Deep Learning and Continuous Representations for Language Processing

Xiaodong He and Scott Wen-tau Yih

Microsoft Research, Redmond, WA

Tutorial presented at IEEE SLT, December 7th, 2014

Tutorial Outline

- Part I: Background
- Part II: Deep learning in spoken language understanding
  - Domain & intent detection using DNN
  - Slot filling using RNN
  - Variants and discussion
- Part III: Learning semantic embedding
  - Semantic embedding: from words to sentences
  - The Deep Structured Semantic Model/Deep Semantic Similarity Model (DSSM)
  - DSSM in practice: Information Retrieval, Auto image captioning
- Part IV: Natural Language Understanding
  - Continuous Word Representations & Lexical Semantics
  - Knowledge Base Embedding
  - Semantic Parsing & Question Answering
- Part V: Conclusion

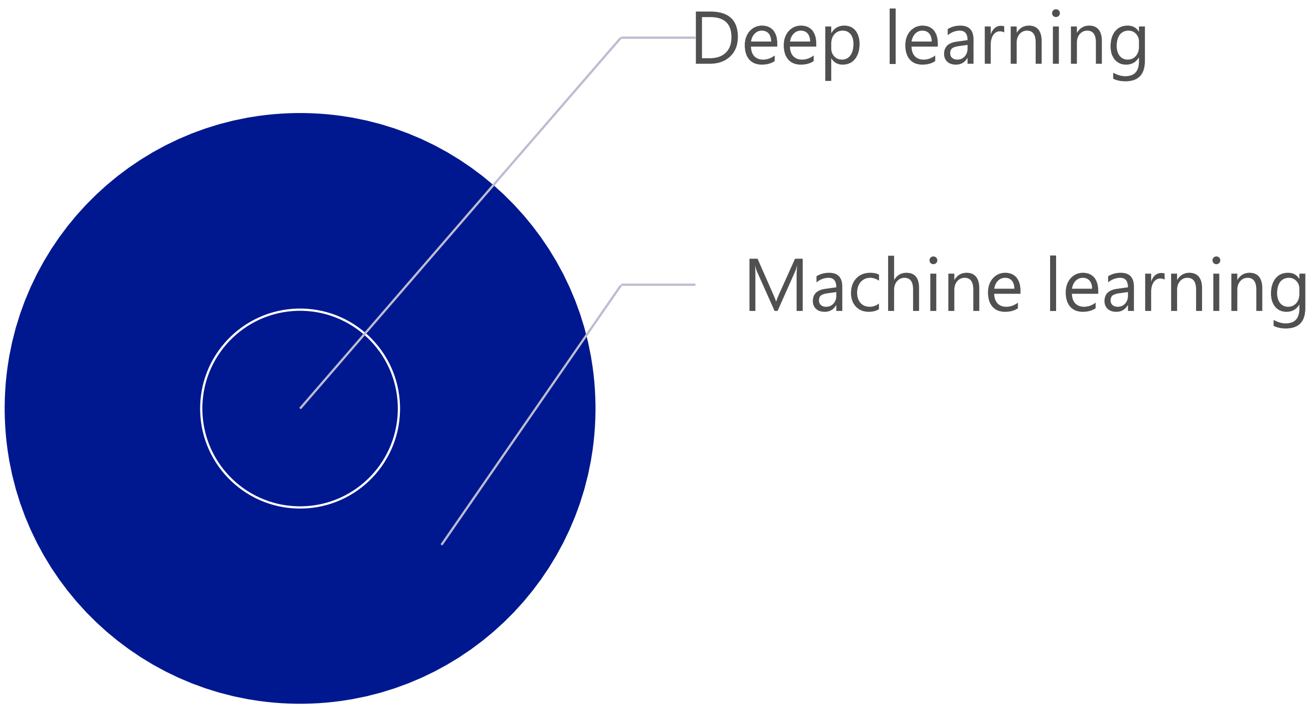
Part I

Background

Background for deep learning

Machine learning

Data → Statistics → Programs



The Universal Translator … *comes true!*

Deep learning technology enabled speech-to-speech translation

The New York Times

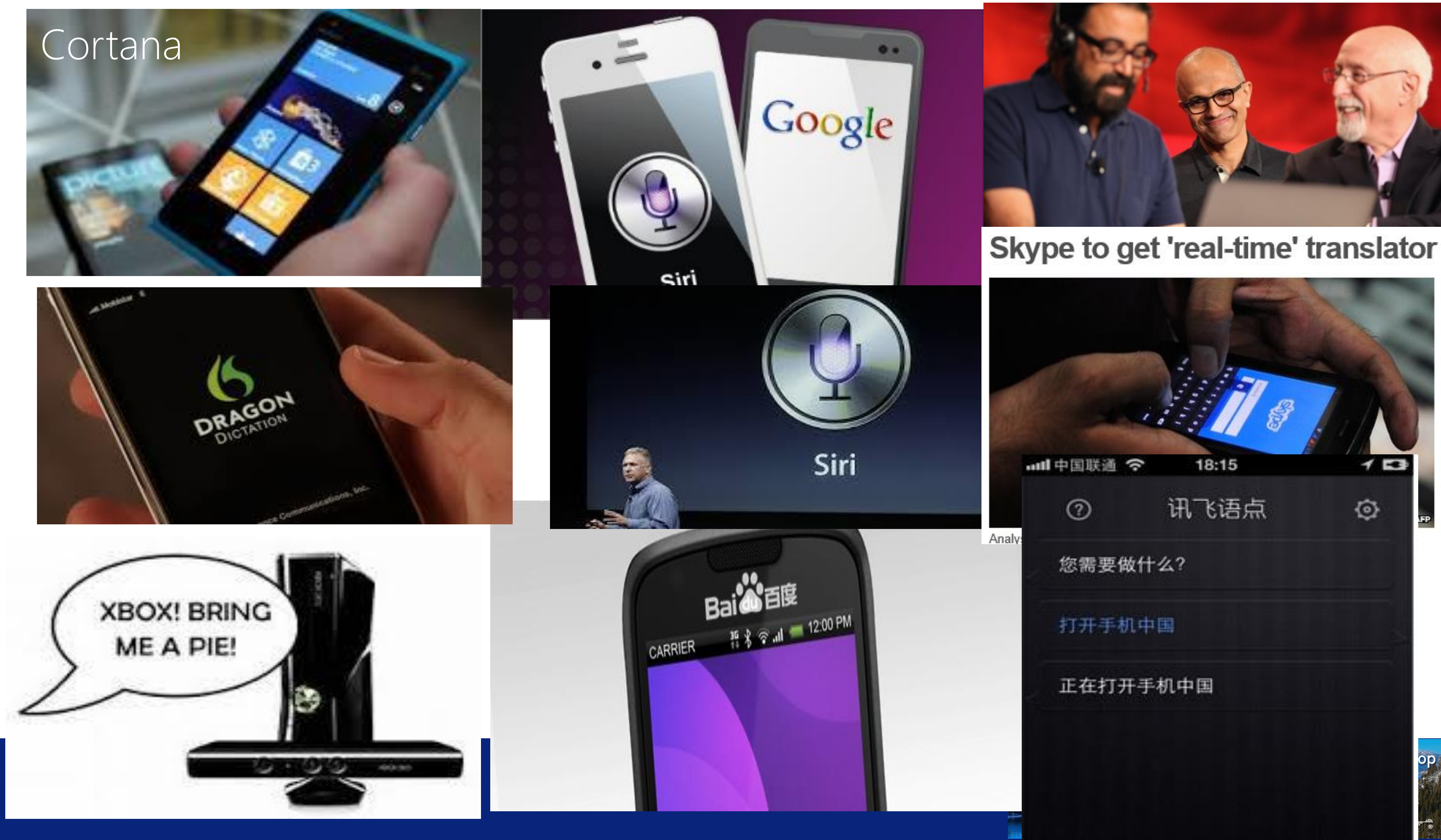
Scientists See Promise in Deep-Learning Programs

John Markoff
November 23, 2012

Rick Rashid in **Tianjin, China**, October, 25, 2012

A voice recognition program translated a speech given by Richard F. Rashid, Microsoft's top scientist, into Mandarin Chinese.

Impact of deep learning in speech technology



MIT Technology Review

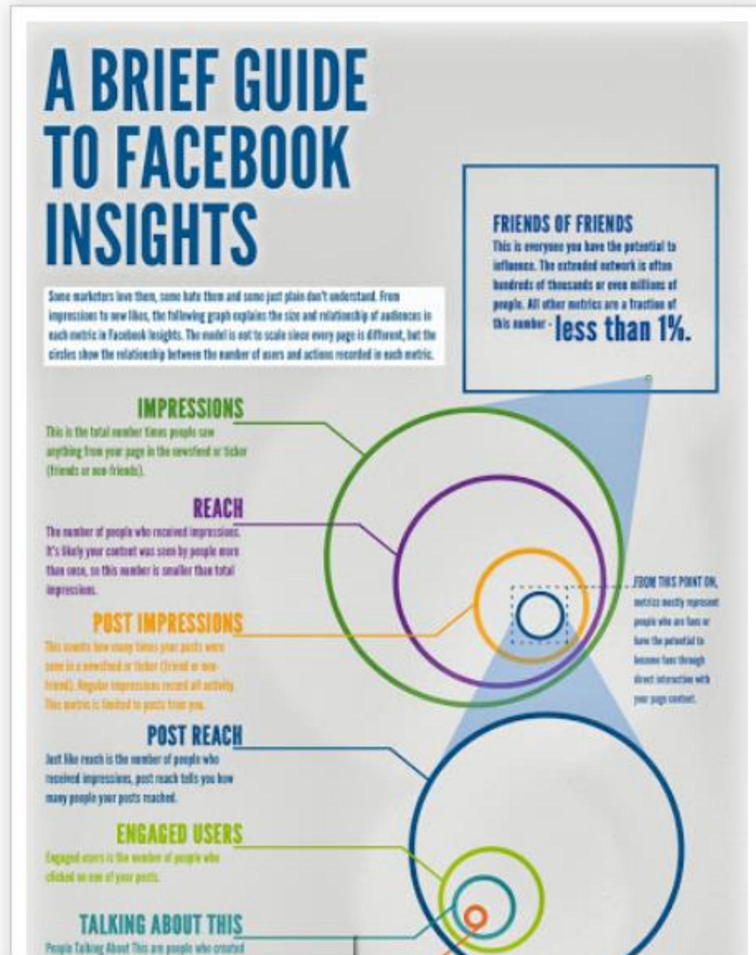
Facebook Launches Advanced AI Effort to Find Meaning in Your Posts

A technique called deep learning could help Facebook understand its users and their data better.

September 20, 2013

By Tom Simonite on September 20, 2013

……Facebook's foray into deep learning sees it following its competitors Google and Microsoft, which have used the approach to impressive effect in the past year. Google has hired and acquired leading talent in the field (see "10 Breakthrough Technologies 2013: Deep Learning"), and last year created software that taught itself to recognize cats and other objects by reviewing stills from YouTube videos. The underlying deep learning technology was later used to slash the error rate of Google's voice recognition services (see "Google's Virtual Brain Goes to Work")….Researchers at Microsoft have used deep learning to build a system that translates speech from English to Mandarin Chinese in real time (see "Microsoft Brings Star Trek's Voice Translator to Life"). Chinese Web giant Baidu also recently established a Silicon Valley research lab to work on deep learning.



MIT Technology Review

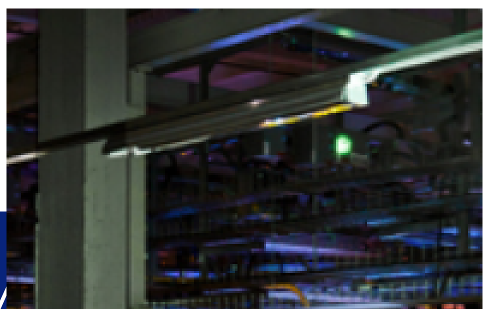
BUSINESS NEWS

8 COMMENTS

Is Google Cornering the Market on Deep Learning?

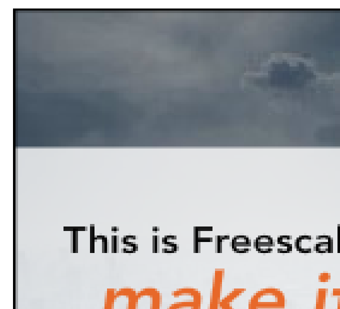
A cutting-edge corner of science is being wooed by Silicon Valley, to the dismay of some academics.

By Antonio Regalado on January 29, 2014



How much are a dozen deep-learning researchers worth? Apparently, more than $400 million.

This week, Google reportedly paid that much to acquire DeepMind Technologies, a startup based in



Geoff Hinton

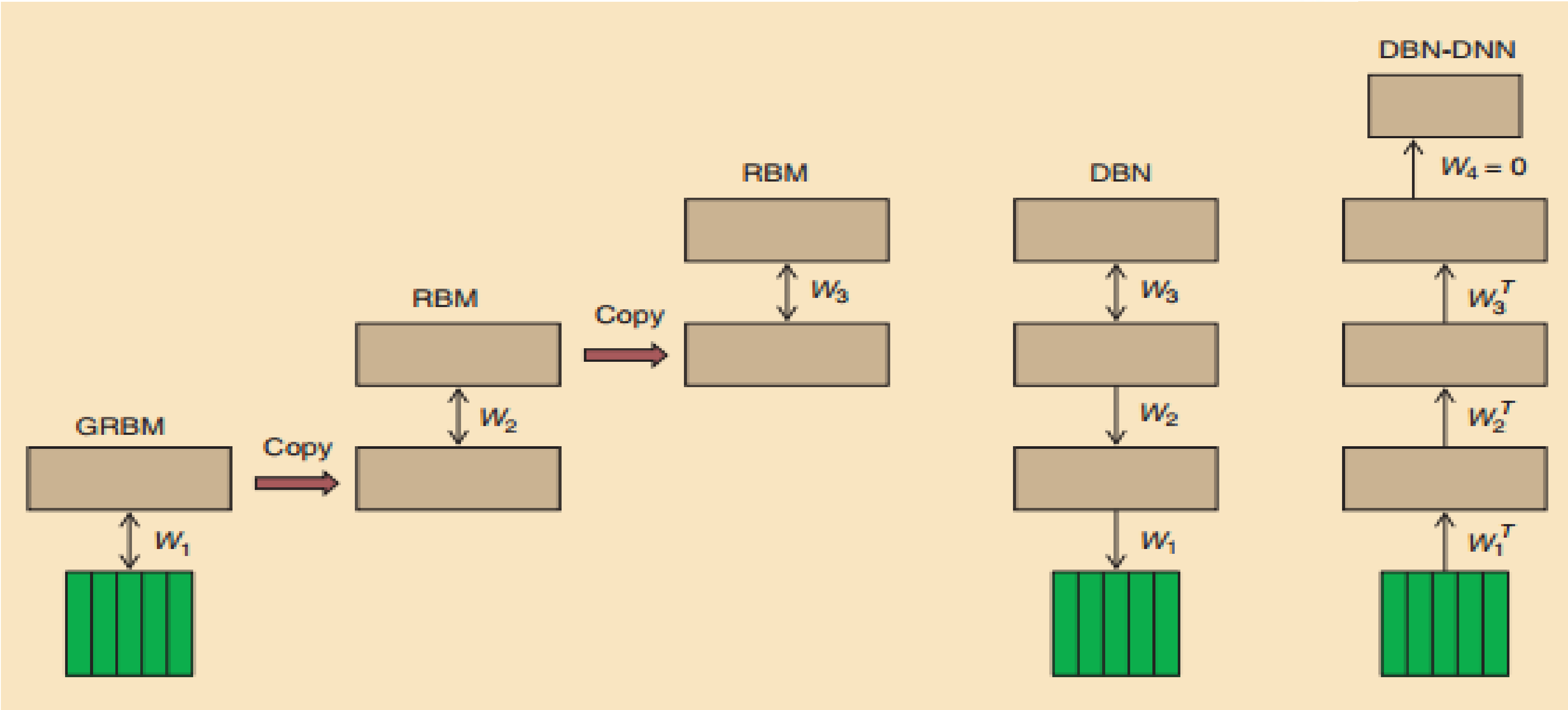
Li Deng

Dong Yu

DNN: (Fully-Connected) Deep Neural Networks

"DNN for acoustic modeling in speech recognition," in *IEEE SPM*, Nov. 2012

Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury

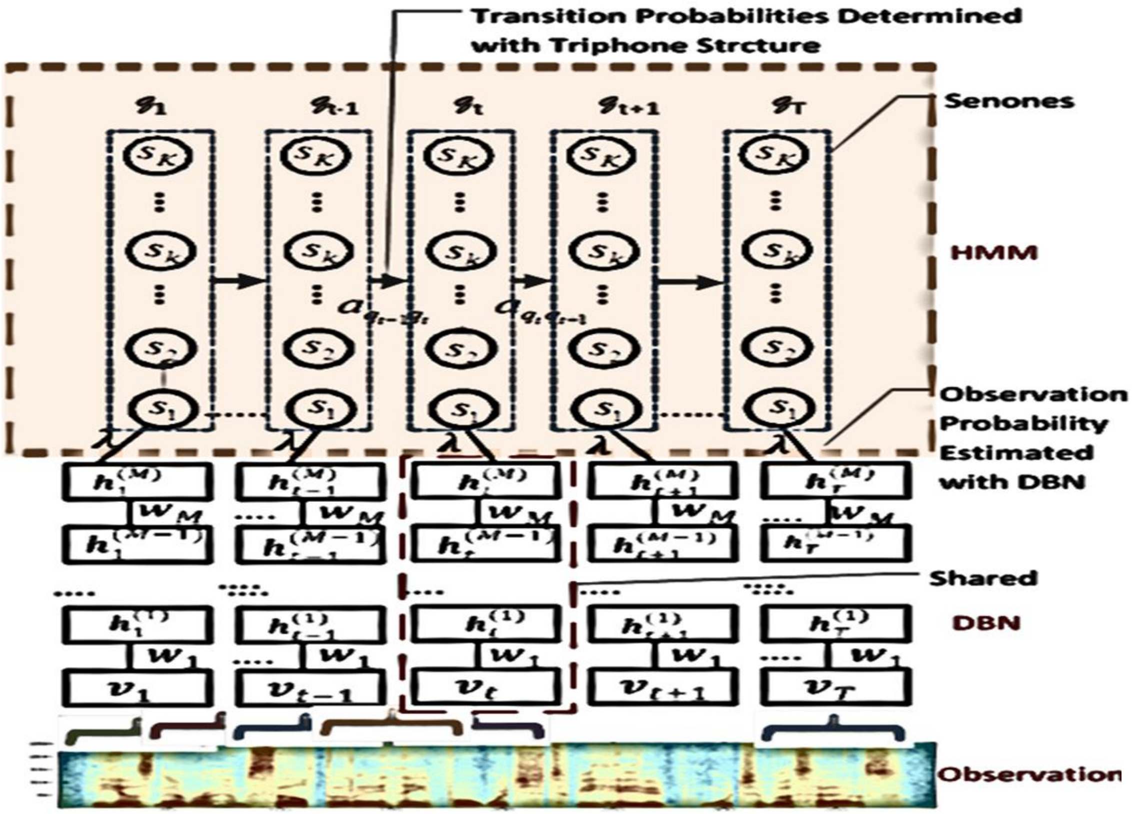


First train a stack of N models each of which has one hidden layer. Each model in the stack treats the hidden variables of the previous model as data.

