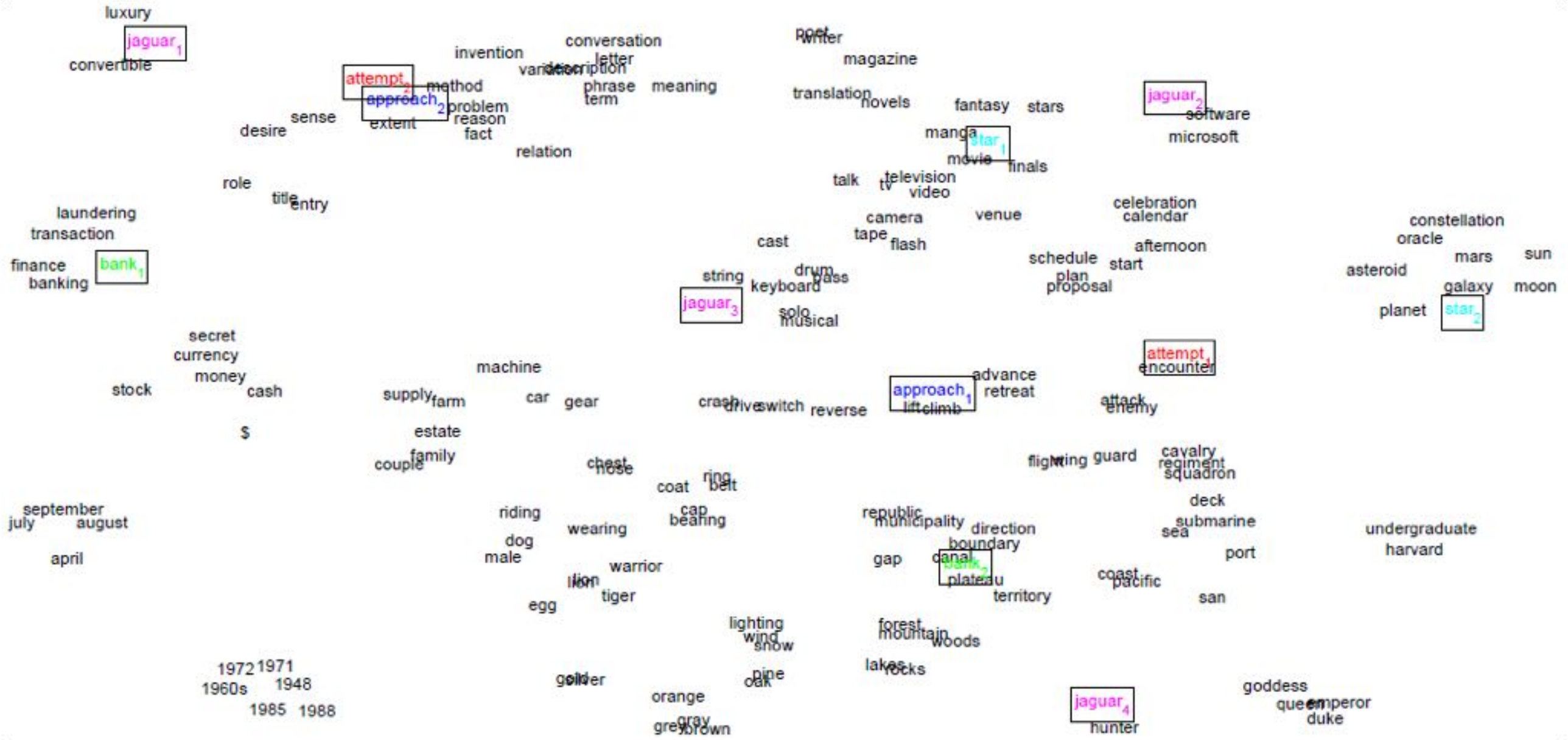


Swivel: Improving Embeddings by Noticing What's Missing (Shazeer et al 2016)

Table 3. Nearest neighbors for some very rare words.

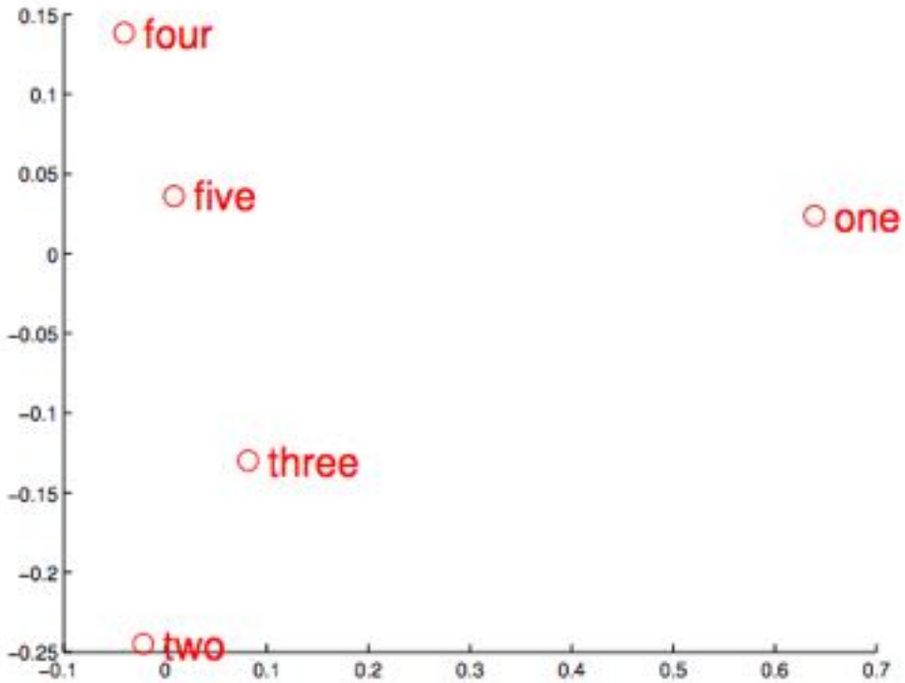
Query	Vocabulary Rank	SGNS	GloVe	Swivel
bootblack	393,709	shoeshiner, newsboy, shoeshine, stage-struck, bartender, bellhop, waiter, housepainter, tinsmith	redbull, 240, align=middle, 18, 119, dannit, concurrence/dissent, 320px, dannitdannit	newsboy, shoeshine, stevedore, bellboy, headwaiter, stowaway, tibbs, mister, tramp
chigger	373,844	chiggers, webworm, hairballs, noctuid, sweetbread, psyllids, rostratus, narrowleaf, pigweed	dannit, dannitdannit, upupidae, bungarus, applause., .774, amolops, maxillaria, paralympic.org	mite, chiggers, mites, batatas, infestation, jigger, infested, mumbo, frog's
decretal	374,123	decretals, ordinatio, sacerdotalis, constitutiones, theodosianus, canonum, papae, romanae, episcoporum	regesta, agatho, afl.com.au, dannitdannit, dannit, emptores, beatifications, 18, 545	decretals, decretum, apostolicae, sententiae, canonum, unigenitus, collectio, fidei, patristic
tuxedoes	396,973	tuxedos, ballgowns, tuxes, well-cut, cable-knit, open-collared, organdy, high-collared, flouncy	hairnets, dhotis, speedos, loincloths, zekrom, shakos, mortarboards, caftans, nightwear	ballgowns, tuxedos, tuxes, cummerbunds, daywear, bridesmaids', gowns, strapless, flouncy

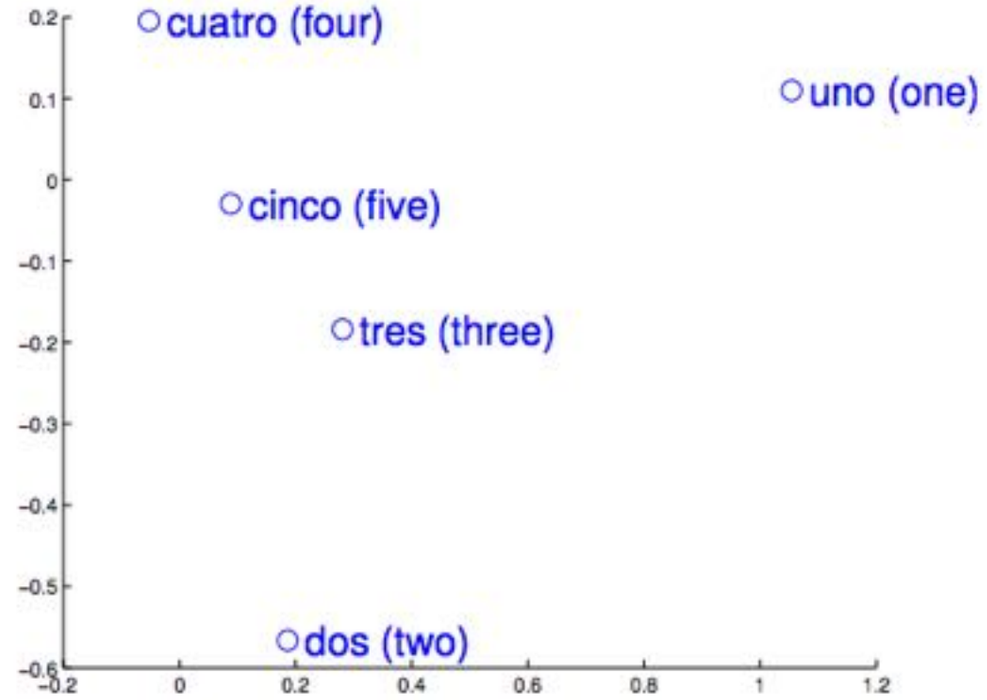
Multi-embeddings: Stanford (2012)

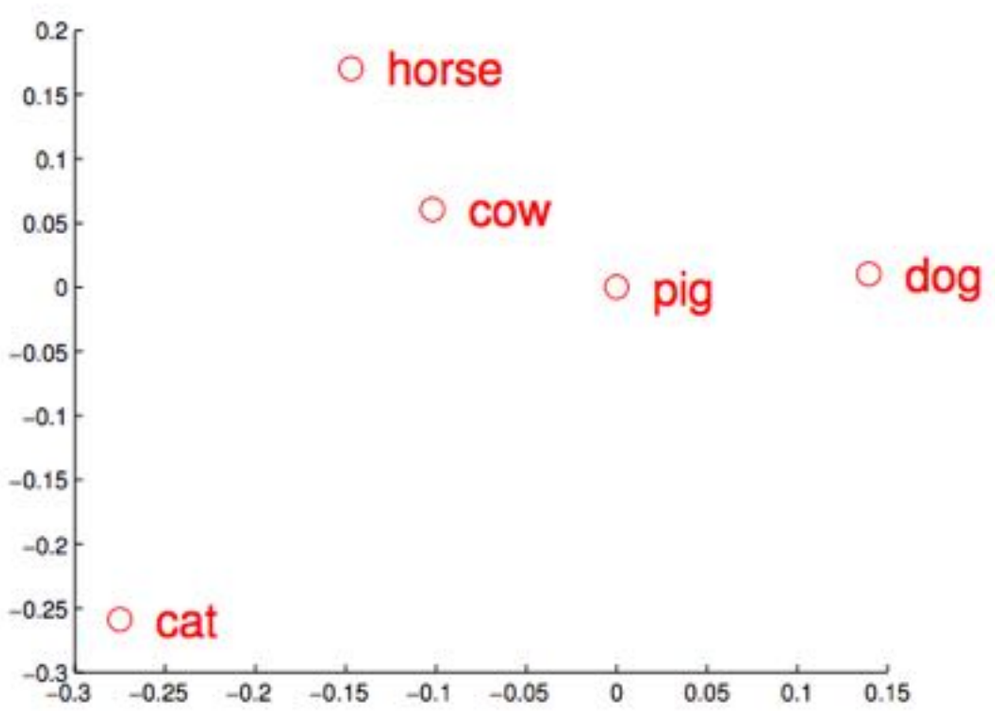


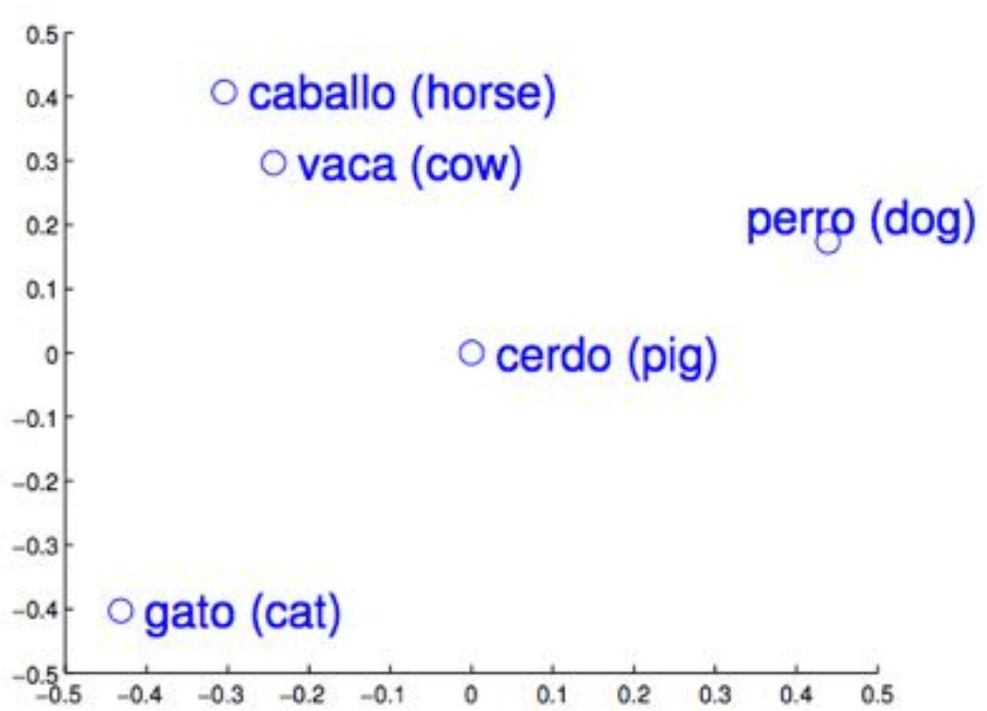
Eric H. Huang, Richard Socher, Christopher D. Manning, Andrew Y. Ng (2012)
Improving Word Representations via Global Context and Multiple Word Prototypes

Word Embeddings for MT: Mikolov (2013)