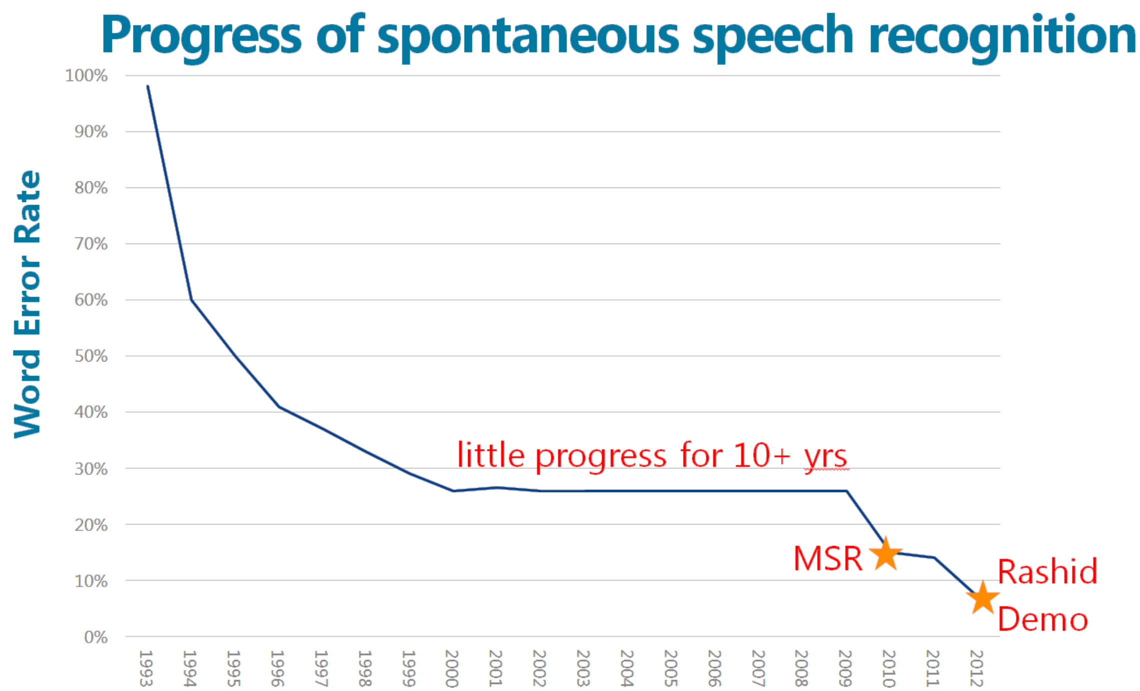
Then compose them into a single Deep Belief Network.

Then add outputs and train the DNN with backprop.

CD-DNN-HMM



Dahl, Yu, Deng, and Acero, "Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition," *IEEE Trans. ASLP*, Jan. 2012

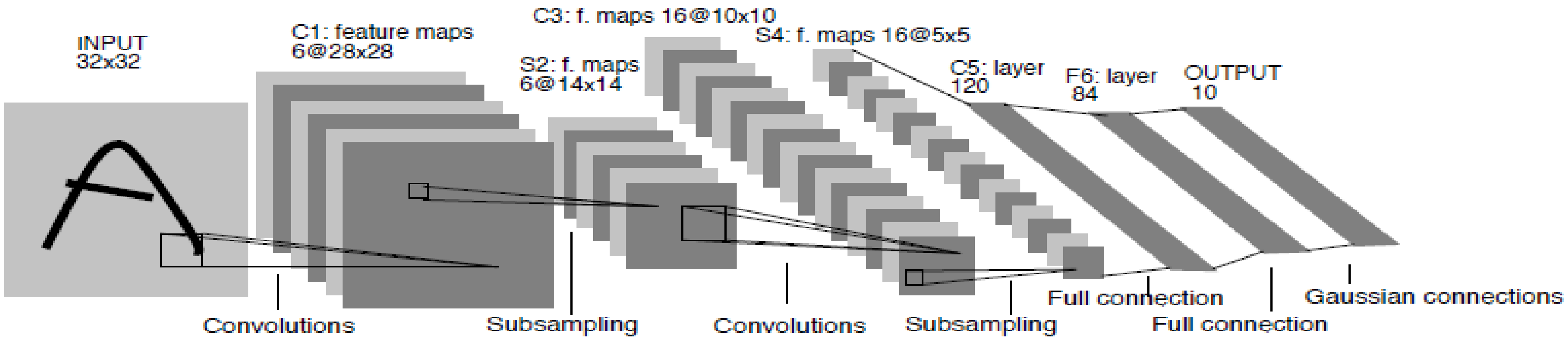


After no improvement for 10+ years by the research community…
…MSR reduced error from **~23%** to **<13%**
(and under 7% for Rick Rashid's S2S demo)!

Deep Convolutional NN for Images

Yann LeCun

CNN: local connections with weight sharing;
pooling for translation invariance

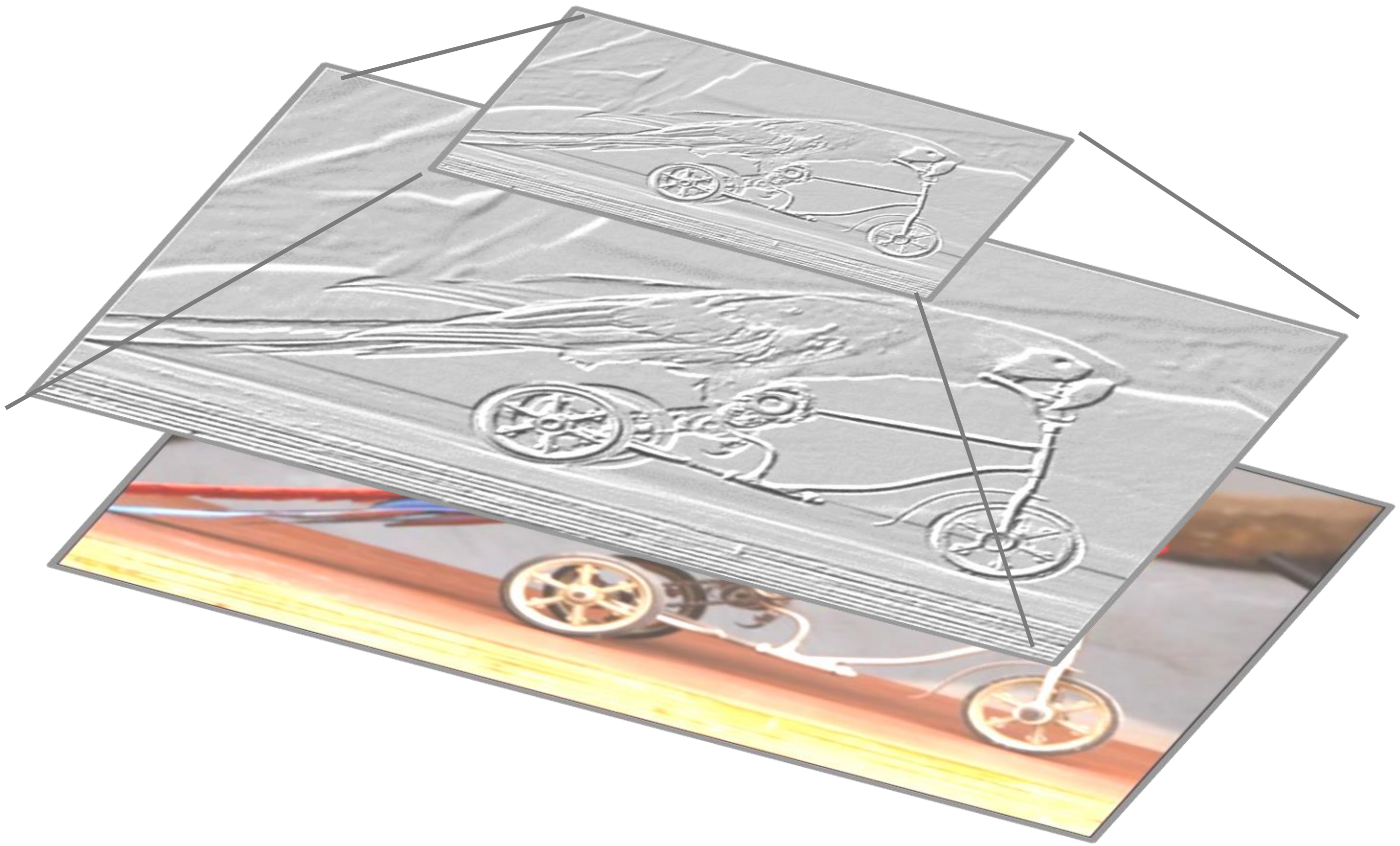


Image

LeCun et al., 1998

Output

A Basic Module of the CNN



Pooling

Convolution

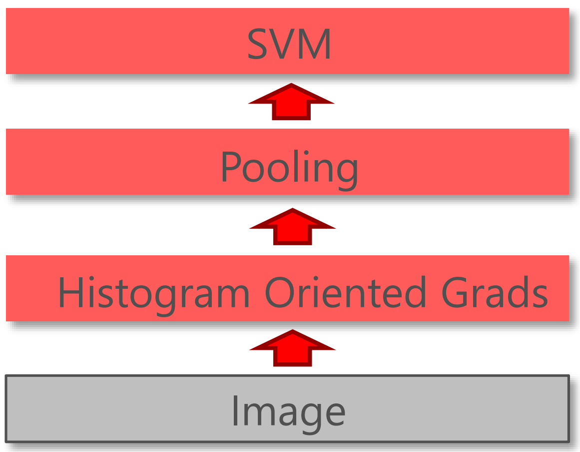
Image

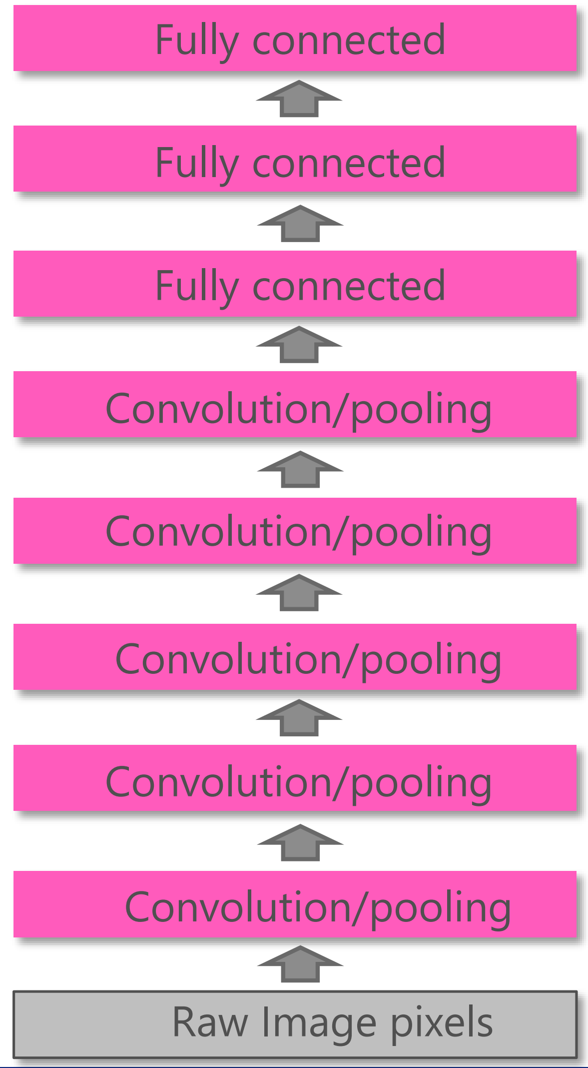
Deep Convolutional NN for Images

2012

A paradigm shift in 2012!

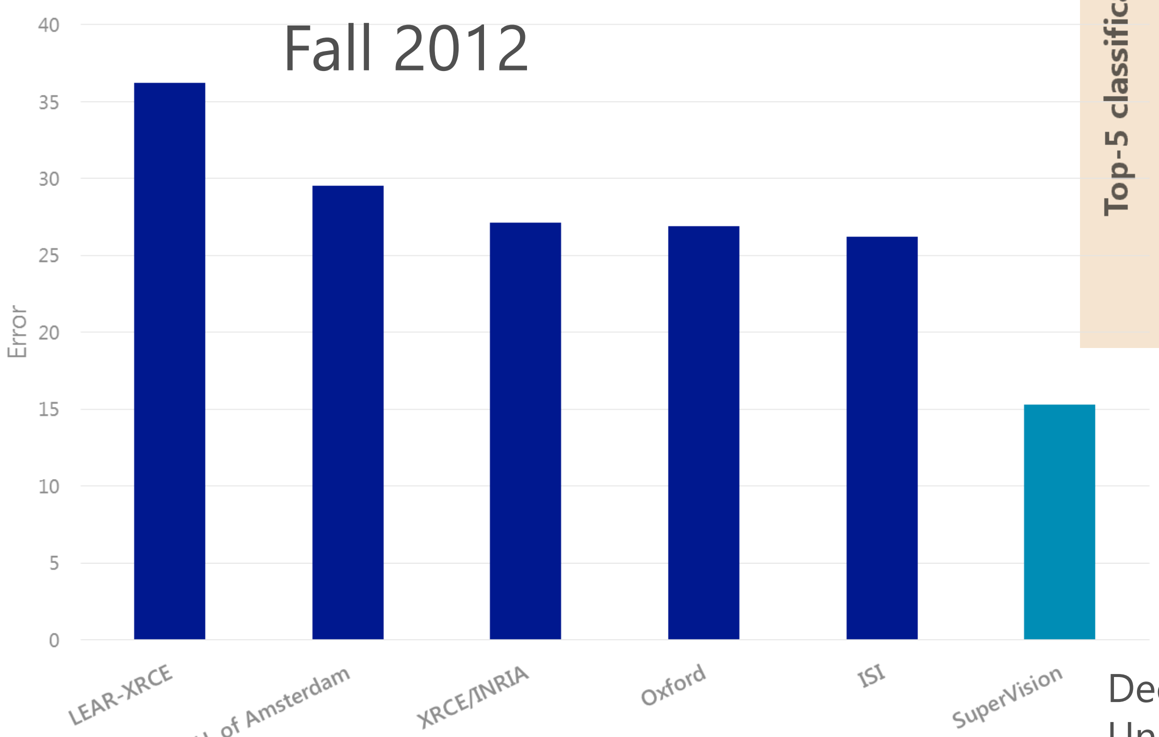
earlier





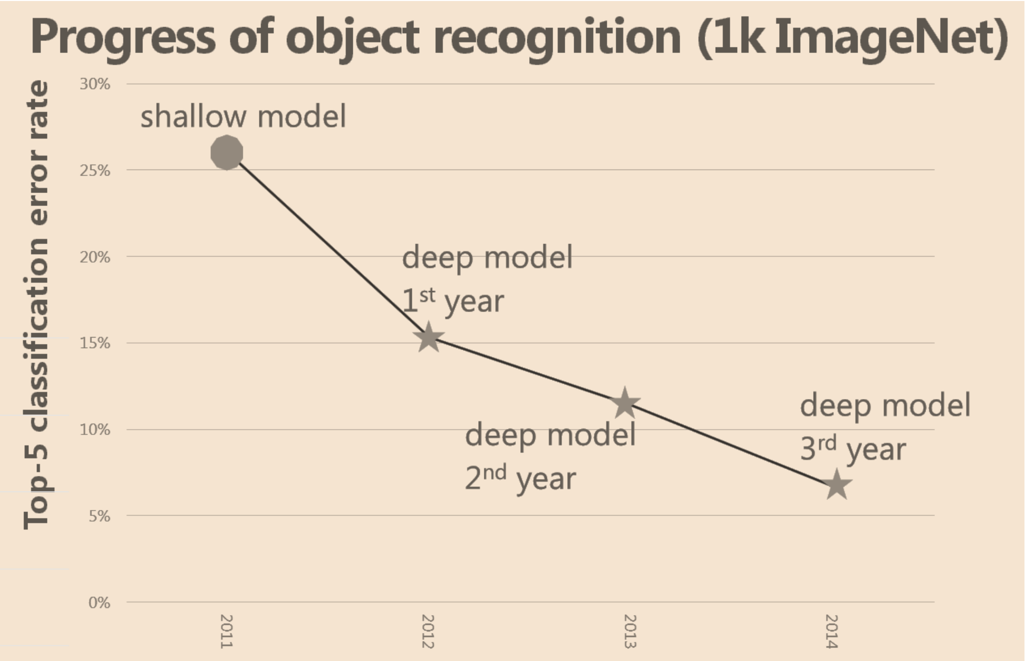
ImageNet 1K Competition

Krizhevsky, Sutskever, Hinton, "ImageNet Classification with Deep Convolutional Neural Networks." *NIPS*, Dec. 2012