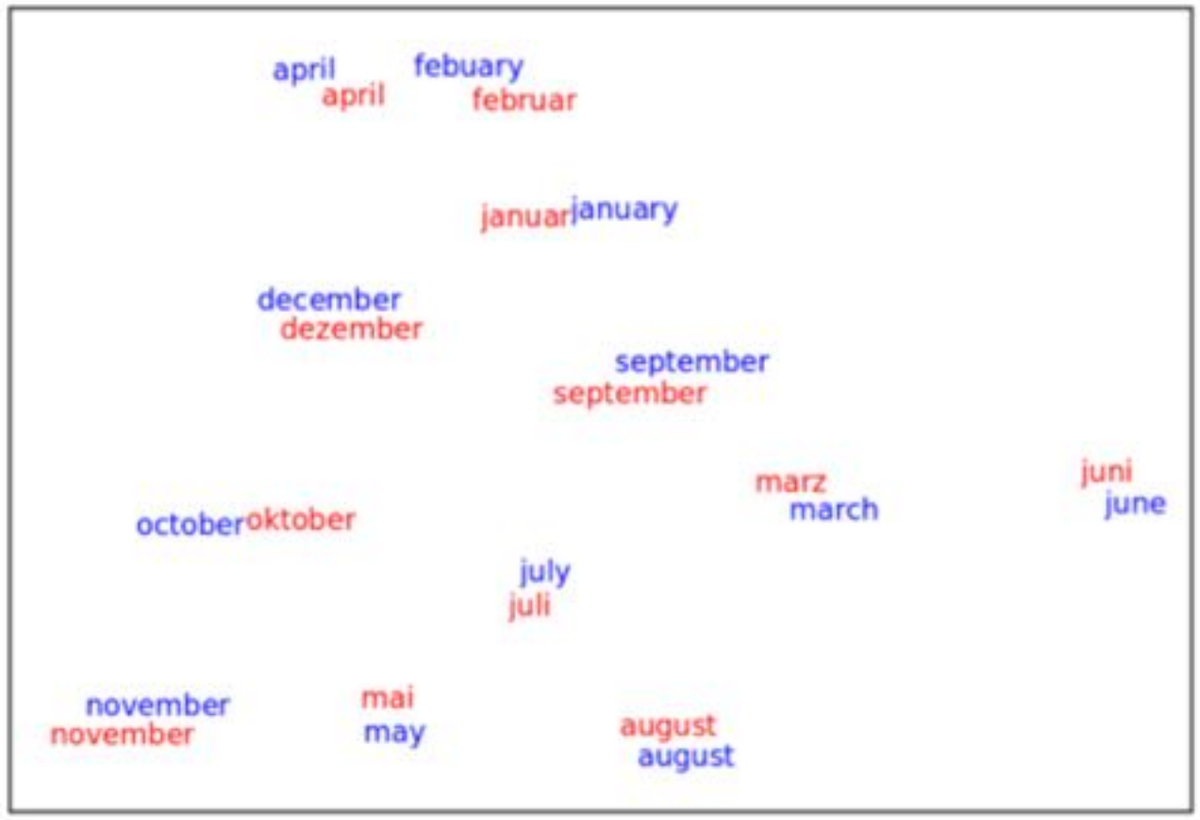




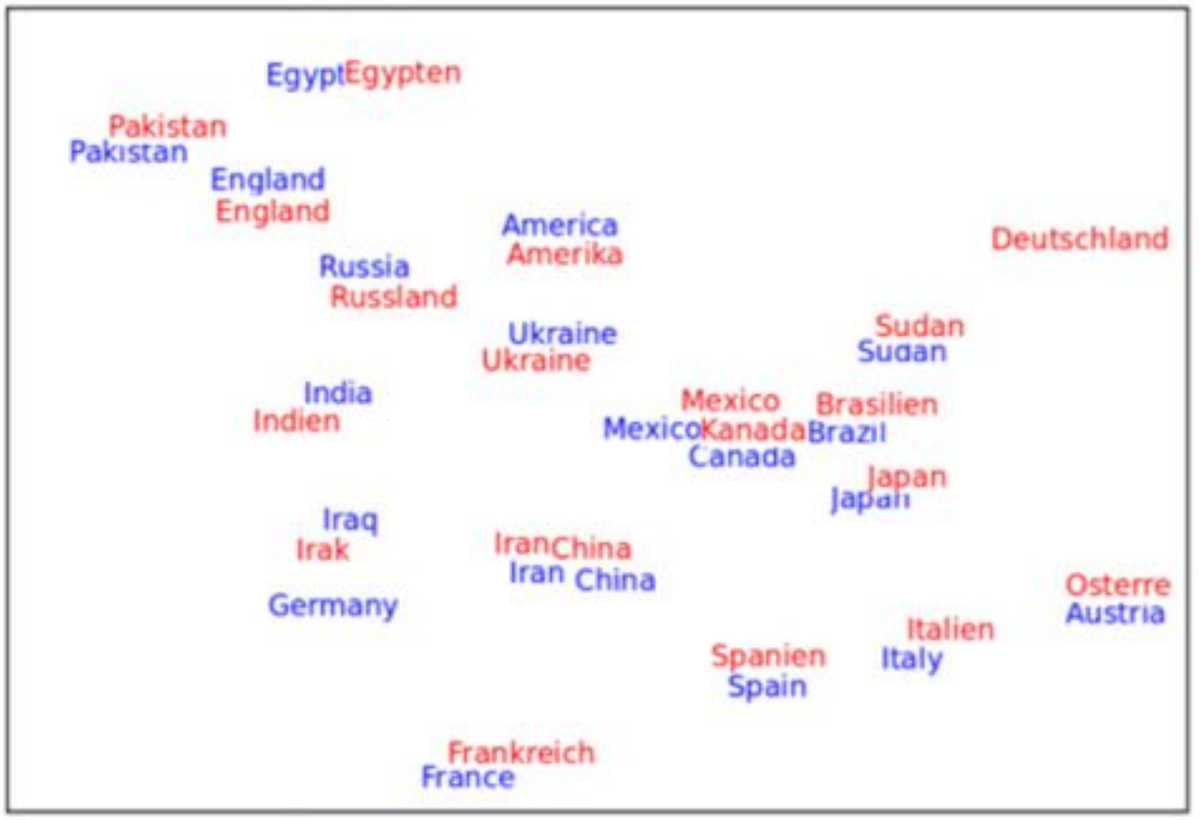


Mikolov, T., Le, V. L., Sutskever, I. (2013) .
Exploiting Similarities among Languages for Machine Translation

Word Embeddings for MT: Kiros (2014)



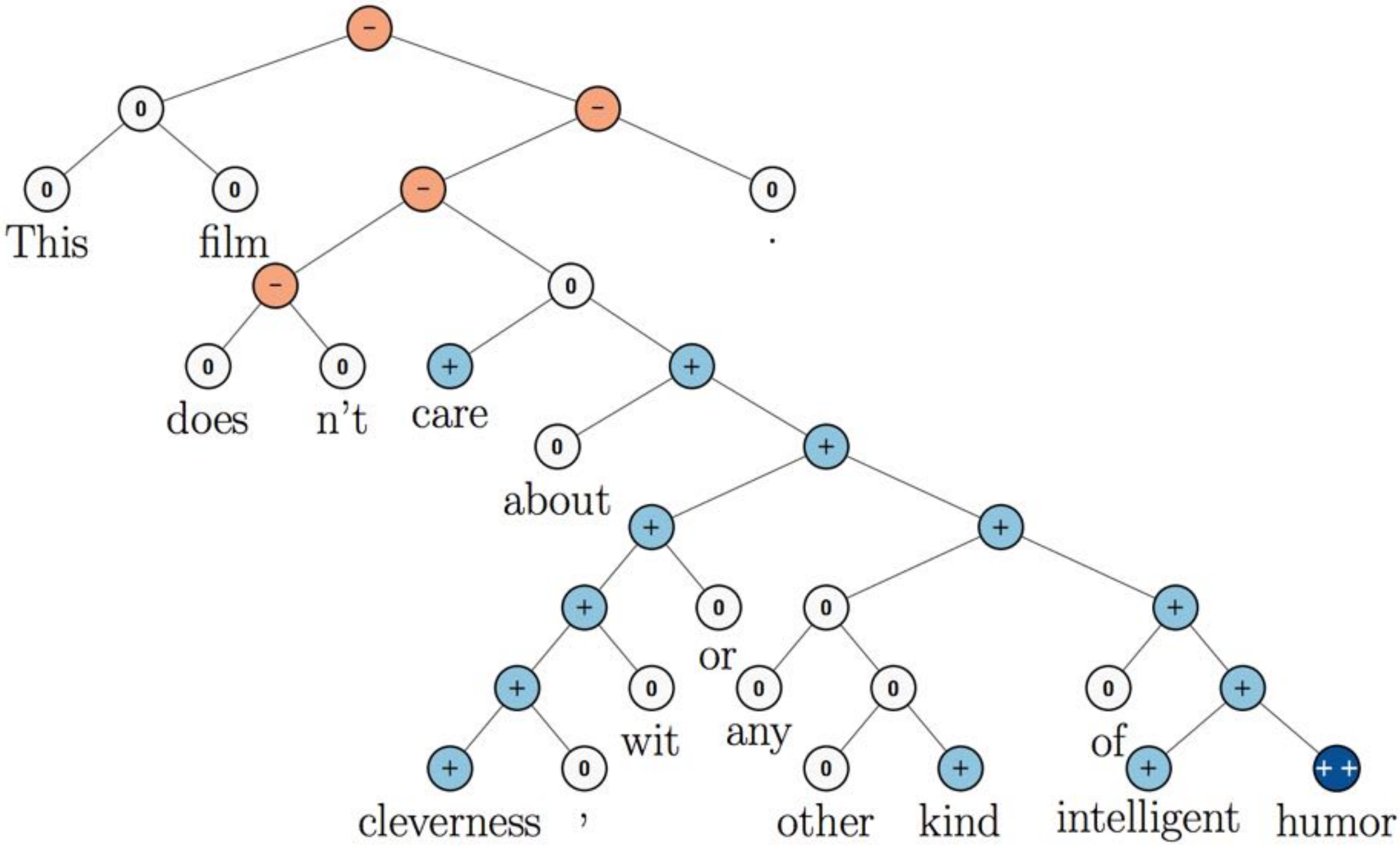
(a) Months



(b) Countries

Kiros, R., Zemel, R. S., Salakhutdinov, R. (2014) .
A Multiplicative Model for Learning Distributed Text-Based Attribute Representations