Top-5 classification error rate

Deep CNN !!!
Univ. Toronto team

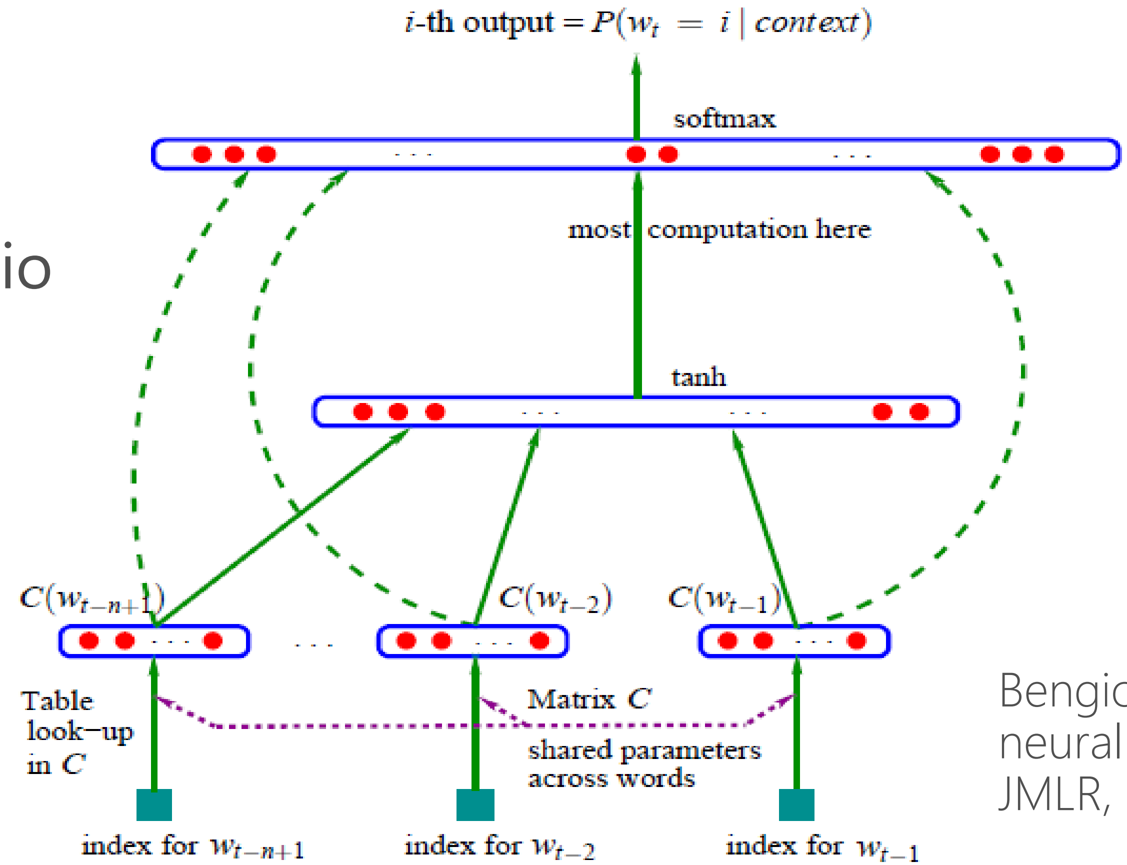


2012 - 2014

Neural network based language model

Yoshua Bengio

LM: predict the next word given the past:
e.g., $p(chases|the\ cat) = ?$, $p(says|the\ cat) = ?$

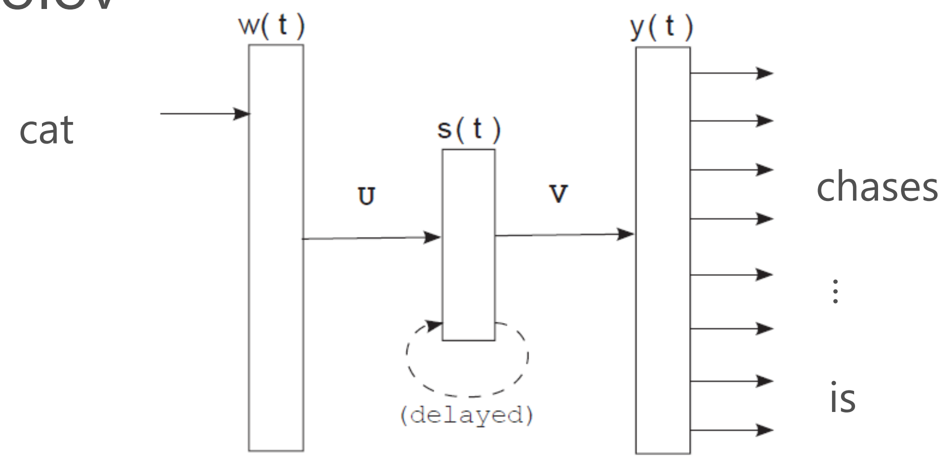


Bengio, Ducharme, Vincent, Jauvin, "A neural probabilistic language model." JMLR, 2003

Recurrent NN based language model

Tomas Mikolov

Mikolov, Karafiat, Burget, Cernocky, Khudanpur, "Recurrent neural network based language model." Interspeech, 2010



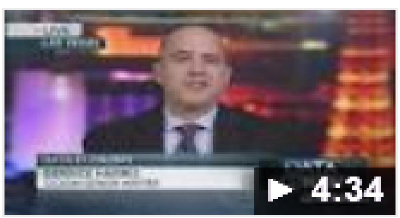
- Large LM perplexity reduction
- Lower ASR WER improvement
- Expensive in learning
- Later turned to FFNN at Google: Word2vec, Skip-gram, etc.
- All UNSUPERVISED

Table 1: *Performance of models on WSJ DEV set when increasing size of training data.*

Model	# words	PPL	WER
KN5 LM	200K	336	16.4
KN5 LM + RNN 90/2	200K	271	15.4
KN5 LM	1M	287	15.1
KN5 LM + RNN 90/2	1M	225	14.0
KN5 LM	6.4M	221	13.5
KN5 LM + RNN 250/5	6.4M	156	11.7

Deep learning demonstrates great success in speech, image, and natural language!

Is **Deep Learning**, the 'holy grail' of big data? - CNBC - Video



video.cnbc.com/gallery/?video=3000192292

Aug 22, 2013

Derrick Harris, GigaOM, explains how "**Deep Learning**" computers are able to process and understand ...

Useful Sites on Deep Learning

- http://www.cs.toronto.edu/~hinton/
- http://ufldl.stanford.edu/wiki/index.php/UFLDL_Recommended_Readings
- http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial (Andrew Ng's group)
- http://deeplearning.net/reading-list/ (Bengio's group)
- http://deeplearning.net/tutorial/
- http://deeplearning.net/deep-learning-research-groups-and-labs/

- Google+ Deep Learning community

18

Interim Summary

- Deep learning sees great impact in Speech, Image, and Text
- Common deep learning architectures
  - DNN (Deep Neural Nets)
  - CNN (Convolutional Neural Nets)
  - RNN (Recurrent Neural Nets)
- The next parts will elaborate on the learning and applications of deep learning/continuous space methods in NLP

19

Part II

Deep learning in spoken language understanding

Deep learning for spoken language processing

The scenarios

- Domain & intent classification
- Semantic slot filling

"Show me flights from Boston to New York today"

Domain: travel

Intent: find_flight

"Show me flights from Boston to New York today"

Semantic slots: City-departure City-arrival Date

Why SLU is difficult?

- Huge variability in the spoken language
  - e.g., both the following two utterances are in the ***Travel*** domain, ***Find_Flight*** intent, and same semantic slots, but are uttered very differently

(1) "I want to fly from San Francisco to New York in a weekend"
(2) "Show me weekend flights from SFO to JFK"

Domain & Intent Classification

- A semantic utterance classification (SUC) problem
  - $\hat{C} = argmax_{\{C\}} P(C|X)$
  - Where
    - $C \in \{C_1, \dots, C_M\}$ belong to one of the M semantic categories (e.g., domain or intent)
    - X is the input utterance

SUC: Common methods

- Common raw features usually include
  - Word n-grams (n=1, 2, 3), e.g., bi-gram,

$$f^{BG}_{c,w_x w_y}(C_r, W_r) = \begin{cases} 1, & \text{if } c = C_r \wedge w_x w_y \in W_r \\ 0, & \text{otherwise.} \end{cases}$$

- Common classifiers
  - Logistic regression

$$P(C|W) = \frac{1}{Z} \sum_i w_i f_i(C, W)$$

  - Boosting, SVM, etc.

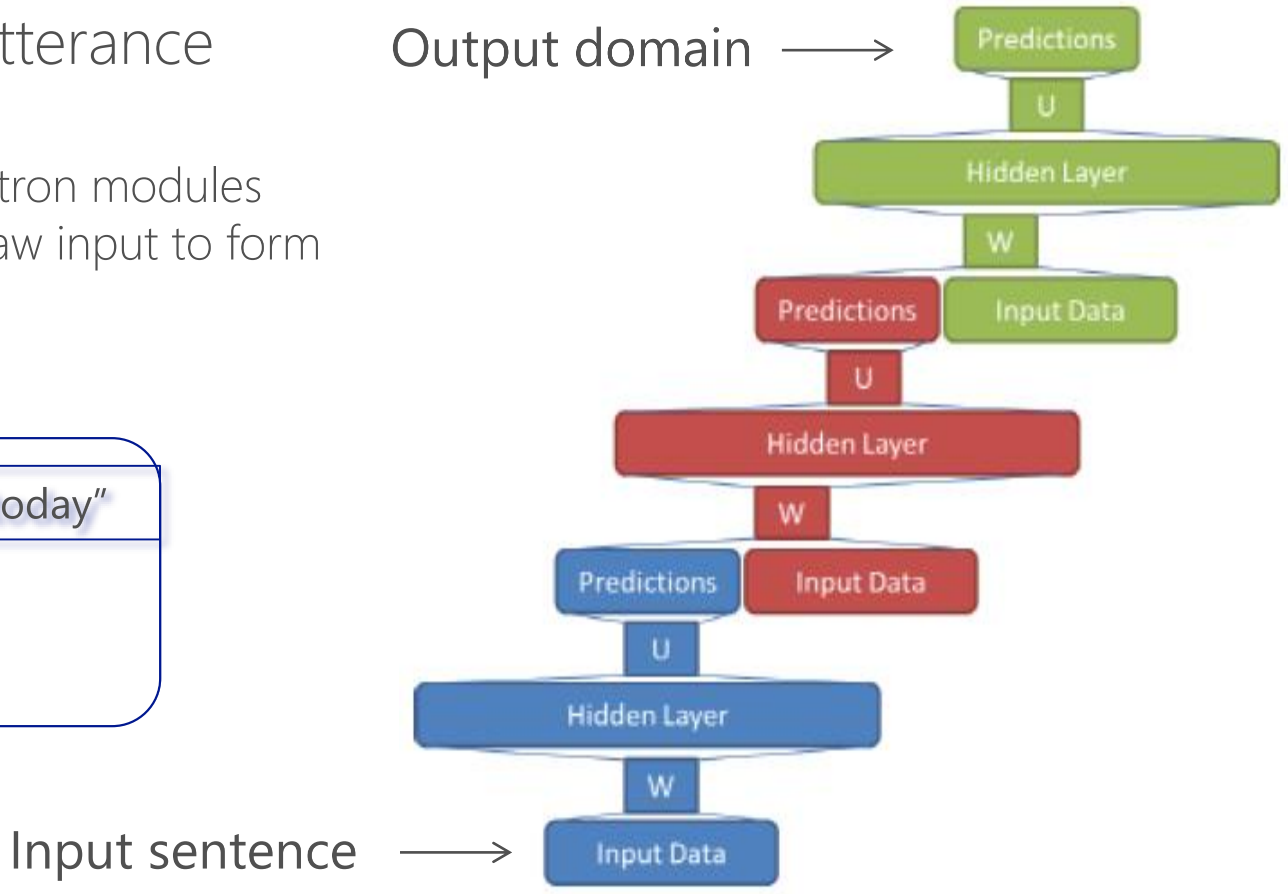
Deep stack net for domain & intent classification

Deep stack net for semantic utterance classification:

1) A stack of a series of 3-layer perceptron modules
2) Output layer is concatenated with raw input to form input layer of the next module

"Show me flights from Boston to New York today"

Domain: travel



[Tur, Deng, Hakkani-Tur, He, 2012; Deng, Tur, He, Hakkani-Tur, 2012]

Domain classification results

Table 2. *Comparisons of the domain classification error rates among the boosting-based baseline system, DCN system, and K-DCN system for a domain classification task. Three types of raw features (lexical, query clicks, and name entities) and four ways of their combinations are used for the evaluation as shown in four rows of the table.*

Feature Sets	Baseline	DCN	K-DCN
lexical features	10.40%	10.09%	**9.52%**
lexical features + Named Entities	9.40%	9.32%	**8.88%**
lexical features + Query clicks	8.50%	7.43%	**5.94%**
lexical features + Query clicks + Named Entities	10.10%	7.26%	**5.89%**

30% error reduction over a boosting-based baseline!

Table 3. *More detailed results of K-DCN in Table 2 with Lexical+QueryClick features. Domain classification error rates (percent) on Train set, Dev set, and Test set as a function of the depth of the K-DCN.*

Depth	Train Err%	Dev Error%	Test Err%
1	9.54	12.90	12.20
2	6.36	10.50	9.99
3	4.12	9.25	8.25
4	1.39	7.00	7.20
5	0.28	6.50	5.94
6	0.26	**6.45**	**5.94**
7	0.26	6.55	6.26
8	0.27	6.60	6.20

Error keeps decreasing until up to six layers are added up

Deng, Tur, He, Hakkani-Tur, Use of kernel deep convex networks and end-to-end learning for spoken language understanding, IEEE-SLT 2012

Semantic slot filling

A example in the Airline Travel Information System (ATIS) corpus

	show	*flights*	*from*	*boston*	*to*	*new*	*york*	*today*
Slots	O	O	O	B-dept	O	B-arr	I-arr	B-date

Slot filling can be viewed as a sequential tagging problem

Slot Filling: Common methods

Conditional random field (CRF)

$$\ell(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_t^{(i)}, y_{t-1}^{(i)}, \mathbf{x}_t^{(i)}) - \sum_{i=1}^{N} \log Z(\mathbf{x}^{(i)}) - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2}.$$

- N: number of training samples
- T: number of words in the sentence i
- K: “observation” functions (feature functions)
- x: input words in the sentence
- y: output tags

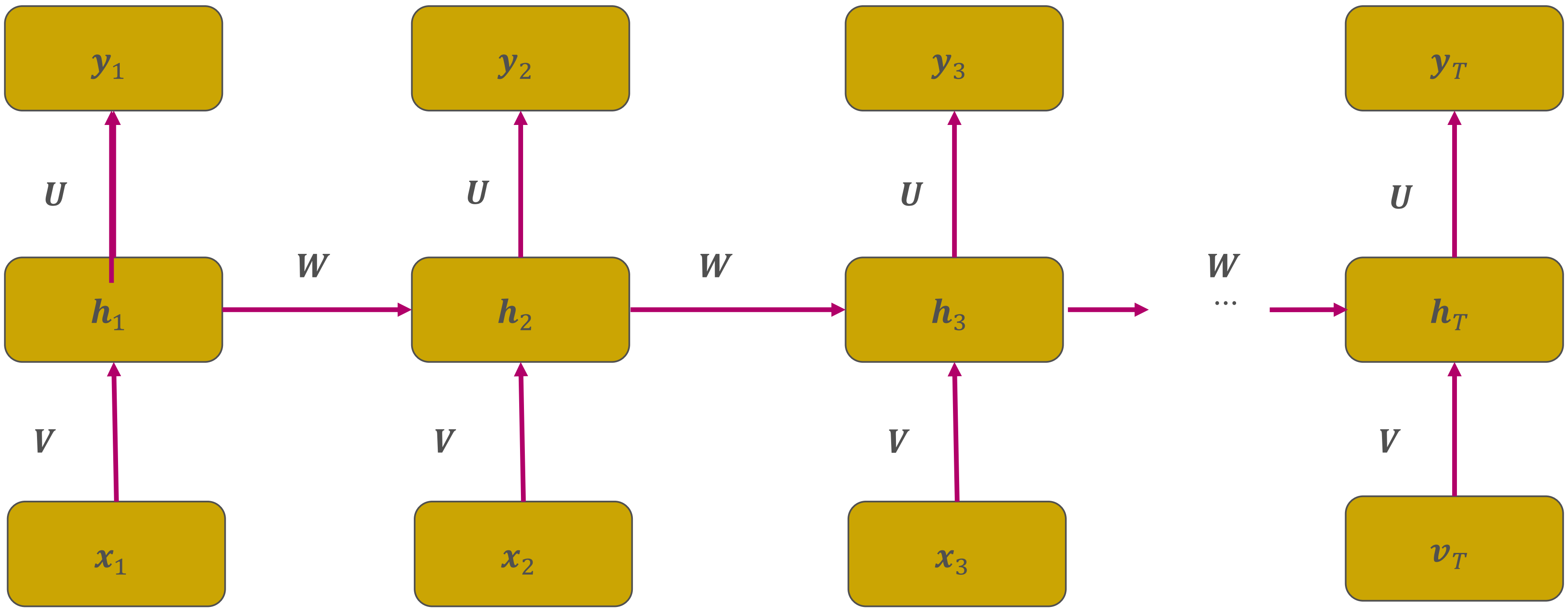
Other variants of CRF exist, e.g., semi-CRF.

Recurrent neural networks for slot filling

h_t is the hidden layer that carries the information from time $0 \sim t$

where x_t: *the input word* , y_t: *the output tag*

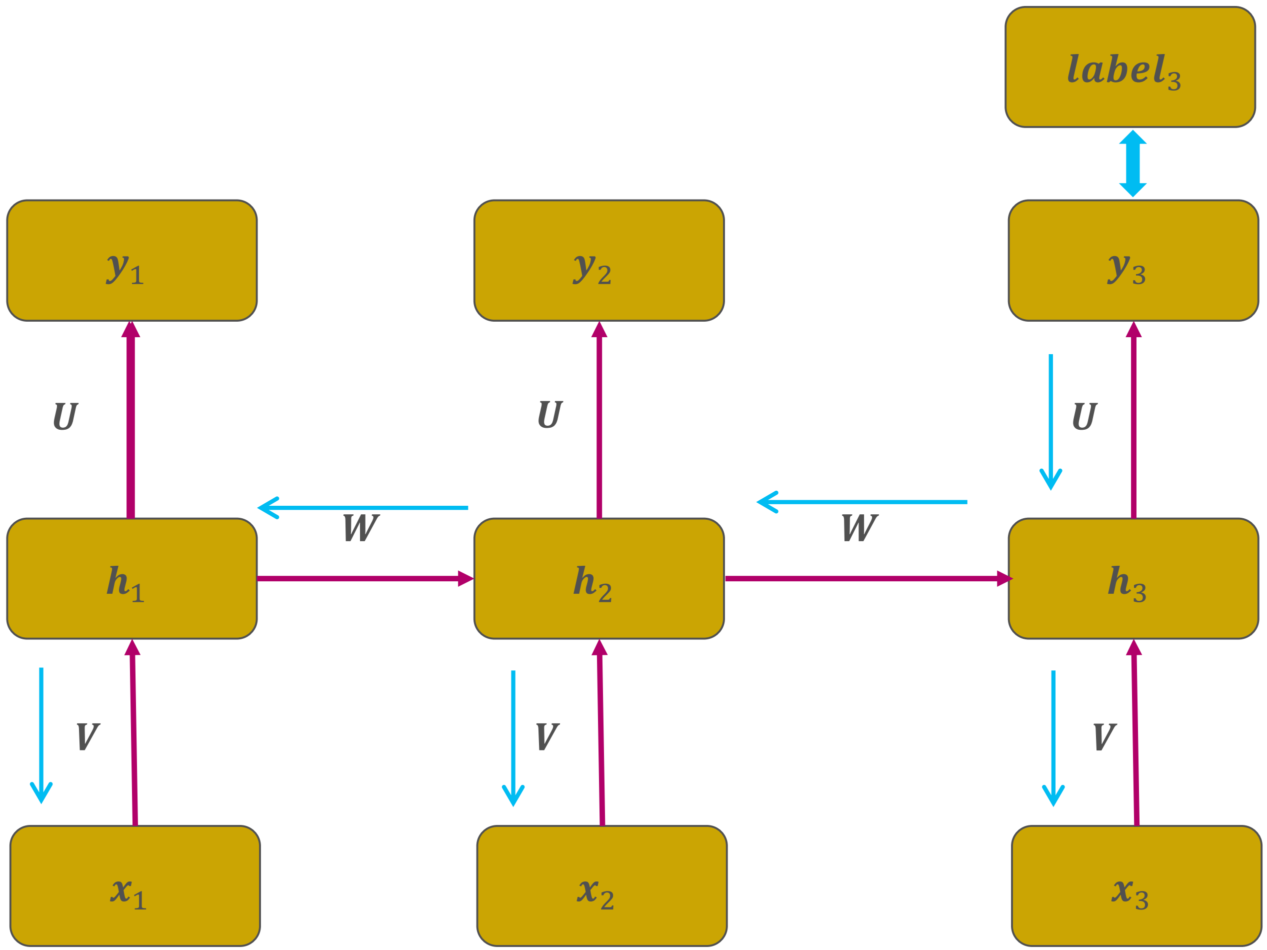
$y_t = SoftMax(U \cdot h_t), where\ h_t = \sigma(W \cdot h_{t-1} + V \cdot x_t)$



Elman RNN

[Mesnil, He, Deng, Bengio, 2013; Yao, Zweig, Hwang, Shi, Yu, 2013]

Back-propagation through time (BPTT)



at time $t = 3$

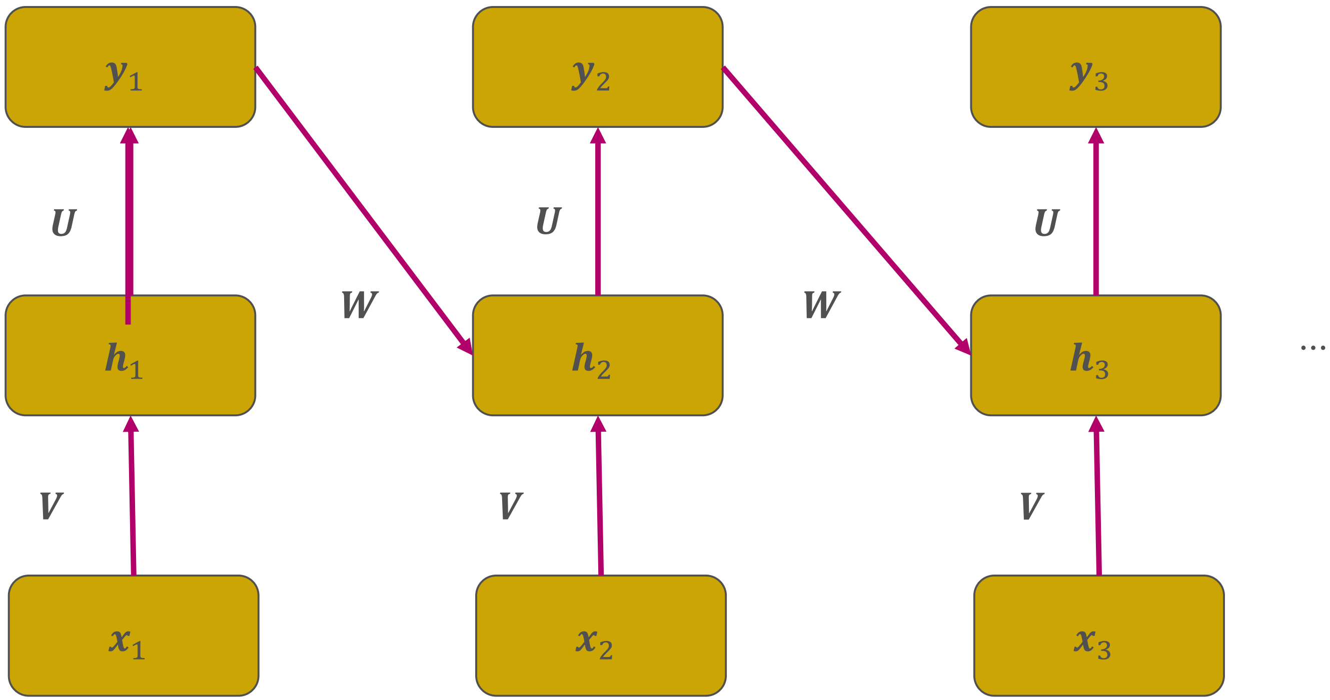
1. Forward propagation
2. Generate output
3. Calculate error
4. Back propagation
5. Back prop. through time

Jordan RNN

h_t is the hidden layer that carries the information from time $0 \sim t$

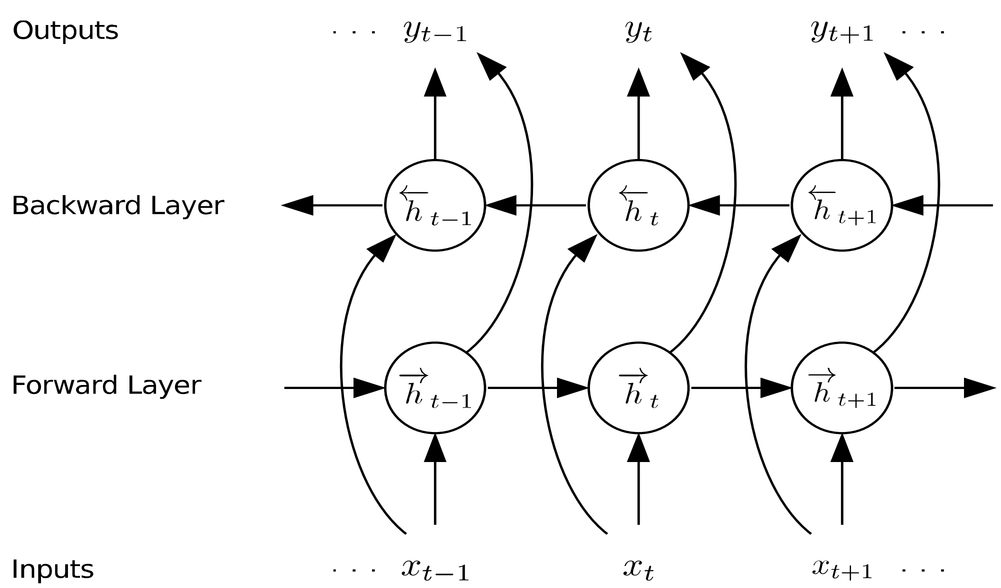
where x_t: *the input word* , y_t: *the output tag*

$$y_t = SoftMax(U \cdot h_t), where\ h_t = \sigma(W \cdot y_{t-1} + V \cdot x_t)$$



Elman-Jordan hybrid RNN is implemented, too.

Bi-directional RNN



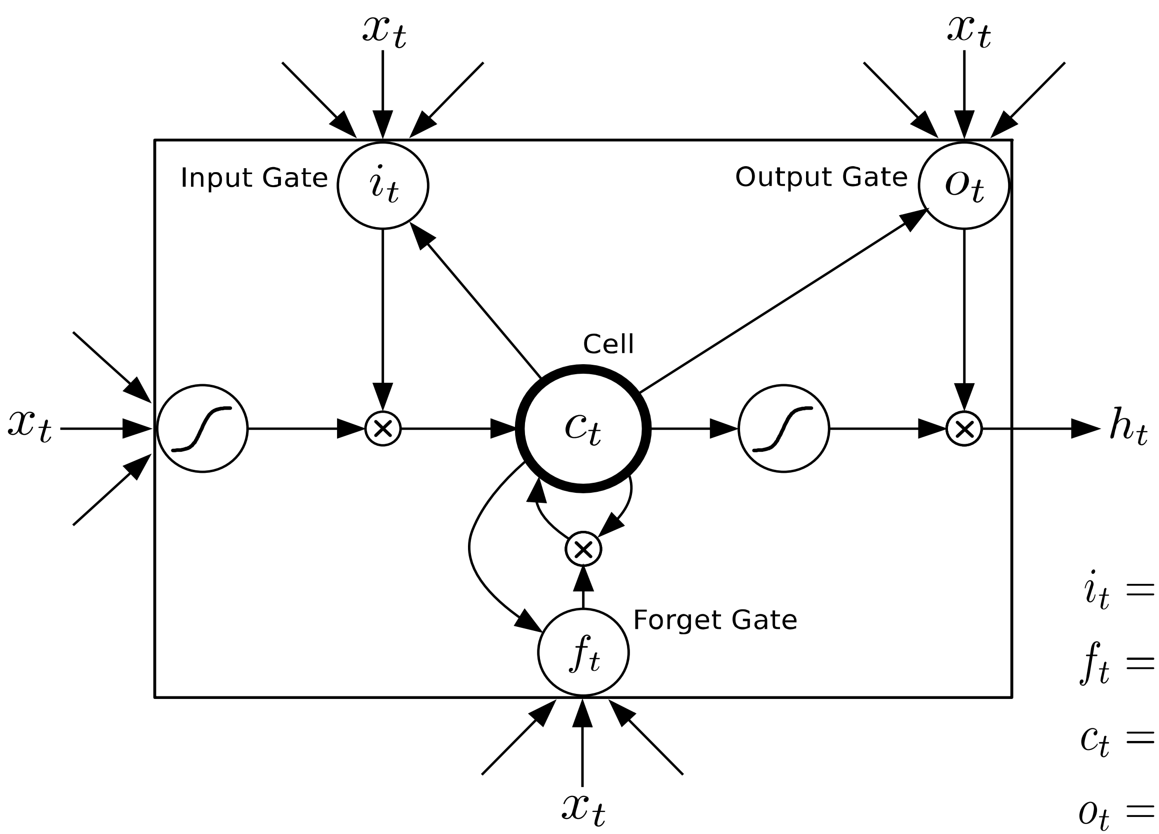
$$\overrightarrow{h}_t = \mathcal{H}\left(W_{x\overrightarrow{h}} x_t + W_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right)$$

$$\overleftarrow{h}_t = \mathcal{H}\left(W_{x\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}\right)$$

$$y_t = W_{\overrightarrow{h}y} \overrightarrow{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y$$

Information flow in the bi-directional RNN, with both diagrammatic and mathematical descriptions.

A Long-Short-Term-Memory Unit in LSTM-RNN



$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right)$$
$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right)$$
$$c_t = f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right)$$
$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right)$$
$$h_t = o_t \tanh(c_t)$$

Information flow in an LSTM unit of the RNN, with both diagrammatic and mathematical descriptions.

Results on the ATIS Benchmark

F1-score %	Elman	Jordan	Hybrid
RNN	94.98	94.29	95.06
FFN		93.32	
CRF		92.94	

Table 1: ATIS test set F1-score of the different models after 200 runs of random sampling for hyper-parameters selection. All models are trained via stochastic gradient. Lexical feature only.

RNN outperforms CRF and simple Feed-Forward Neural network significantly

F1-score %		Elman	Jordan	Hybrid
STO	Min	93.23	92.91	94.19
	Max	95.04	94.31	95.06
	Avg	94.44 ±0.41	93.81 ±0.32	94.61 ±0.18
MB	Min	92.8	93.17	93.06
	Max	94.42	94.15	94.21
	Avg	93.58 ±0.30	93.72 ±0.24	93.66 ±0.30

Table 2. Measurement of the impact of using different ways of training the models and random seed on the performance.

Stochastic gradient training gives better results than mini-batch training

The variations of Stochastic gradient training is slightly larger

Mesnil, Dauphin, Yao, Bengio, Deng, Hakkani-Tur, He, Heck, Tur, Yu, Zweig, "Using recurrent neural networks for slot filling in spoken language understanding," IEEE TASLP

Results: importance of using local context

F1-score	Elman	Jordan	Hybrid	CRF
Single, w/o context	93.15	65.23	93.32	69.68
BiDir, w/o context	93.46	90.31	93.16	
Single, context	94.98 (9)	94.29 (9)	95.06 (7)	92.94 (9)
Bidir, context	94.73 (5)	94.03 (9)	94.15 (7)	

Table 3. F1-score of single and Bi-Directional models with or w/o context windows. We report the best context window size hyper-parameter as the number in the round brackets.

Without using local n-gram feature, CRF's performance degrades significantly

RNN models degrade much less except Jordan RNN (recall Jordan vs. Elman RNN)

Bi-direction modeling helps a lot for Jordan RNN (because of bring in context)

But with rich local context, bi-direction modeling doesn't help

Results: using extra features / noisy input

F1-score	Elman	Jordan	Hybrid	CRF
Word	94.98	94.29	95.06	92.94
Word+NE	96.24	95.25	95.85	95.16

Table 4. Performance with Named Entity features.

RNN, just like CRF, can take benefit of using extra features (like Named Entity ID)

F1-score	Elman	Jordan	Hybrid	CRF
Word	94.98	94.29	95.06	92.94
ASR	85.05	85.02	84.76	81.15

Table 5. Comparison between manually labeled word and ASR output.

RNN is robust under noisy input condition

More Results

Adding a Viterbi decoding process on top of RNN's output helps, especially for difficult task

Drop-out provides effective regularization for RNN training

F1-score	Elman	Jordan	Hybrid
ATIS Word	94.98	94.29	95.06
ATIS Word +Viterbi	94.99 (+0.01)	94.25 (-0.04)	94.77 (-0.29)
ATIS Word/CRF		92.94	
ATIS ASR	85.05	85.02	84.76
ATIS ASR +Viterbi	86.16 (+1.11)	85.21 (+0.19)	85.36 (+0.6)
ATIS ASR/CRF		81.15	
Entertainment	88.67	88.70	89.04
Entertainment +Viterbi	90.19 (+1.42)	90.62 (+1.92)	90.01 (+0.97)
Entertainment +Viterbi +Dropout	-	91.14 (+2.44)	-
Entertainment /CRF		90.64	

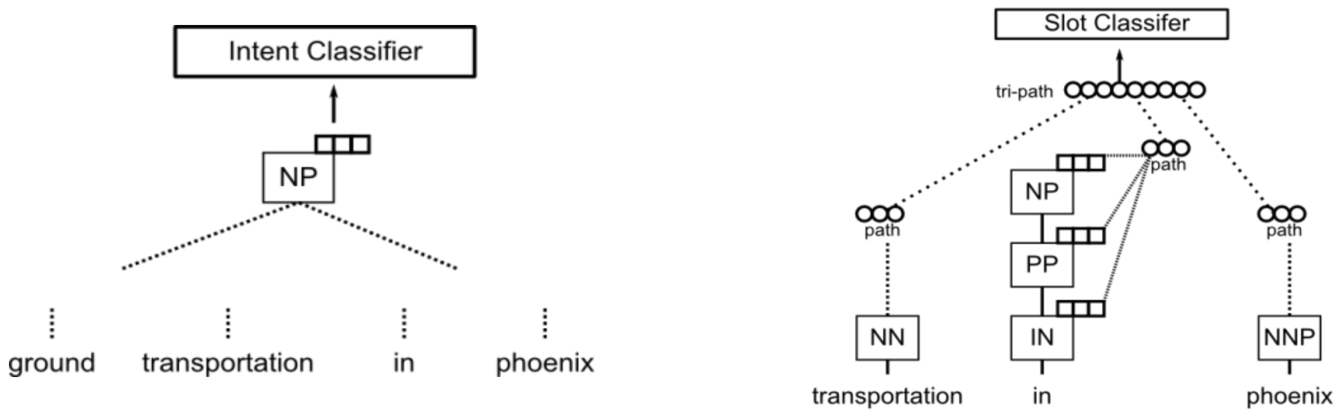
Table 6. Comparison with Viterbi decoding with different methods on several datasets

Other relevant work

Deep Believe Network [Deoras and Sarikaya, 2013]
Recurrent CRF [Yao, Peng, Zweig, Yu, Li, Gao, 2014]

Recursive NN [Guo, Tur, Yih, Zweig, IEEE-SLT 2014]

Use one recursive NN model to jointly predict the semantic label and slots of utterances from a spoken dialog system



Interim Summary

- Introduction to SLU
- DNN/DCN/K-DCN for Domain/intent detection
- RNN and its variants for slot filling
- Deep learning models demonstrate superior performances on these tasks

However, understanding human language is more challenging than that …

Part III

Learning Semantic Embedding

Why understanding language is difficult?