Recursive Embeddings for Sentiment: Socher (2013)



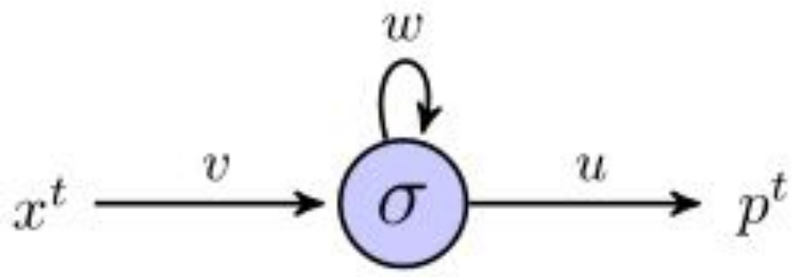
Socher, R., Perelygin, A., Wu, J., Chuang, J.,Manning, C., Ng, A., Potts, C. (2013)
Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank.

code & demo: http://nlp.stanford.edu/sentiment/index.html

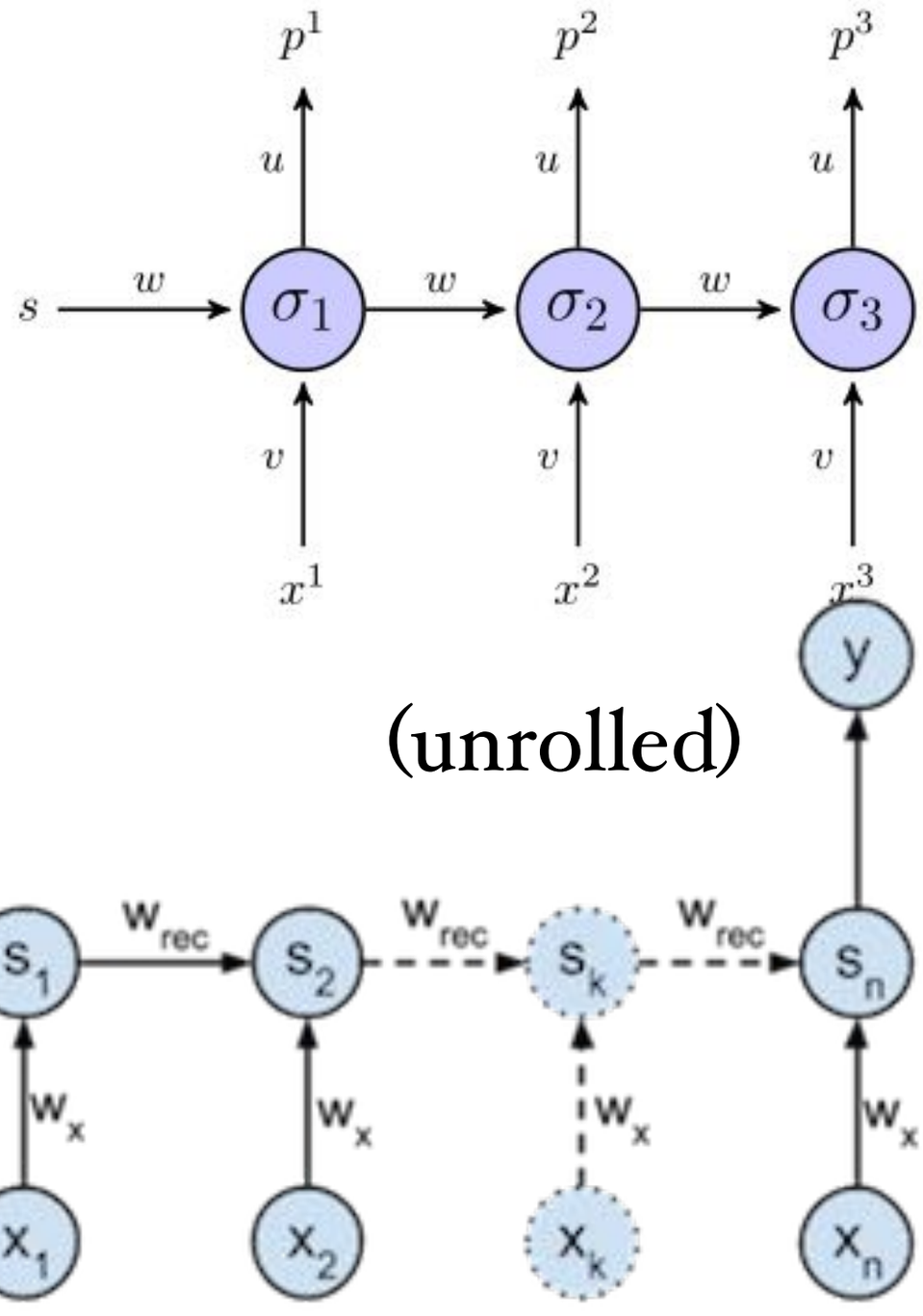
INSERT COFFEE
TO CONTINUE

RNN

- RNN



=

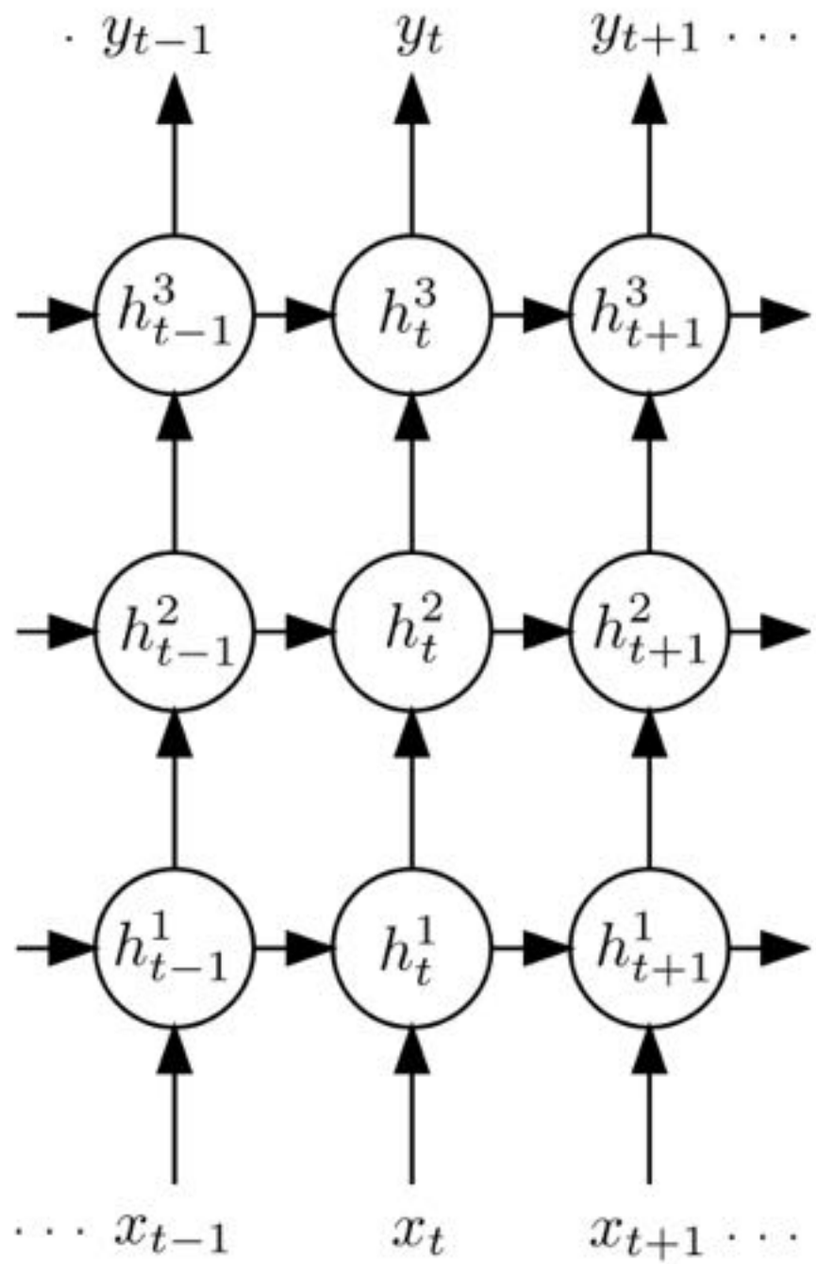


(unrolled)

>Word Embeddings: Sentiment Analysis

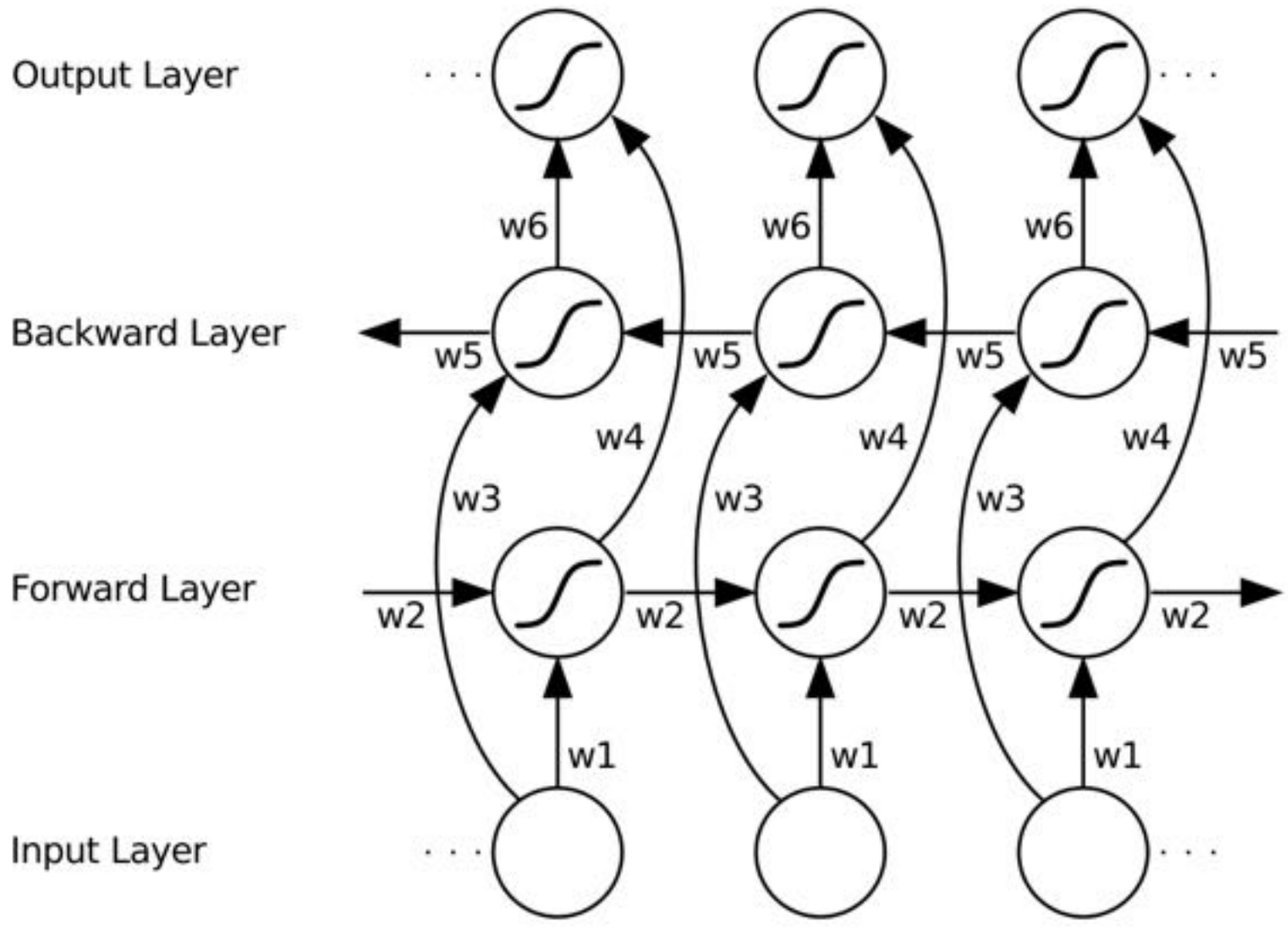
- RNN

$$h_t = \mathcal{H}\left(W_{ih}x_t + W_{hh}h_{t-1} + b_h\right)$$
$$y_t = W_{ho}h_t + b_o$$



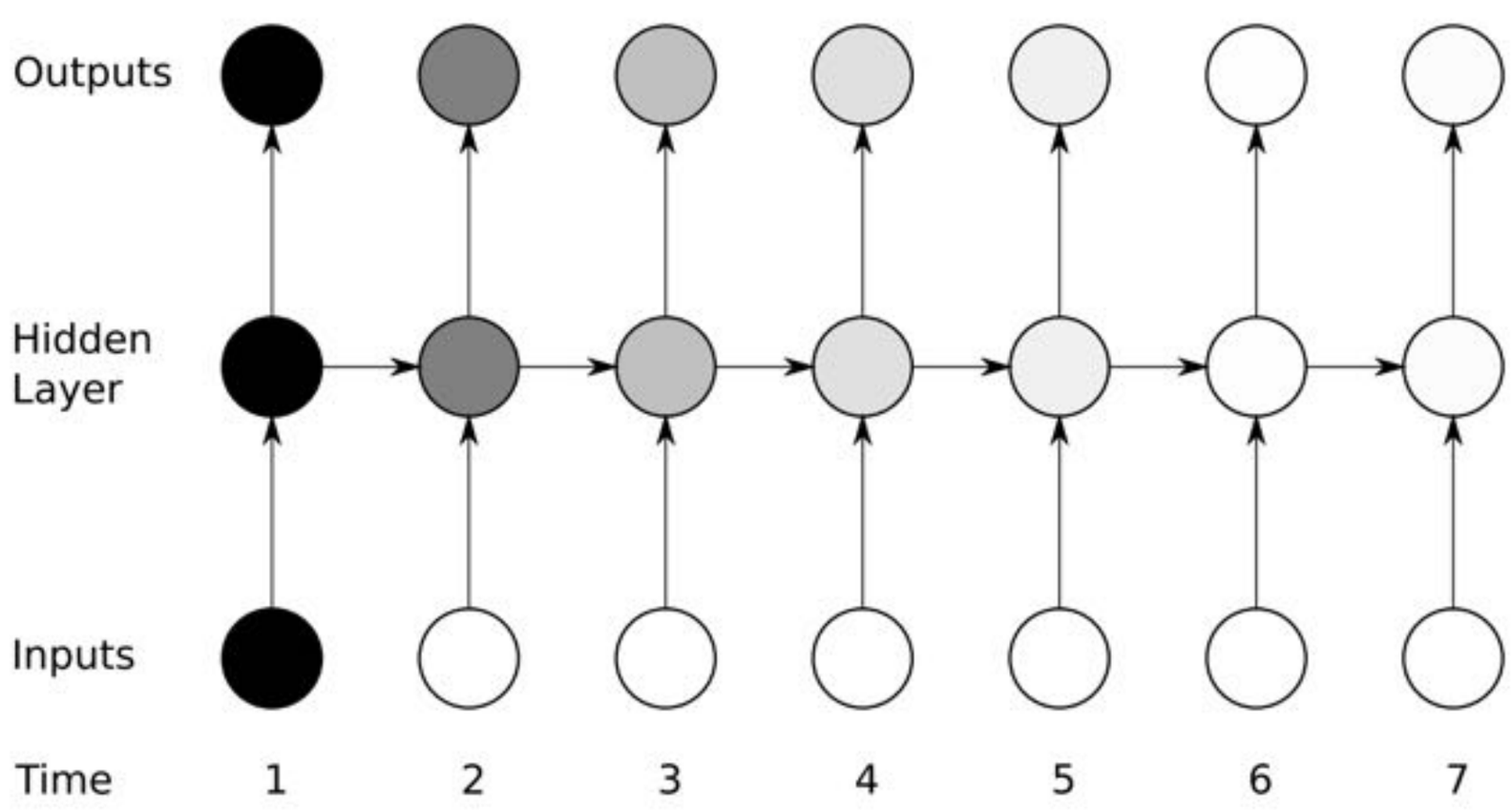
>Word Embeddings: Sentiment Analysis

- Bidirectional RNN



>Word Embeddings: Sentiment Analysis

- Vanishing gradient problem



>Word Embeddings: Sentiment Analysis

- LSTM

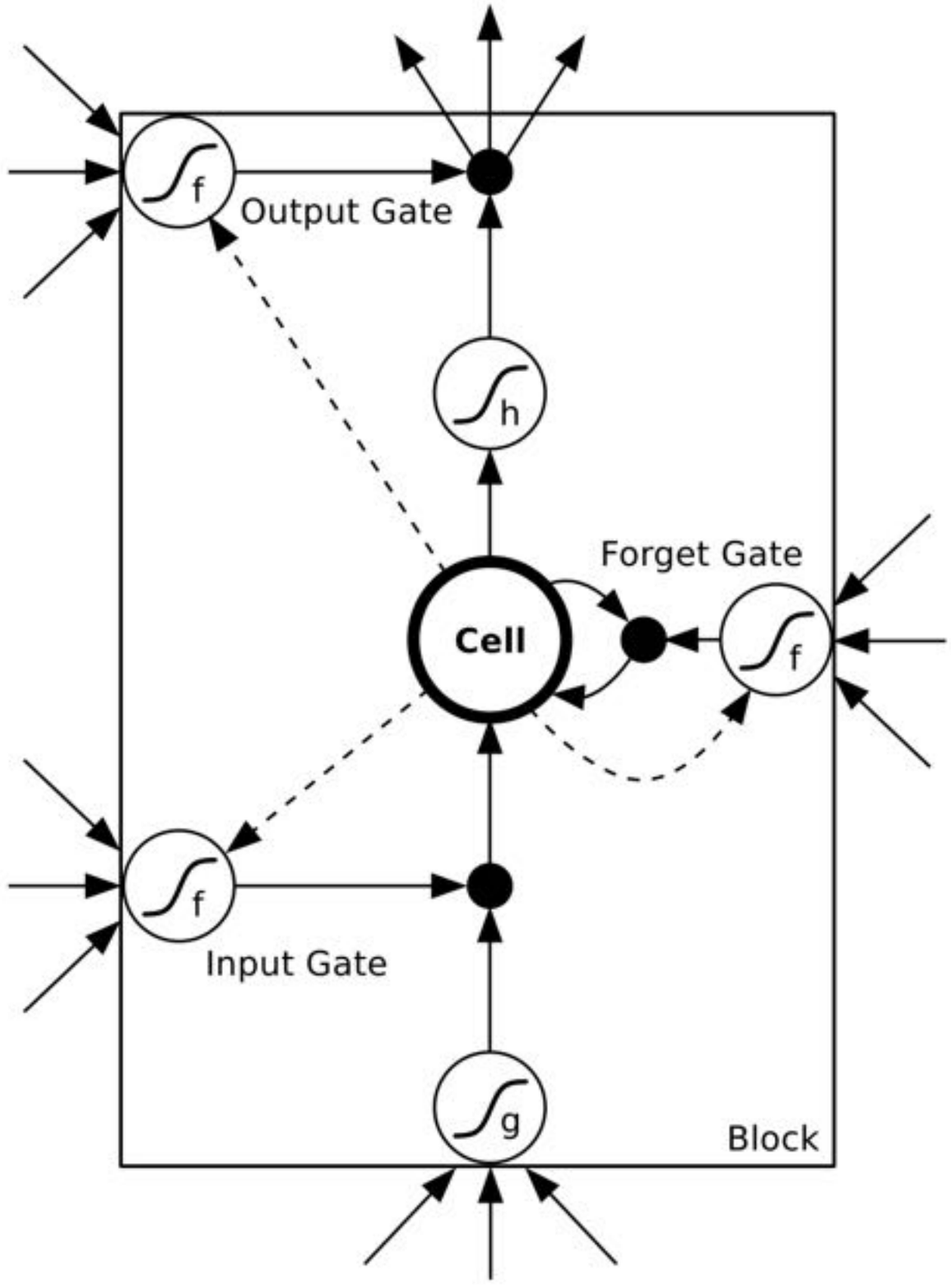
$$i_t = \delta(W^i x_t + R^i h_{t-1} + p^i \odot c_{t-1} + b^i)$$