The meaning of text is usually vague and latent

e.g., no clear "supervision" signal to learn from as in speech/image recog.

and many NLP tasks are not classification tasks

Human language has great variability

similar concepts are expressed in different ways, e.g., *kitty* vs. *cat*

Human language has great ambiguity

similar expressions mean different concepts, e.g.,

new york vs. *new york times*

Learning semantic meaning of texts is a key challenge in language processing

Semantic embedding

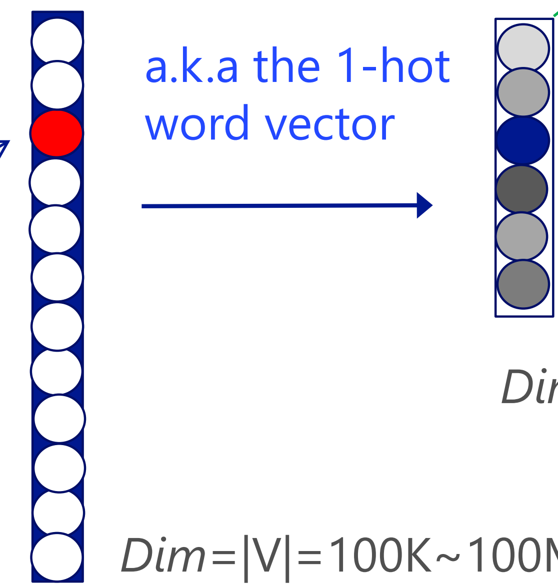
Project raw text into a continuous semantic space

e.g., word embedding

Captures the word meaning in a semantic space

$f(\boldsymbol{cat}) =$

The index of "cat" in the vocabulary

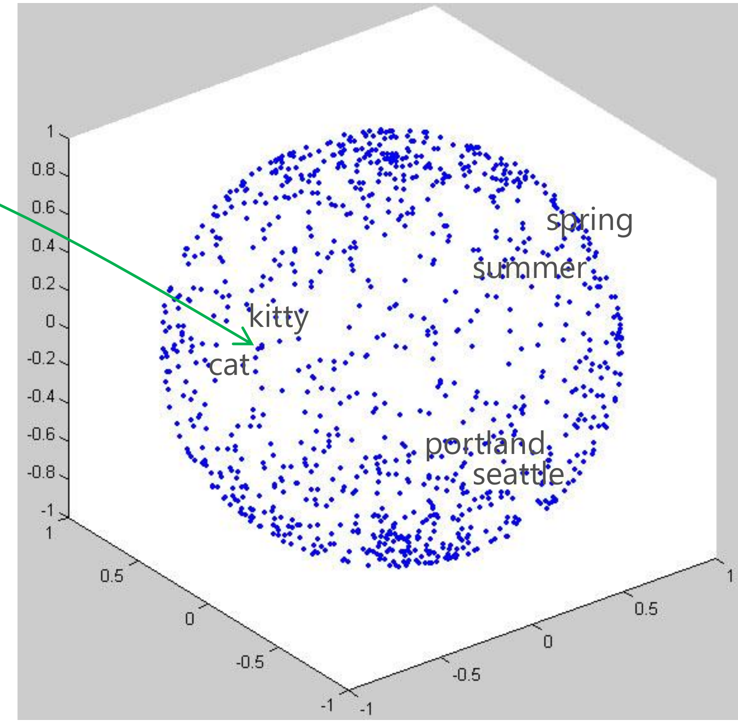


a.k.a the 1-hot word vector

word embedding vector in the semantic space

Dim=100~1000

Dim=|V|=100K~100M



Deerwester, Dumais, Furnas, Landauer, Harshman, "Indexing by latent semantic analysis," JASIS 1990

SENNA word embedding

Scoring:

$$Score(w_1, w_2, w_3, w_4, w_5) = U^T \sigma(W[f_1, f_2, f_3, f_4, f_5] + b)$$

Training:

$$J = \max(0, 1 + S^- - S^+)$$ e.g., update the model until $S^+ > 1 + S^-$

Where

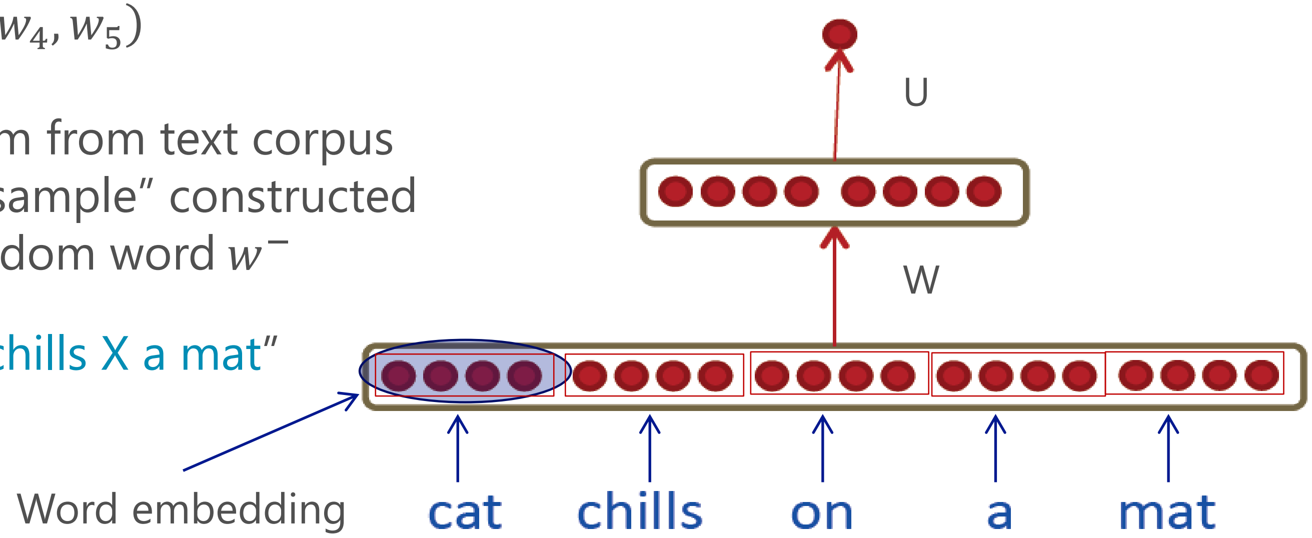
$$S^+ = Score(w_1, w_2, w_3, w_4, w_5)$$
$$S^- = Score(w_1, w_2, w^-, w_4, w_5)$$

And

$< w_1, w_2, w_3, w_4, w_5 >$ is a valid 5-gram from text corpus

$< w_1, w_2, w^-, w_4, w_5 >$ is a "negative sample" constructed by replacing the word w_3 with a random word w^-

e.g., a negative example: "cat chills X a mat"

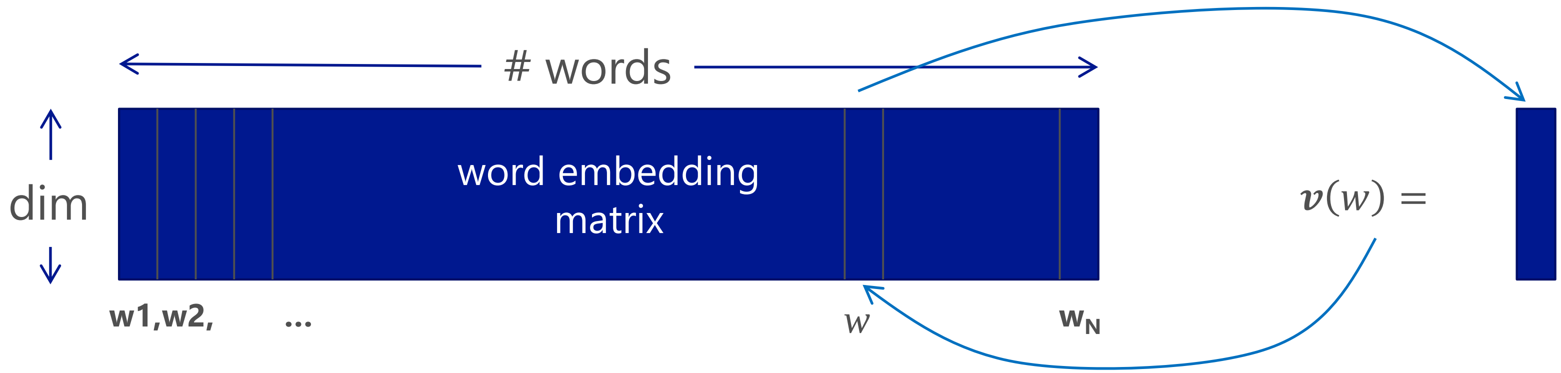


Collobert, Weston, Bottou, Karlen, Kavukcuoglu, Kuksa, "Natural Language Processing (Almost) from Scratch," JMLR 2011

[and Mikolov, Yih, Zweig, NAACL 2013; Mikolov et al., ICLR 2013; etc.]

Word embedding: rethinking

- Word embedding is a neat and effective representation:

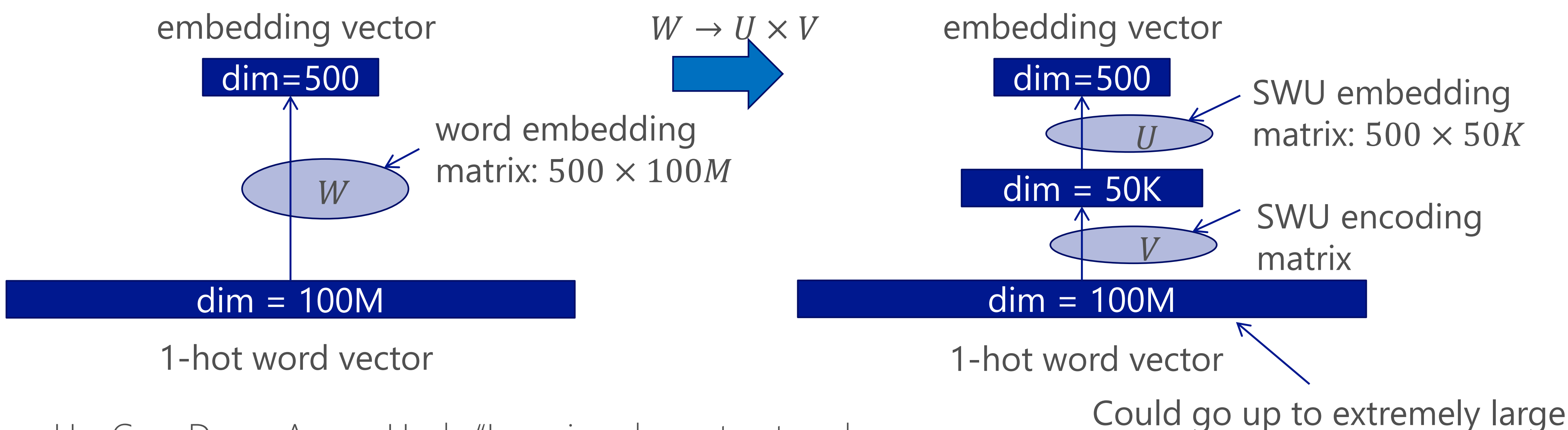


- However, for large scale NL tasks a decomposable, robust representation is preferable
  - Vocabulary of real-world big data tasks could be huge (*scalability*)
    - >100M unique words in a modern commercial search engine log, and keeps growing
  - New words, misspellings, and word fragments frequently occur (*generalizability*)

Build semantic embedding on top of sub-word units

Learn semantic embedding on top of sub-word units (SWU)

- Decompose *any* word into sub-word units
- *Scale* the capacity to handle almost unbounded variability (word) based on bounded variability (sub-word)



Huang, He, Gao, Deng, Acero, Heck, "Learning deep structured semantic models for web search using clickthrough data," CIKM, 2013

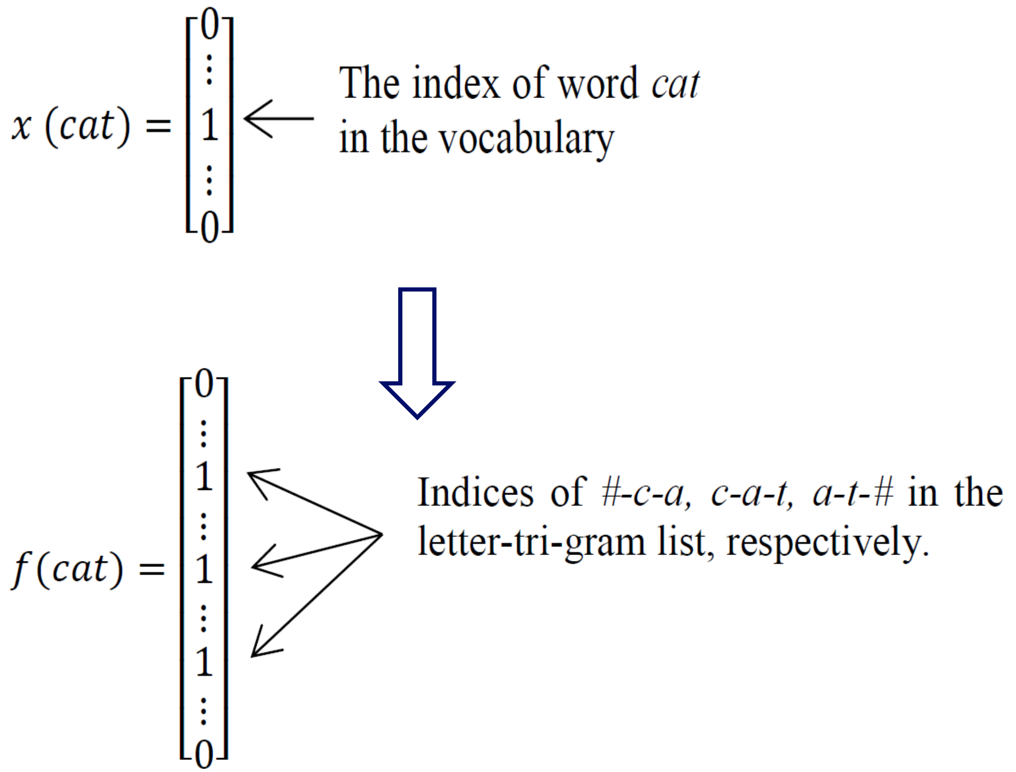
Sub-word unit

- Letters, context-dept letters, positioned-phones, context-dept phones, positioned-roots/morphs, context-dept morphs
- Multi-hashing approach to word input representation

Or random projection (random basis)

Sub-word unit encoding

- E.g., letter-trigram based Word Hashing of "cat"
  - -> #cat#
  - Tri-letters: #-c-a, c-a-t, a-t-#.
- Compact representation
  - |Voc| (500K) → |Letter-trigram| (30K)
- Generalize to unseen words
- Robust to misspelling, inflection, etc.



What if different words have the same word hashing vector (collision)?

Vocabulary size	Unique letter-tg observed in voc	Number of Collisions
40K	10306	2 (0.005%)
500K	30621	22 (0.004%)

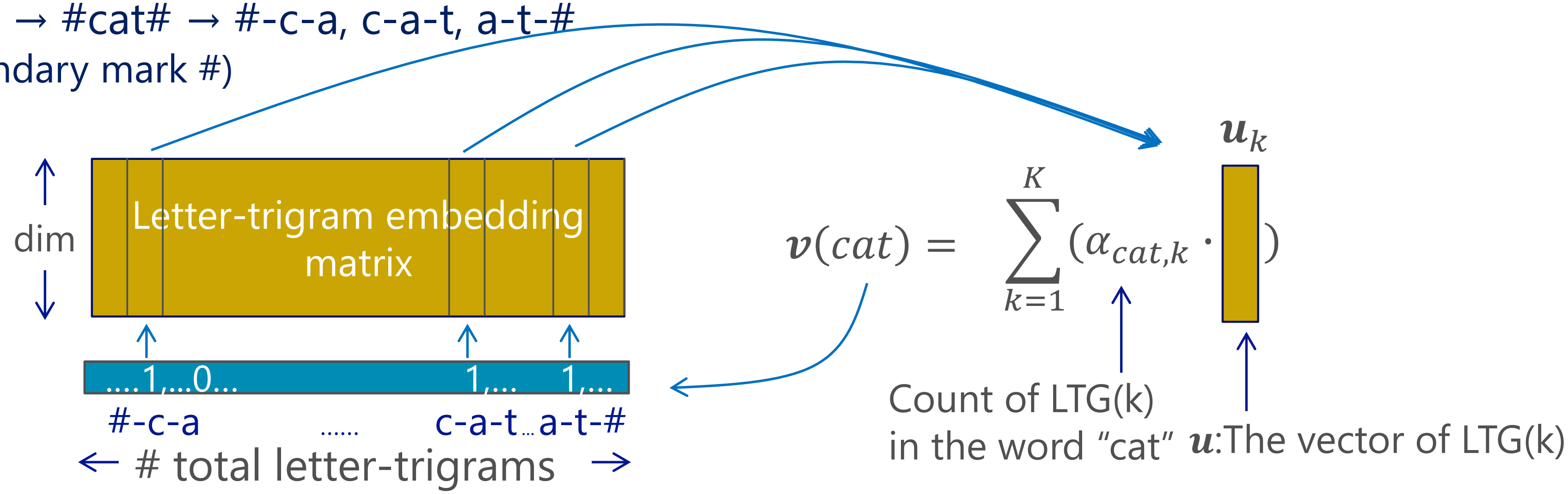
From sub-word unit embedding vectors to word vectors

SWU uses context-dependent letter, e.g., letter-trigram.

Learn **one vector per letter-trigram** (LTG), the encoding matrix is a fixed matrix

- Use the count of each LTG in the word for encoding

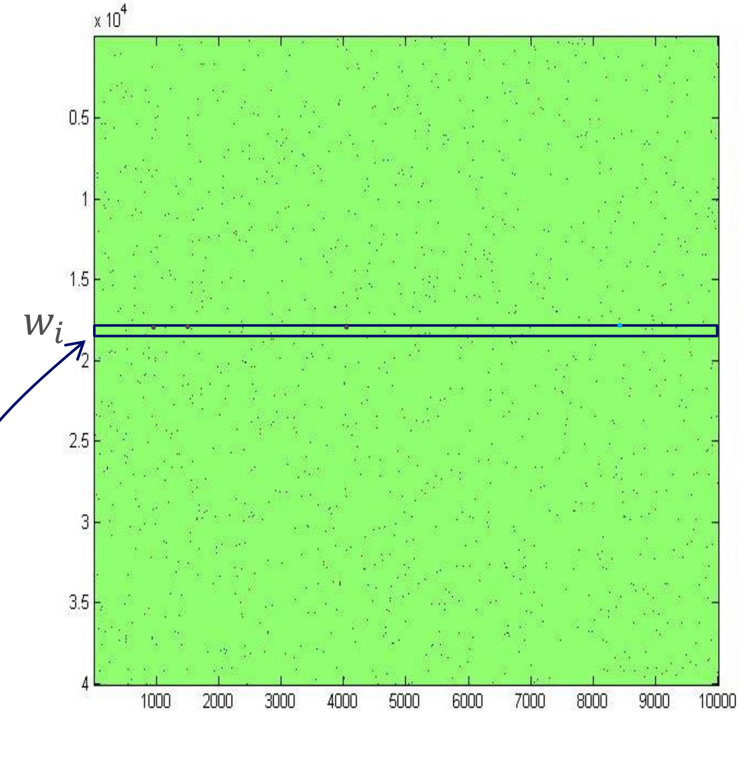
Example: cat → #cat# → #-c-a, c-a-t, a-t-#
(w/ word boundary mark #)



Two words has the same LTG: collision rate ≈ 0.004%

Other representation: random projection

- Sparse random projection matrix R with entries sampled i.i.d. from a distribution over [0, 1, -1]
- Entries of 1 and -1 are equally probable
- $P\left(R_{ij} = 0\right) = 1 - \frac{1}{\sqrt{d}}$, where d is the original input dimensionality.



Each word will have a set of sparse random encoding of the 10000 basic units

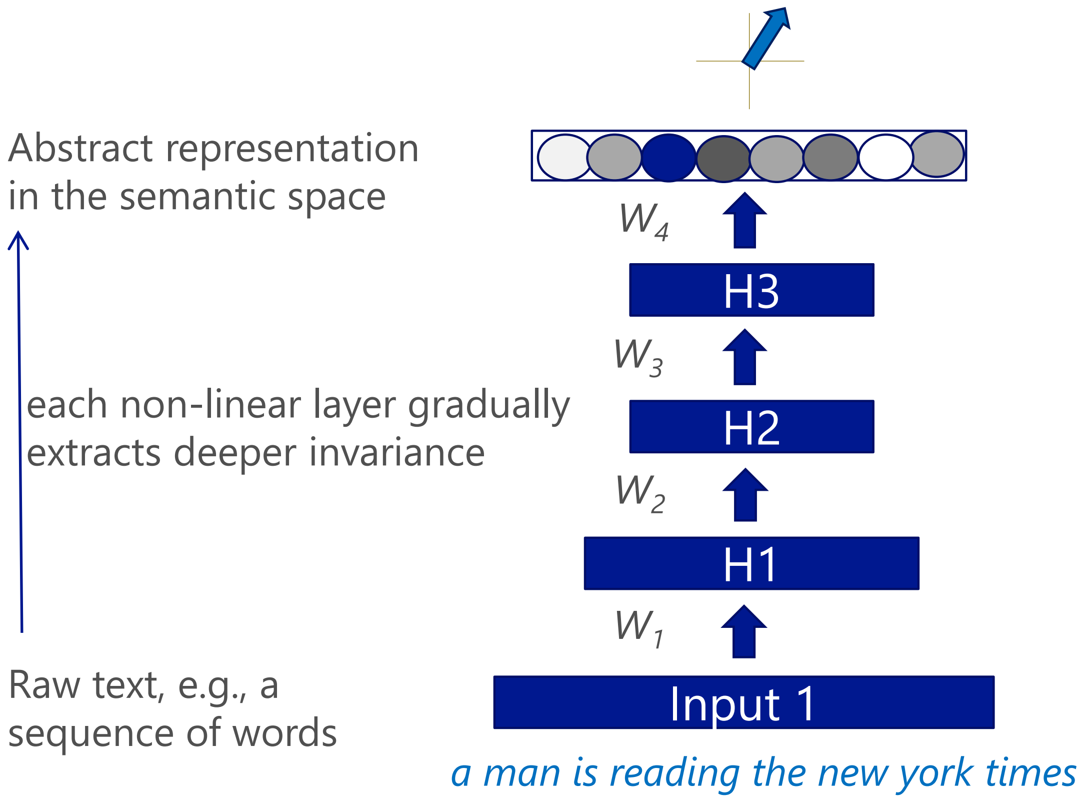
[Li, Hastie, and Church 2006]

Semantic embedding: from words to sentences

The semantic intent is better defined at the phrase/sentence level rather than at the word level

- The meaning of a single word is often ambiguous
- A phrase/sentence/document contains rich contextual information that could be leveraged

Deep learning for semantic embedding



However

- the semantic meaning of texts – to be learned – is latent
- no clear target for the model to learn
- How to do back-propagation / training?

Fortunately

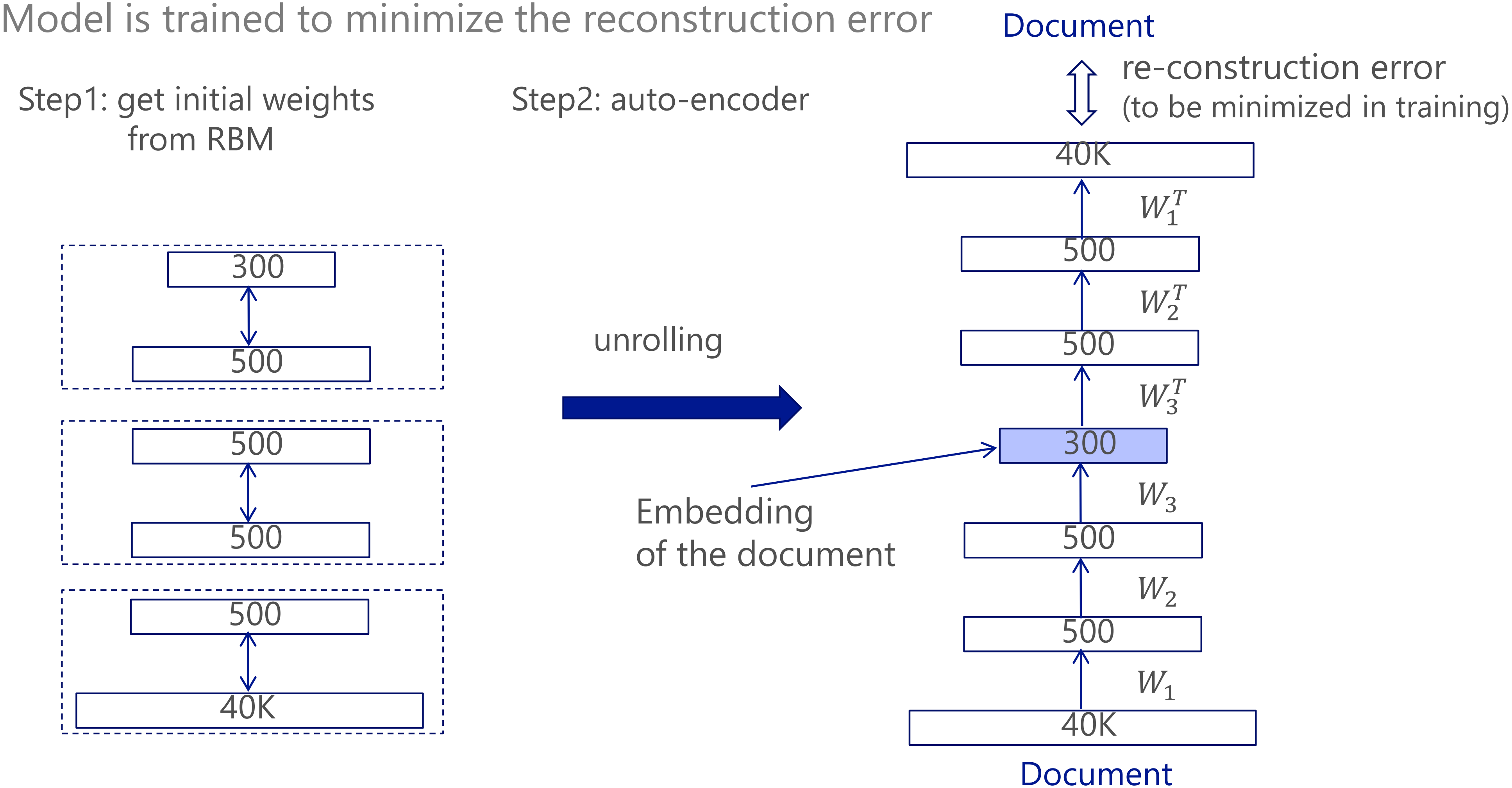
- we usually know if two texts are "similar" or not.
- That's the signal for semantic representation learning.

Semantic Hashing

[Salakhutdinov & Hinton 2007, 2010]

1) Single layer learning: Restricted Boltzmann Machine (RBM)
2) Multi-layer training: deep auto-encoder, learn internal representations

Model is trained to minimize the reconstruction error



Auto-encoder: rethinking

- The objective of the auto-encoder?
  - What is the relation between minimizing re-construction error and learning a good embedding?
- What is a *good* embedding?
  - General embedding or useful embedding for tasks?
    - Optimizing embedding directly instead of minimizing the doc re-construction error
    - Learning the model with end-to-end user behavior log data (weak supervision) beside documents

Deep Structured Semantic Model

- Deep Structured Semantic Model/Deep Semantic Similarity Model (**DSSM**)
  - the DSSM learns phrase/sentence level semantic vector representation, e.g., query, document
- The DSSM is built upon sub-word units for scalability and generalizability
  - e.g., letter-trigram, phones, roots/morphs
- The DSSM is trained by an similarity-driven objective
  - projecting semantically similar phrases to vectors close to each other
  - projecting semantically different phrases to vectors far apart
- The DSSM is trained using various signals, with or without human labeling effort
  - semantically-similar text pairs
  - e.g., user behavior log data, contextual text

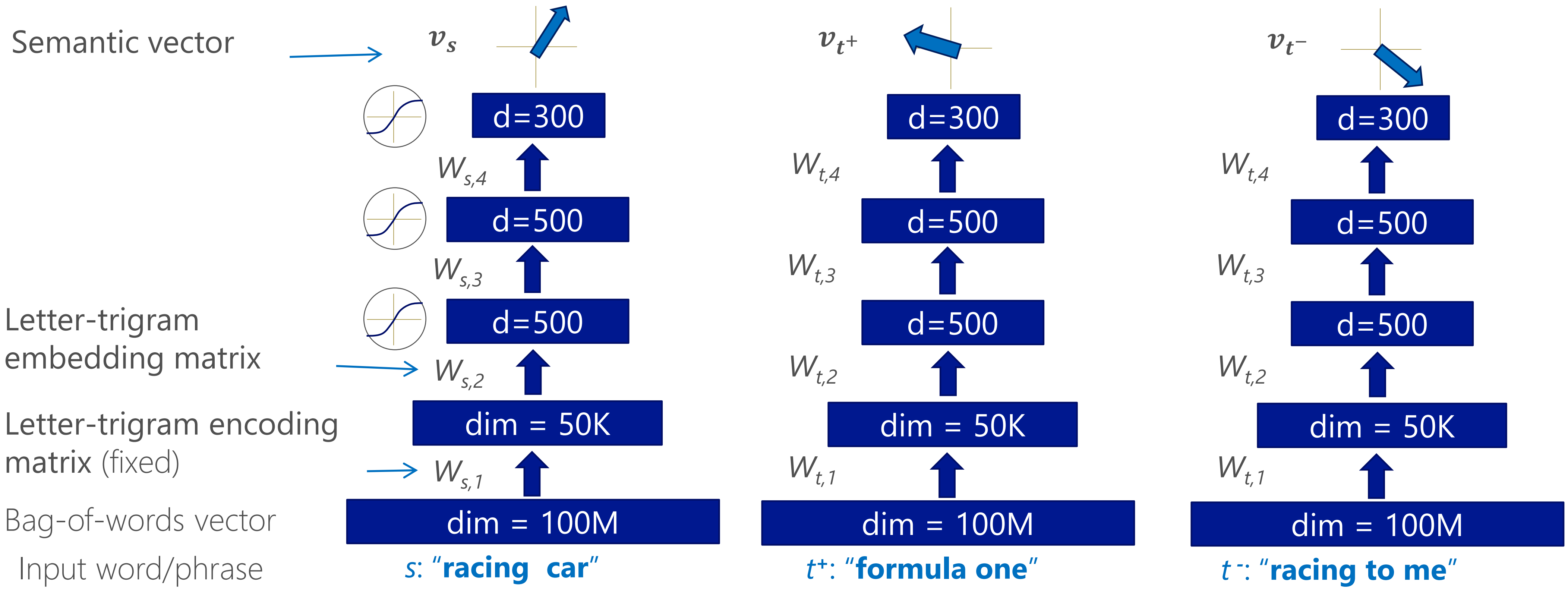
[Huang, He, Gao, Deng, Acero, Heck, CIKM2013]
[Shen, He, Gao, Deng, Mesnil, WWW2014]
[Gao, He, Yih, Deng, ACL2014]
[Yih, He, Meek, ACL2014]
[Song, He, Gao, Deng, Shen, MSR-TR 2014]
[Gao, Pantel, Gamon, He, Deng, Shen, EMNLP2014]
[Shen, He, Gao, Deng, Mesnil, CIKM2014]
[He, Gao, Deng, ICASSP2014]

DSSM for semantic embedding Learning

Initialization:

Neural networks are initialized with random weights

Huang, He, Gao, Deng, Acero, Heck, "Learning deep structured semantic models for web search using clickthrough data," CIKM, 2013

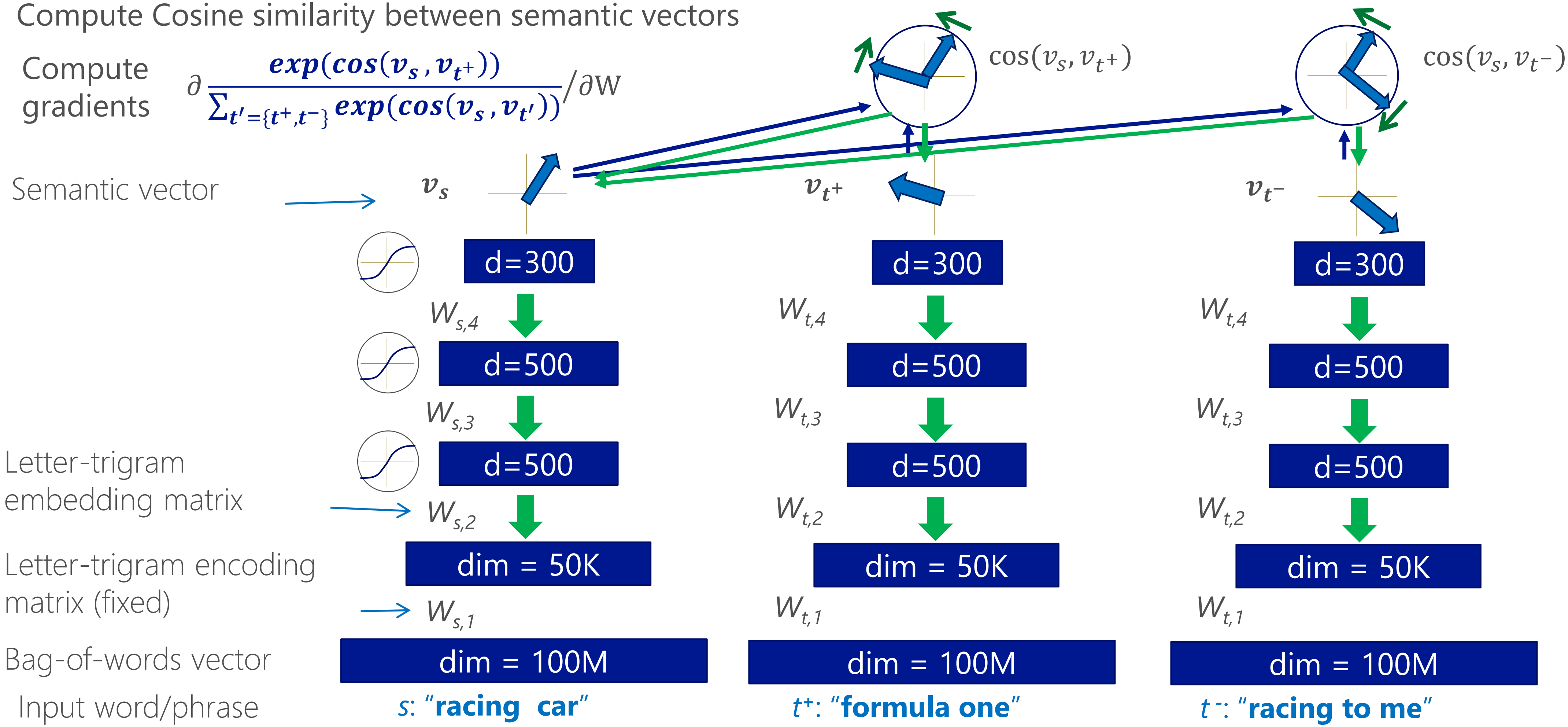


DSSM for semantic embedding learning

Training:

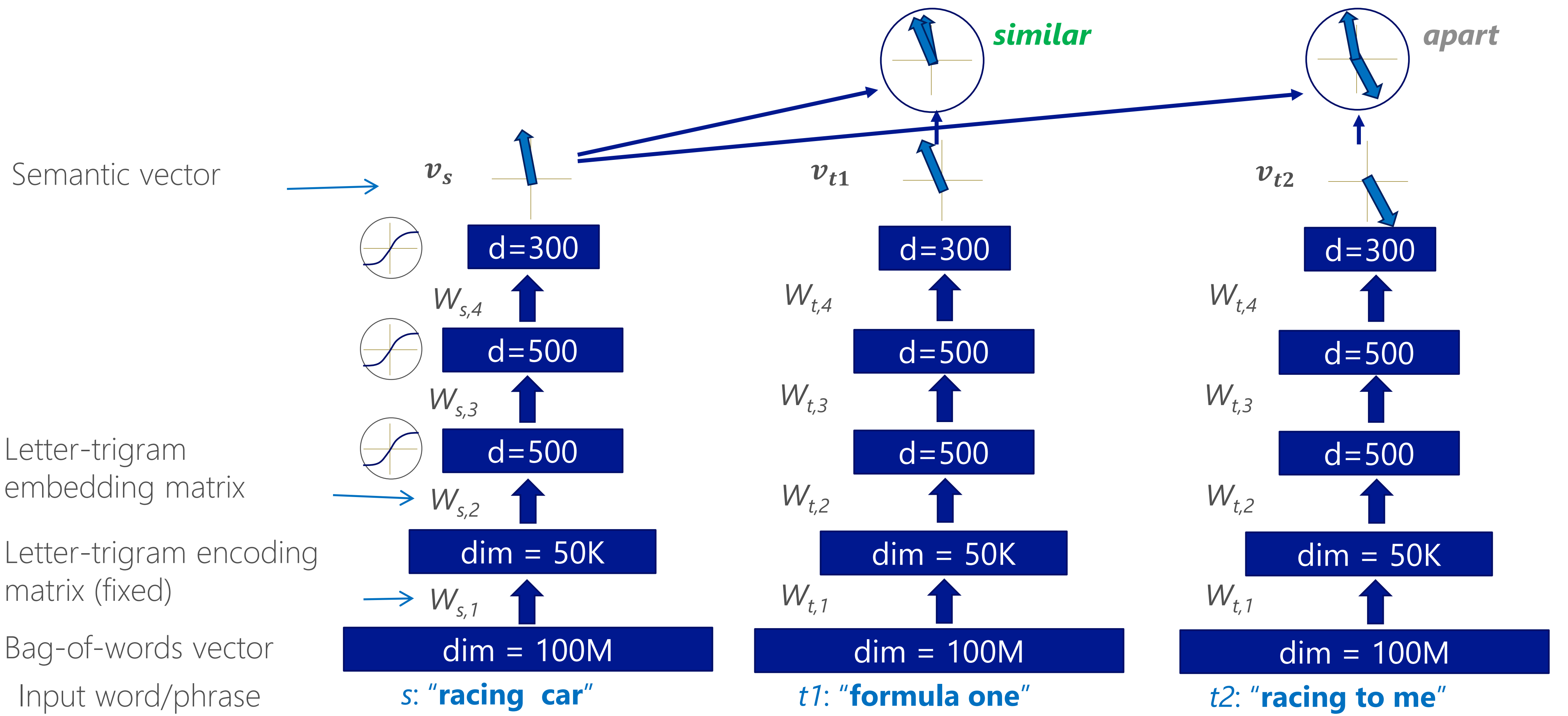
Compute Cosine similarity between semantic vectors

Compute gradients $\partial \dfrac{exp(cos(v_s, v_{t^+}))}{\sum_{t'=\{t^+,t^-\}} exp(cos(v_s, v_{t'}))} / \partial W$



DSSM for semantic embedding learning

Runtime:



Training of the DSSM

Data: semantically-similar text pairs

e.g., context <-> word in word embedding vector learning
query <-> clicked-doc in Web Search
pattern<-> predicate in Question Answering

Objective: cosine similarity based loss

- Web search as an example: a query $\boldsymbol{q}$ and a list of docs $\boldsymbol{D} = \{\boldsymbol{d}^+, \boldsymbol{d}_1^-, \dots \boldsymbol{d}_K^-\}$
  - $\boldsymbol{d}^+$ positive doc; $\boldsymbol{d}_1^-, \dots \boldsymbol{d}_K^-$ are negative docs to $\boldsymbol{q}$ (e.g., sampled from not clicked docs)