$$f_t = \delta(W^f x_t + R^f h_{t-1} + p^f \odot c_{t-1} + b^f)$$

$$c_t = f_t \odot c_{t-1} \odot g(W^c x_t + R^c h_{t-1} + b^c)$$

$$o_t = \delta(W^o x_t + R^o h_{t-1} + p^i \odot c_{t-1} + b^i)$$

$$i_t = \delta(W_i x_t + R^i h_{t-1} + p^o \odot c_{t-1} + b^o)$$



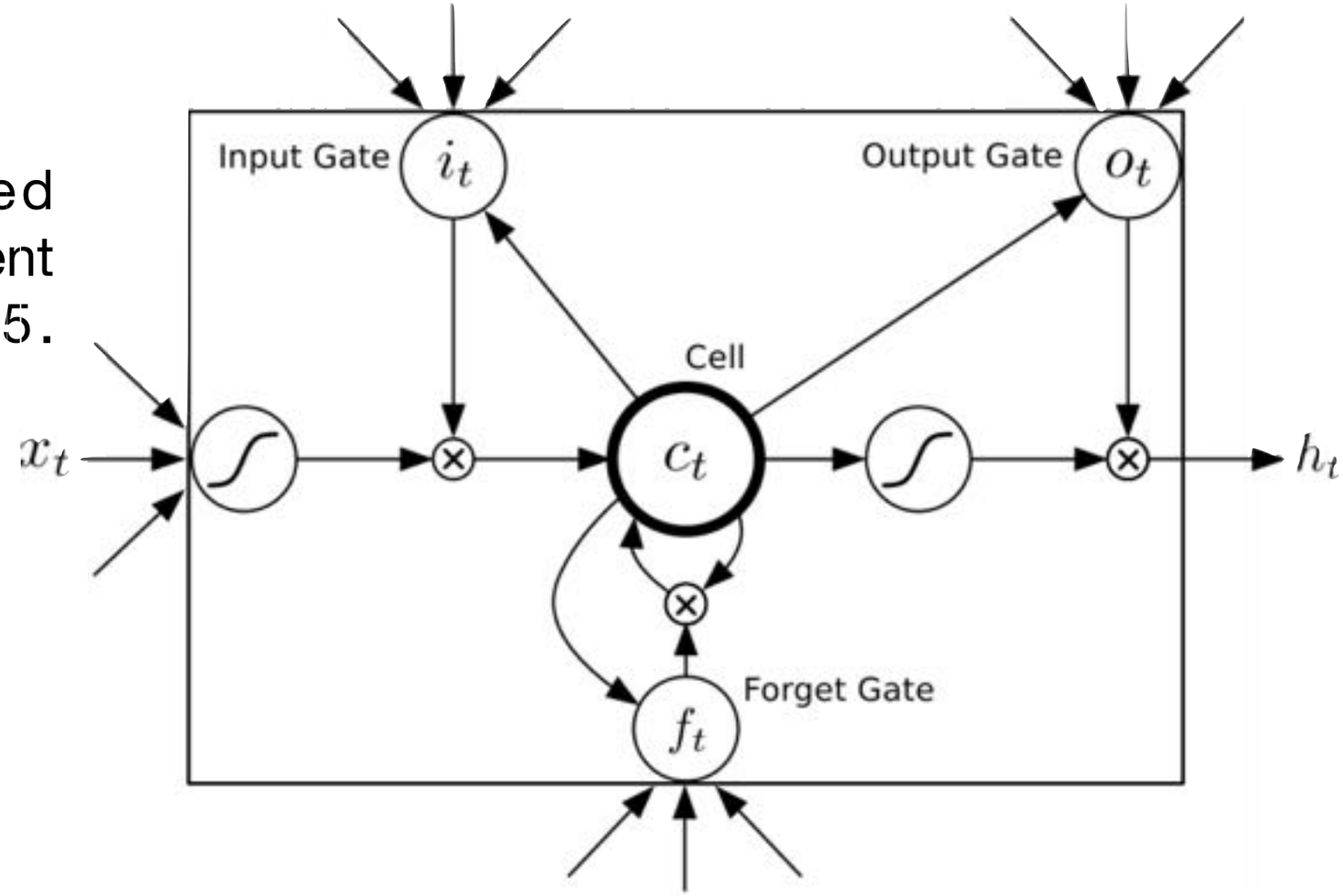
>Word Embeddings: Sentiment Analysis

- LSTM

Introduction of the LSTM model:
- Hochreiter, S., & Schmidhuber, J. (**1997**). Long short-term memory. Neural computation, 9(8), 1735-1780.

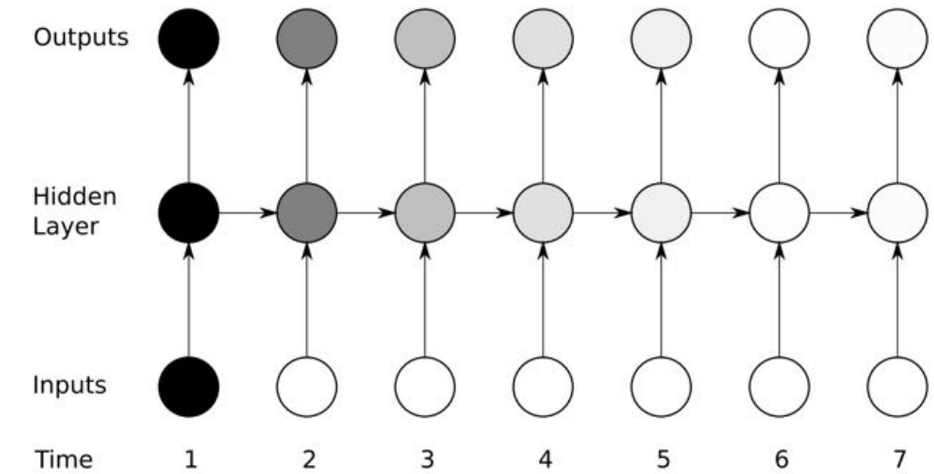
Addition of the forget gate to the LSTM model:
- Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural computation, 12(10), 2451-2471.

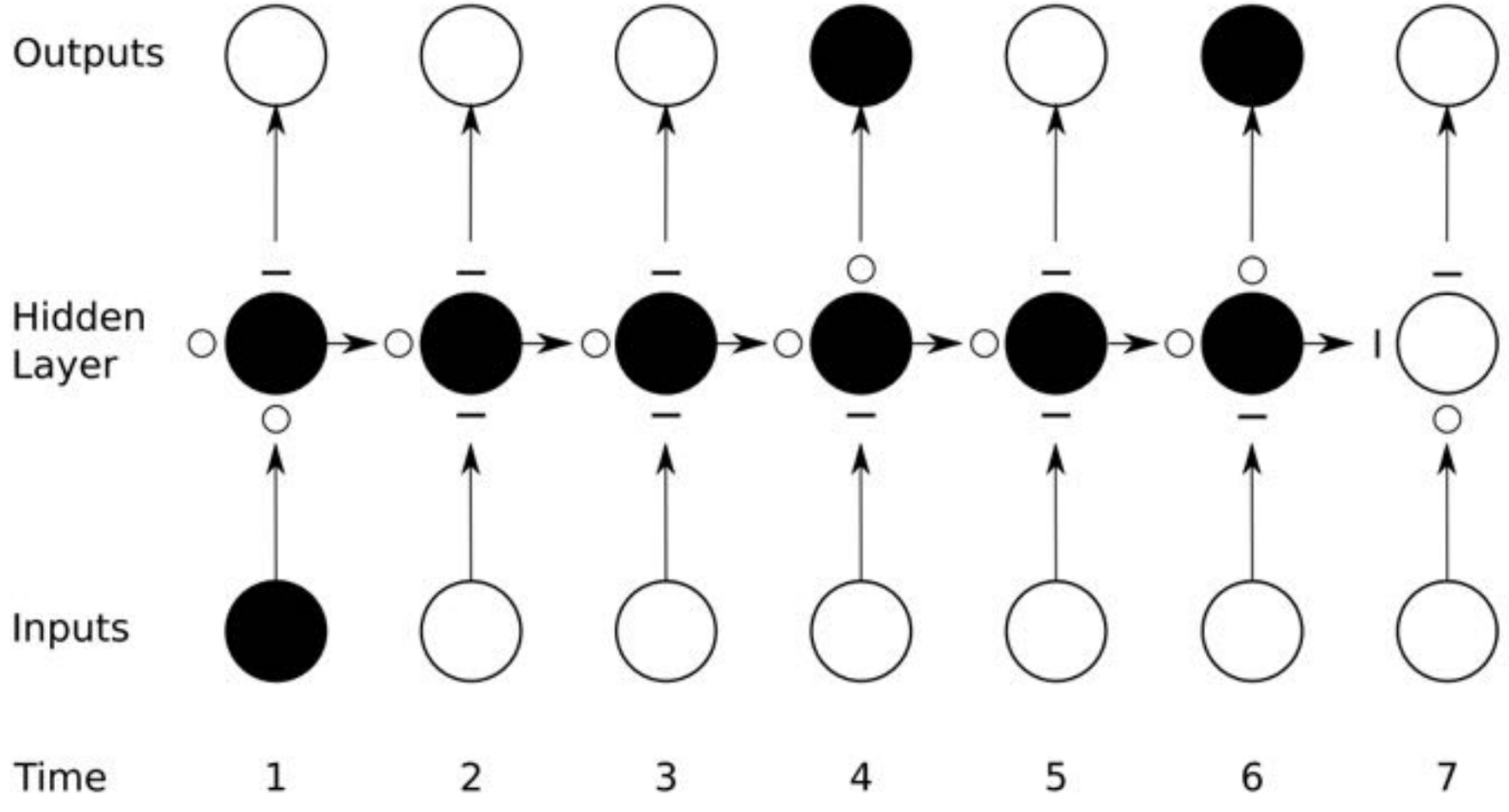
Most cited LSTM paper:
- Graves, Alex. Supervised sequence labelling with recurrent neural networks. Vol. 385. Springer, 2012.



>Word Embeddings: Sentiment Analysis

- Vanishing gradient problem... solved





Wanna be Doing Deep Learning?