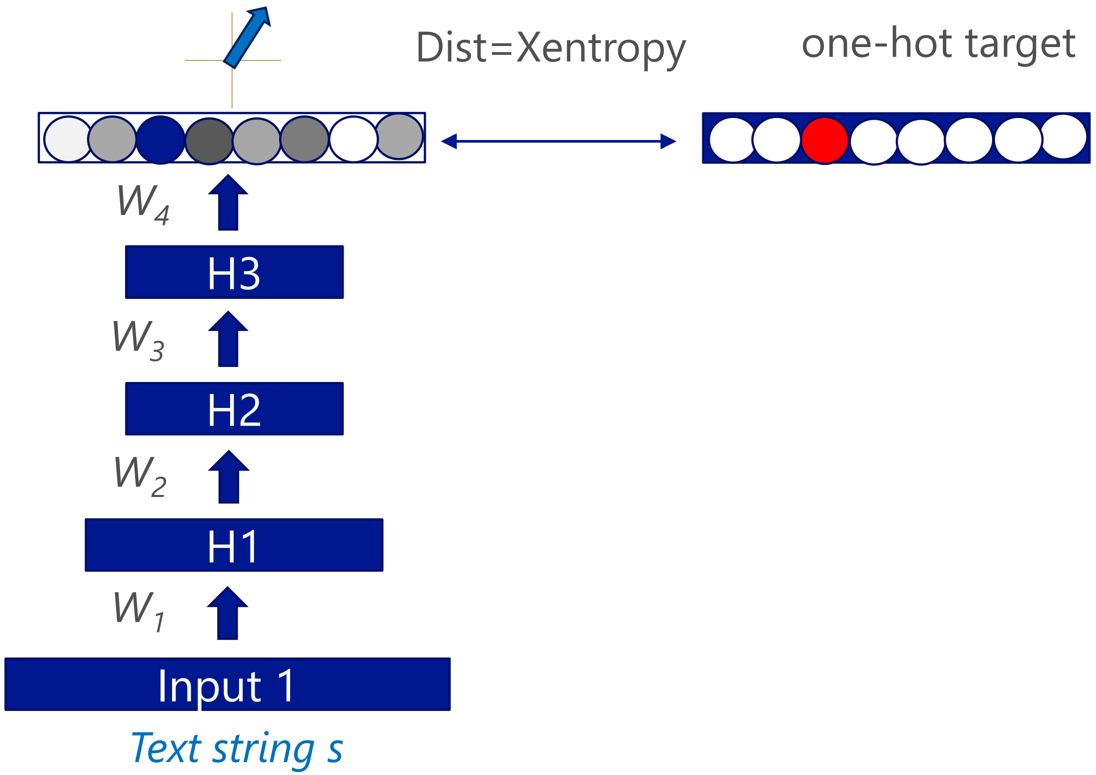
- Objective: the posterior probability of clicked document given query

$$P(d^+|q) = \frac{\exp\left(\gamma\, cos(q, d^+)\right)}{\sum_{d \in \boldsymbol{D}} \exp(\gamma\, cos(q, d))}$$

- Optimize $\boldsymbol{\theta}$ to maximize $\boldsymbol{P(d^+|q)}$. SGD training on GPU (NVidia K20x)

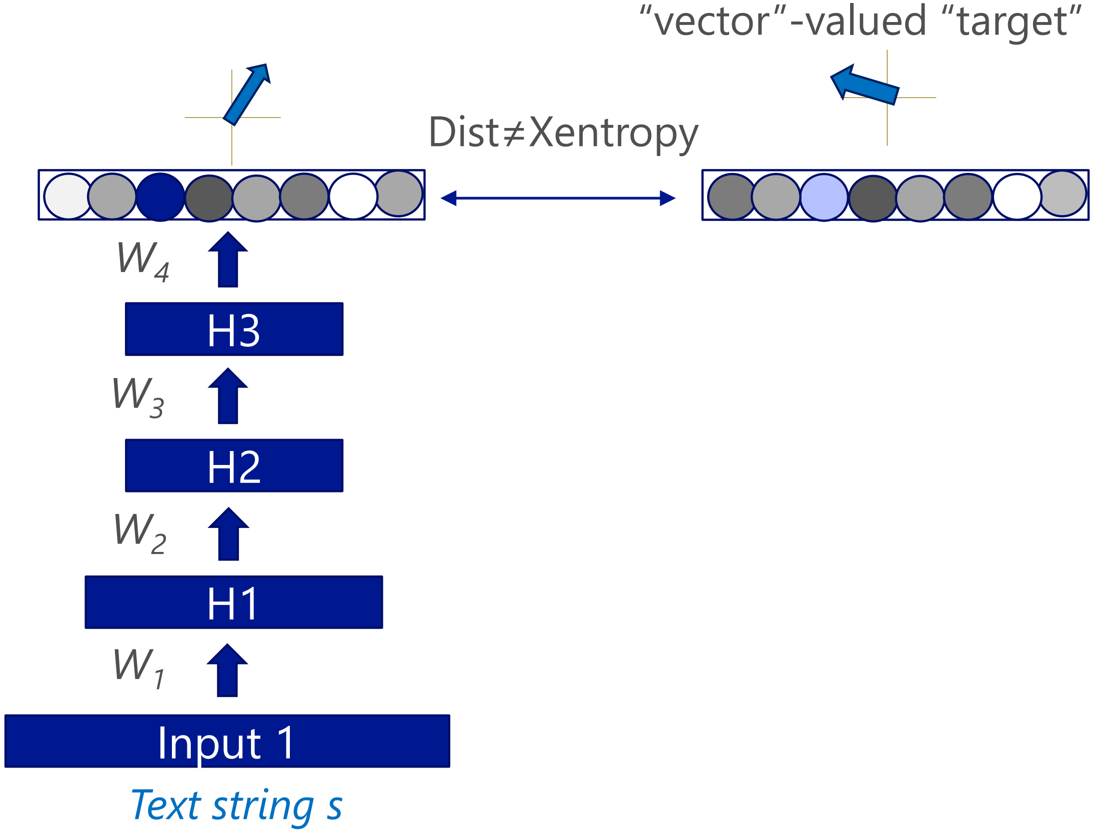
Reflection: from DNN to DSSM

- Common deep neural network models:
  - Mainly for **classification** (speech reco, image reco, SLU, LM)
  - Target: one-hot vector
  - Example of DNN:



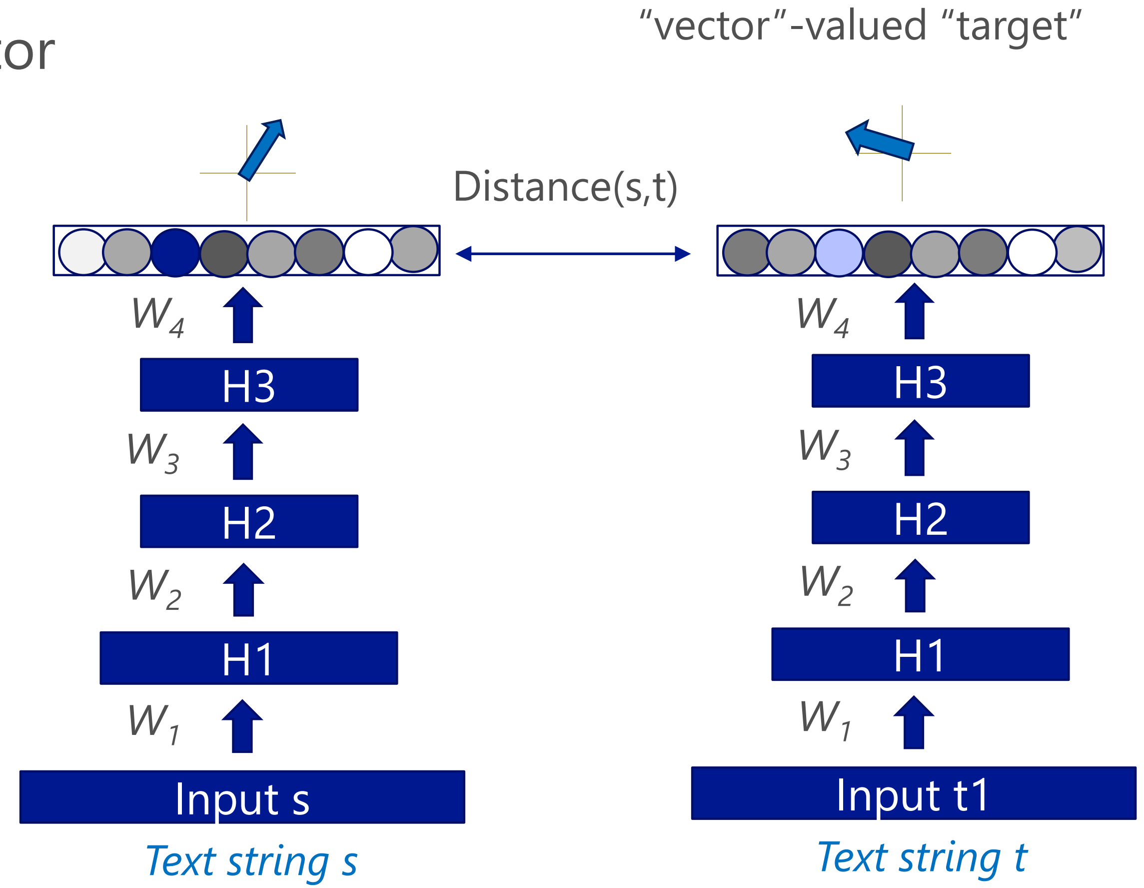
Reflection: from DNN to DSSM

- DSSM
  - For **semantic matching / ranking** (not classification with DNN)
  - Step 1: target from "one-hot" to continuous-valued vectors



Reflection: from DNN to DSSM

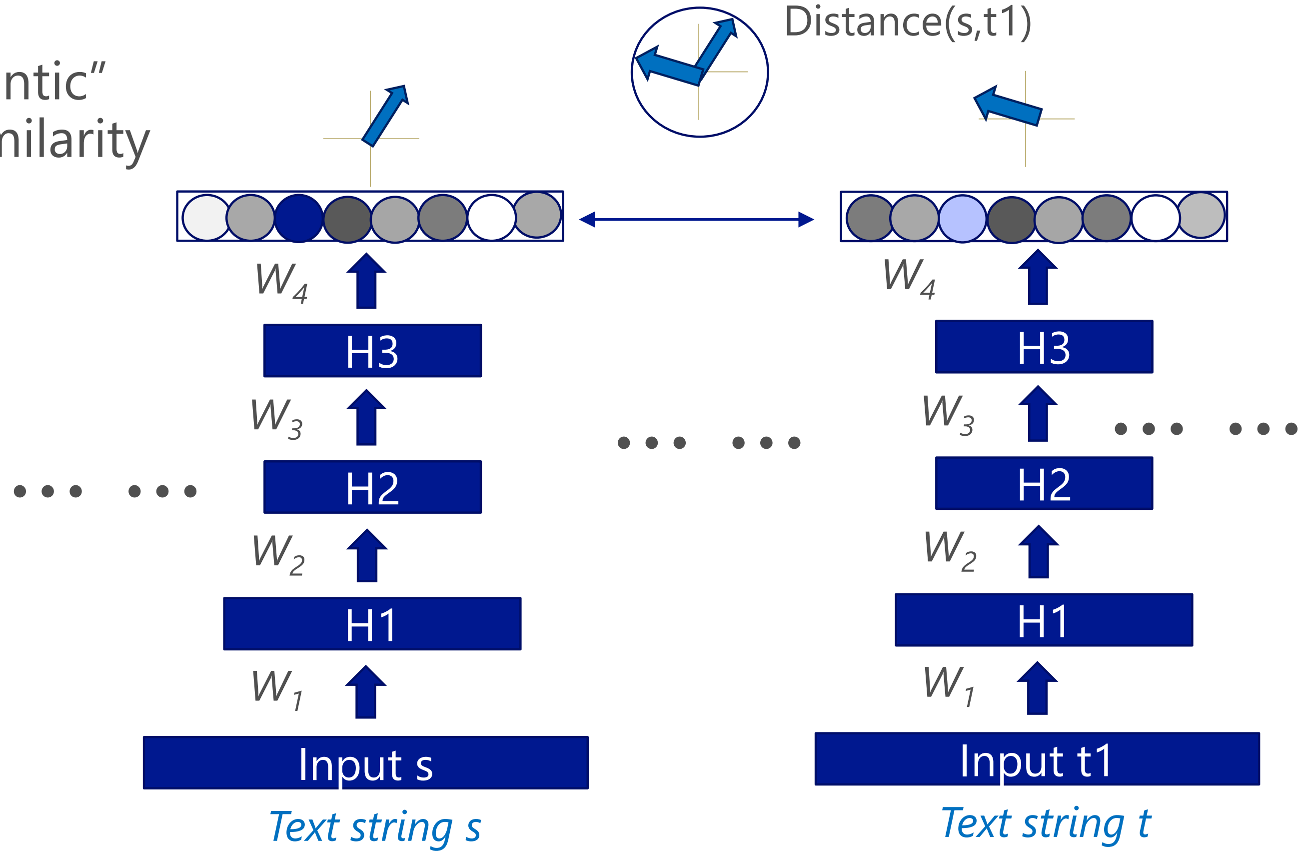
- To construct a DSSM
  - Step 1: target from "one-hot"
    to continuous-valued vectors
  - Step 2: derive the "target" vector
    using a deep net



Reflection: from DNN to DSSM

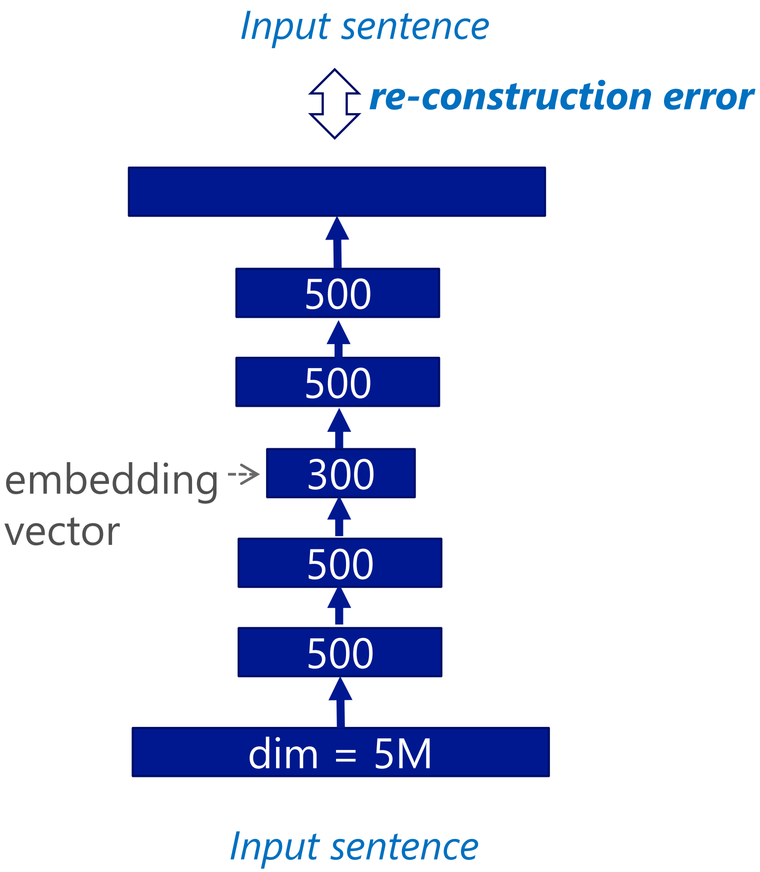
- To construct a DSSM
  - Step 1: target from "one-hot" to a continuous-valued vector
  - Step 2: derive the "target" vector using a deep net
  - Step 3: normalize two "semantic" vectors & computer their similarity

From classification to semantic matching



Reflection: from Auto-encoder to DSSM

Auto-encoder

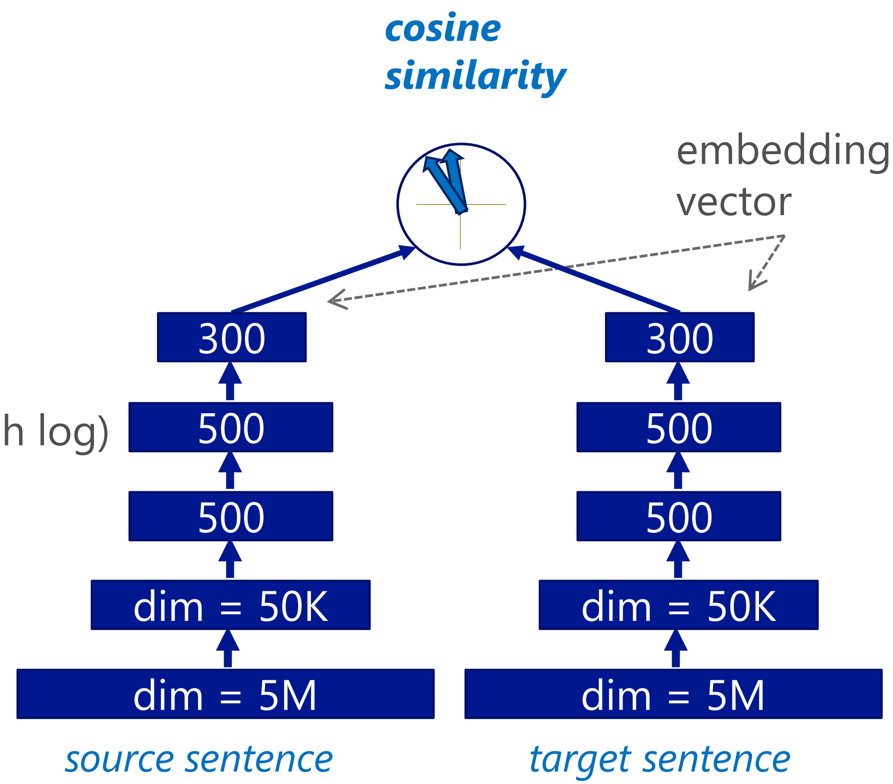


Training loss func.:
AE: reconstruction error of the input
DSSM: distance between embedding vectors

Training data:
AE: unsupervised (e.g., doc<->doc)
DSSM: weakly supervised (e.g., query<->doc search log)

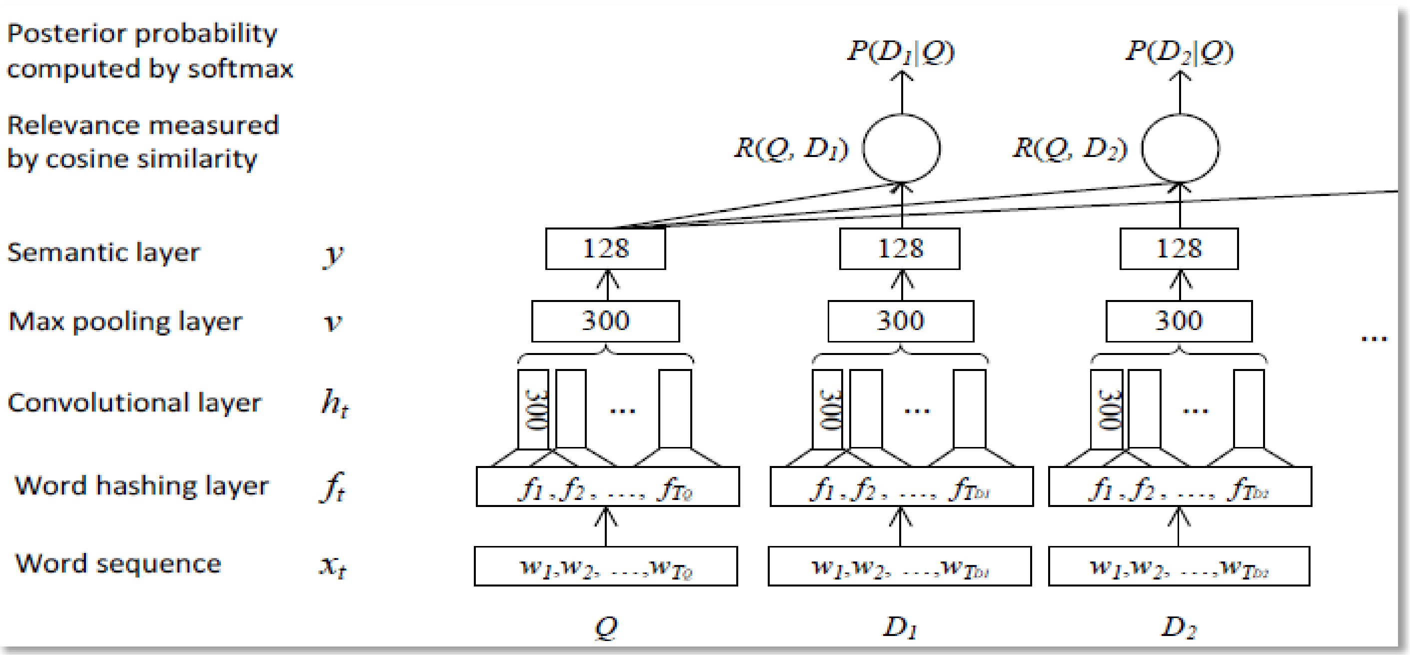
Input:
AE: 1-hot word vector
DSSM: sub-word unit (e.g., letter-trigram)

DSSM



The DSSM can be trained using a variety of weak supervision signals without human labeling effort (e.g., user behavior log data).