Deep Learning with Python

python has a wide range of deep learning-related libraries available

and of course: scikit learn

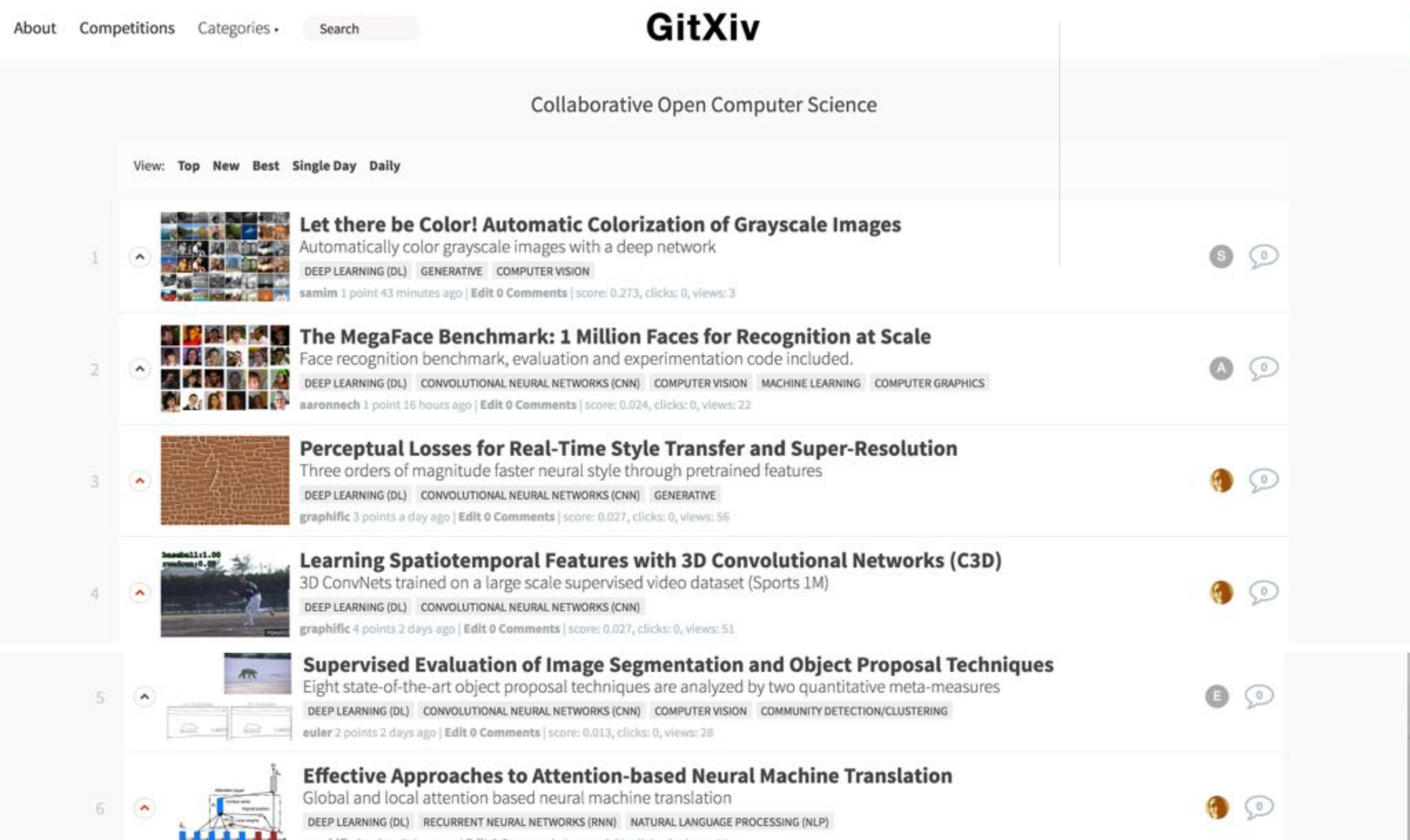
High level

Low level

- Keras — keras.io
- Lasagne — lasagne.readthedocs.org/en/latest
- TensorFlow — tensorflow.org/
- Caffe — caffe.berkeleyvision.org
- theano — deeplearning.net/software/theano

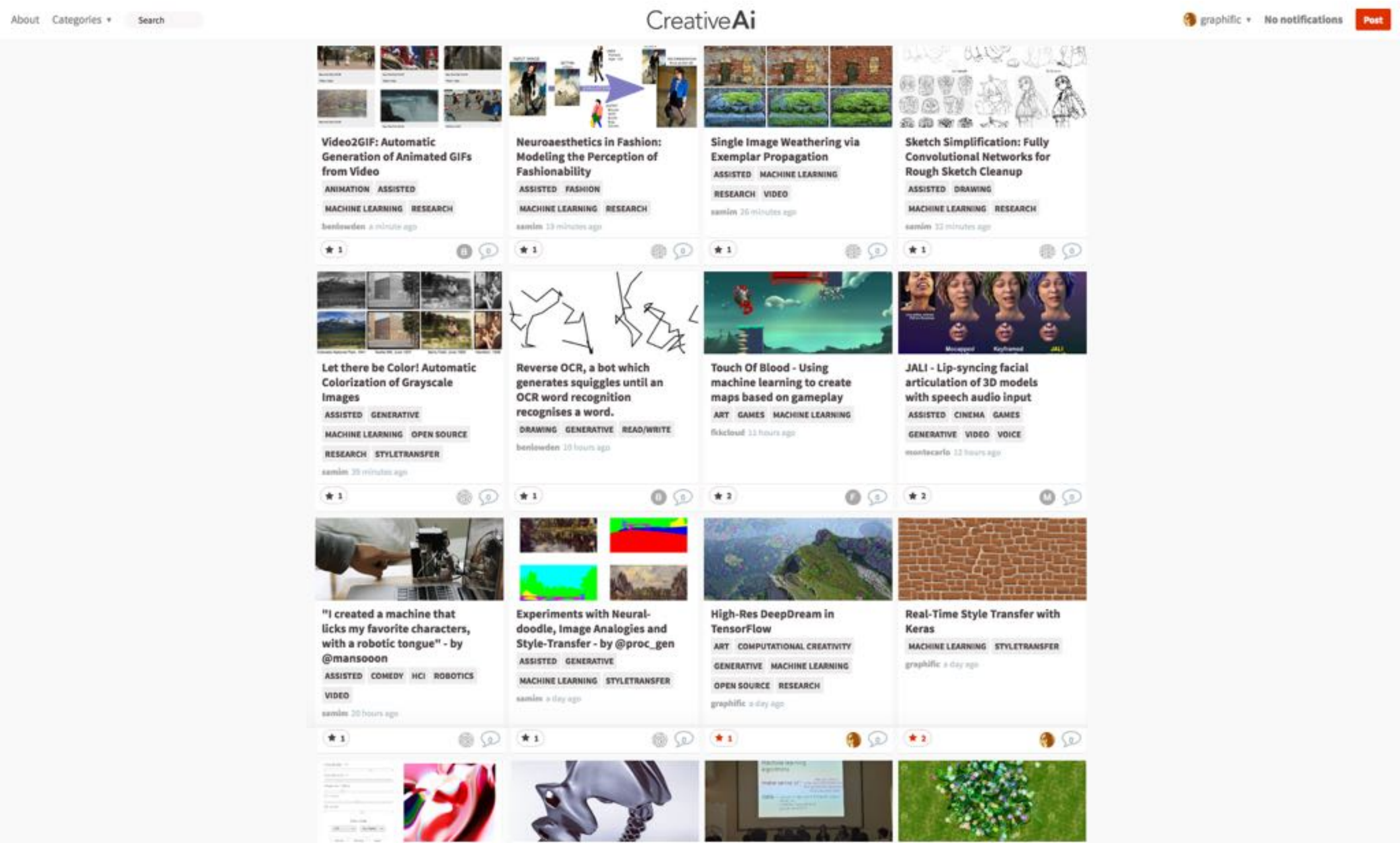
Code & Papers?

http://gitxiv.com/ #GitXiv



Creative AI projects?

http://www.creativeai.net/ #CreativeAI



Questions?

love letters? existential dilemma's? academic questions? gifts?

find me at:

www.csc.kth.se/~roelof/

roelof@kth.se

@graphific

KTH VETENSKAP OCH KONST

ROYAL INSTITUTE OF TECHNOLOGY

Oh, and soon we're looking for Creative AI enthusiasts !

AE ARTIFICIAL EXPERIENCE

- job
- internship
- thesis work

in **AI (Deep Learning) & Creativity**

INSERT COFFEE
TO CONTINUE