Further extension: Convolutional DSSM

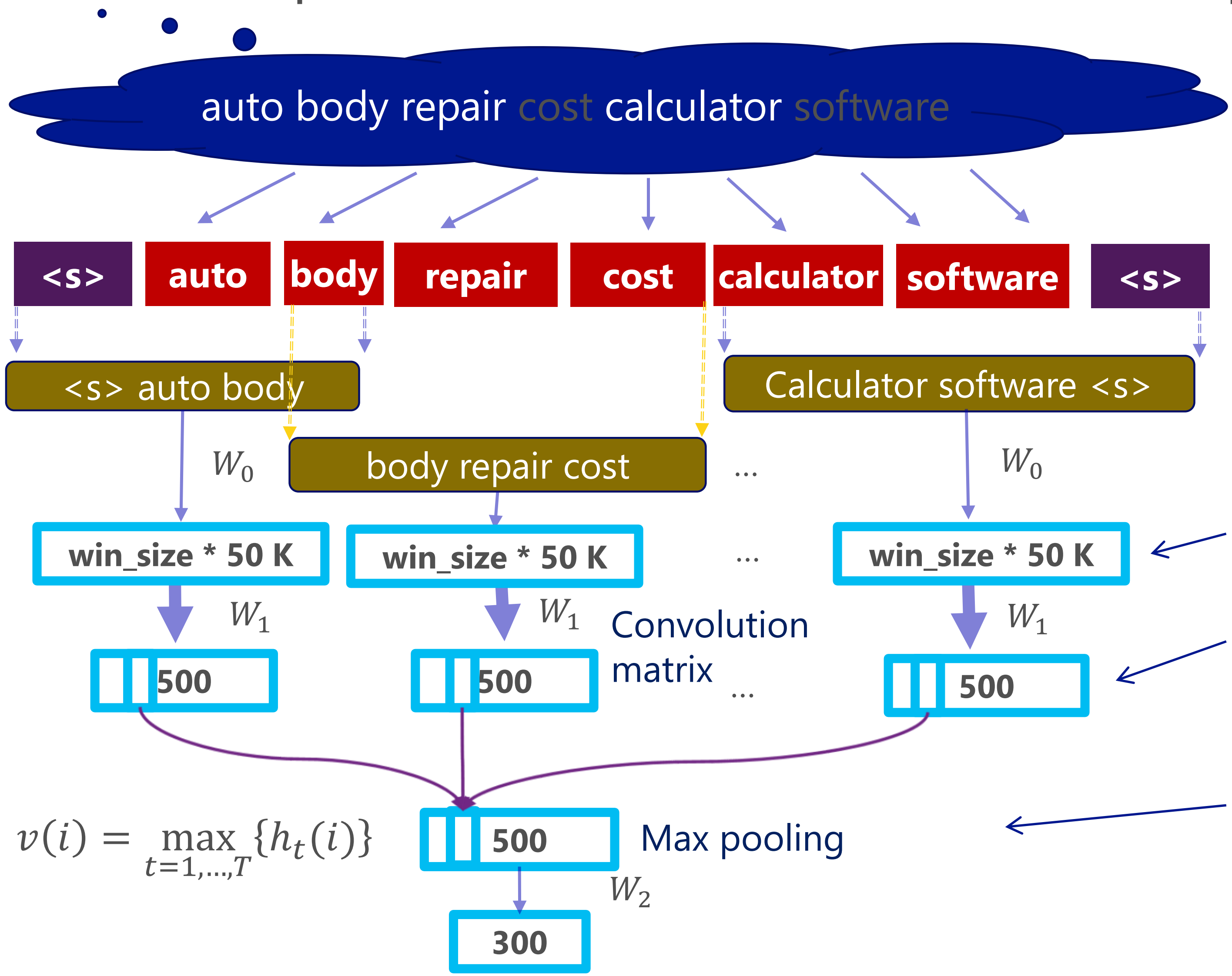


Word sequence input: capture the sequential structure in the text (in stead of using bag-of-words)

Convolutional and Max-pooling layer: identify key words/concepts in Q and D

Shen, He, Gao, Deng, Mesnil, "A latent semantic model with convolutional-pooling structure for IR," CIKM 2014

Example: semantic intent representation



Query as a word sequence rather than "bag of words"

Sliding Window input: n-gram phrase (n = 3)

Letter-trigram representation

Convolutional Layer *h*: generate word-within-context embedding

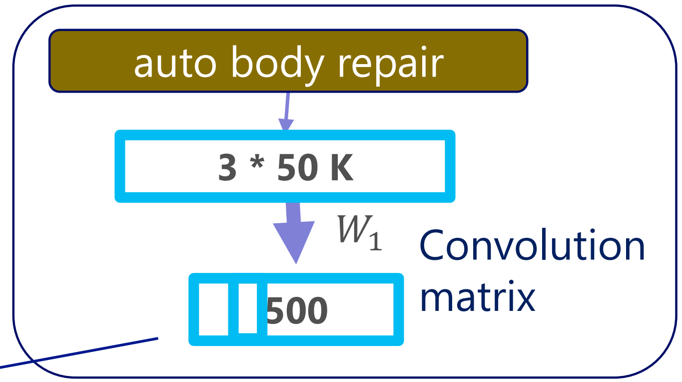
Max Pooling Layer *v*: identify key words in a query

Semantic Layer *y*

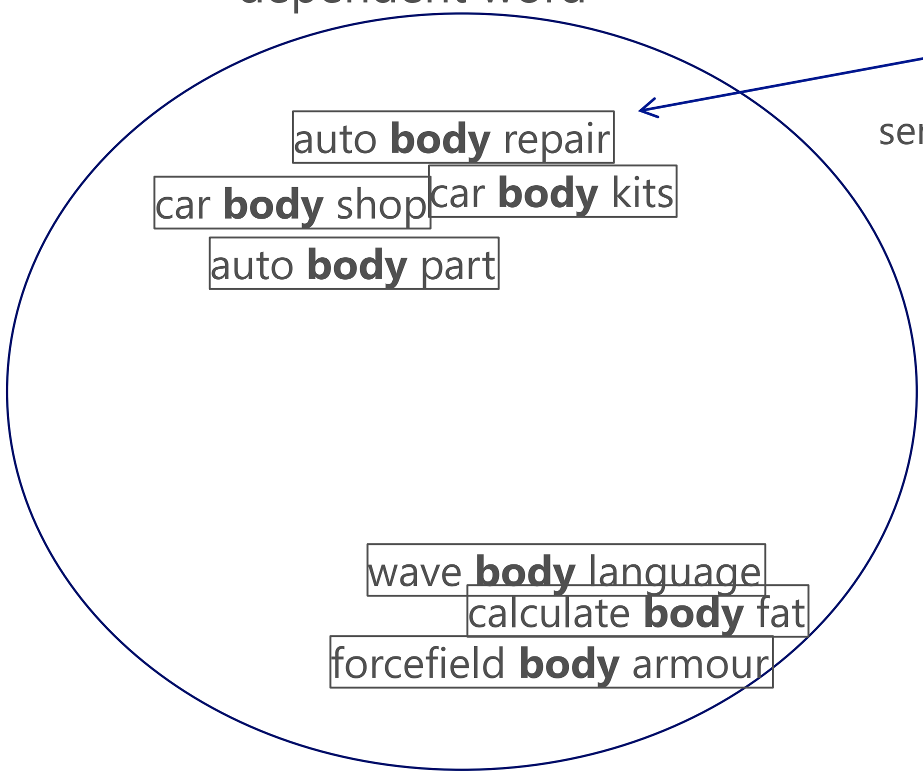
– What does the model learn at the convolutional layer?

Capture the local context dependent word sense

- Learn one embedding vector for each local context-dependent word



The embedding vector of “auto **body** repair”



semantic space

The similarity between different “***body***” within contexts

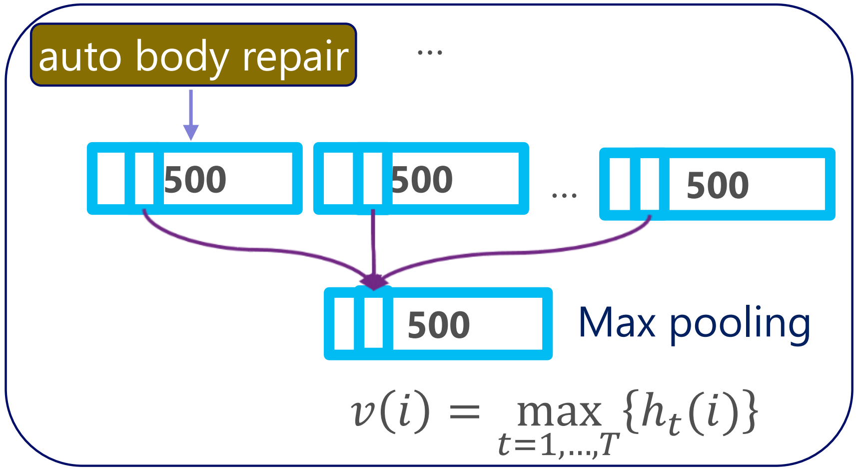
car **body** shop	**cosine similarity**
car **body** kits	0.698
auto **body** repair	0.578
auto **body** parts	0.555
wave **body** language	0.301
calculate **body** fat	0.220
forcefield **body** armour	0.165

high similarity

low similarity

CDSSM: What happens at the max-pooling layer?

- Aggregate *local topics* to form the *global intent*
- Identify salient words/phrase at the max-pooling layer



Words that win the most active neurons at the **max-pooling layers:**

auto body repair cost calculator software

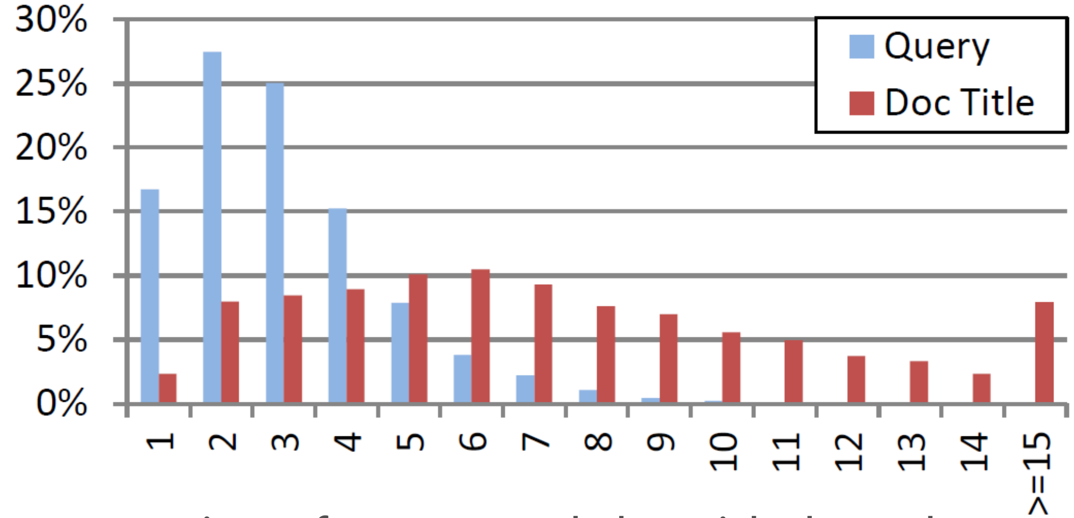
Usually, those are salient words containing clear intents/topics

NLP applications that the DSSM applies

Tasks	***X***	***Y***
Web search	*search query*	*web documents*
Ad selection	*search query*	*ad keywords*
Entity ranking	*mention (highlighted)*	*entities*
Recommendation	*doc in reading*	*interesting things / other docs*
Machine translation	*sentence in language a*	*translations in language b*
Knowledge-base construction	*entity*	*entity*
Question answering	*pattern / mention*	*relation / entity*
Semantic reasoning	*context*	*word*
Text/Image retrieval	*text*	*image*
…		

DSSM for Information Retrieval

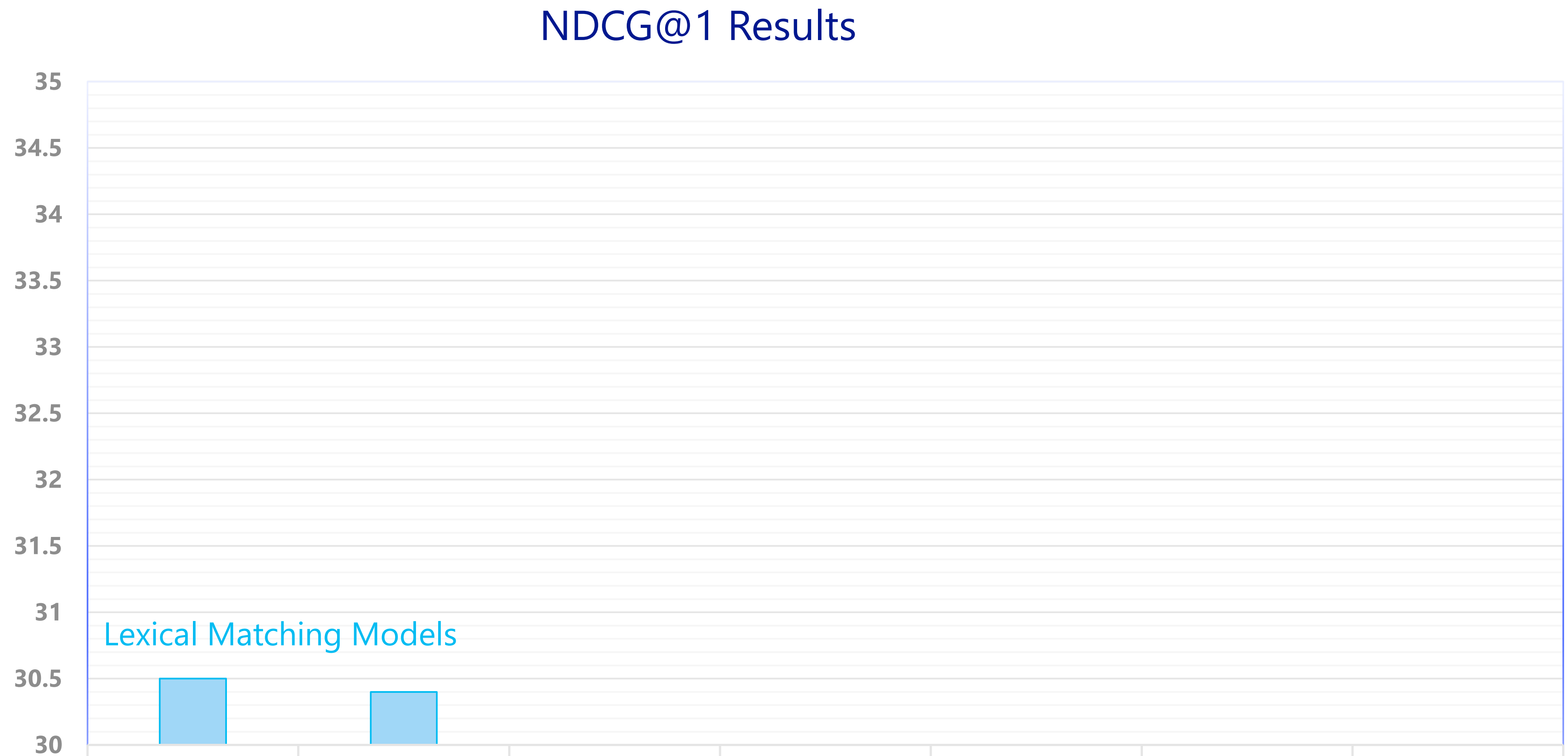
- Training Dataset
  - 30 Million (Query, Document) Click Pairs
- Testing Dataset
  - **12,071** English queries
  - around 65 web document associated to each query in average
  - Human gives each <query, doc> pair the label, with range **0 to 4**
  - 0: Bad 1: Fair 2: Good 3: Perfect 4: Excellent
- Evaluation Metric: (higher the better)
  - NDCG
- GPU (Cuda NVidia GPU K20x)



Dist. of query and doc title length

Main Experiment Results

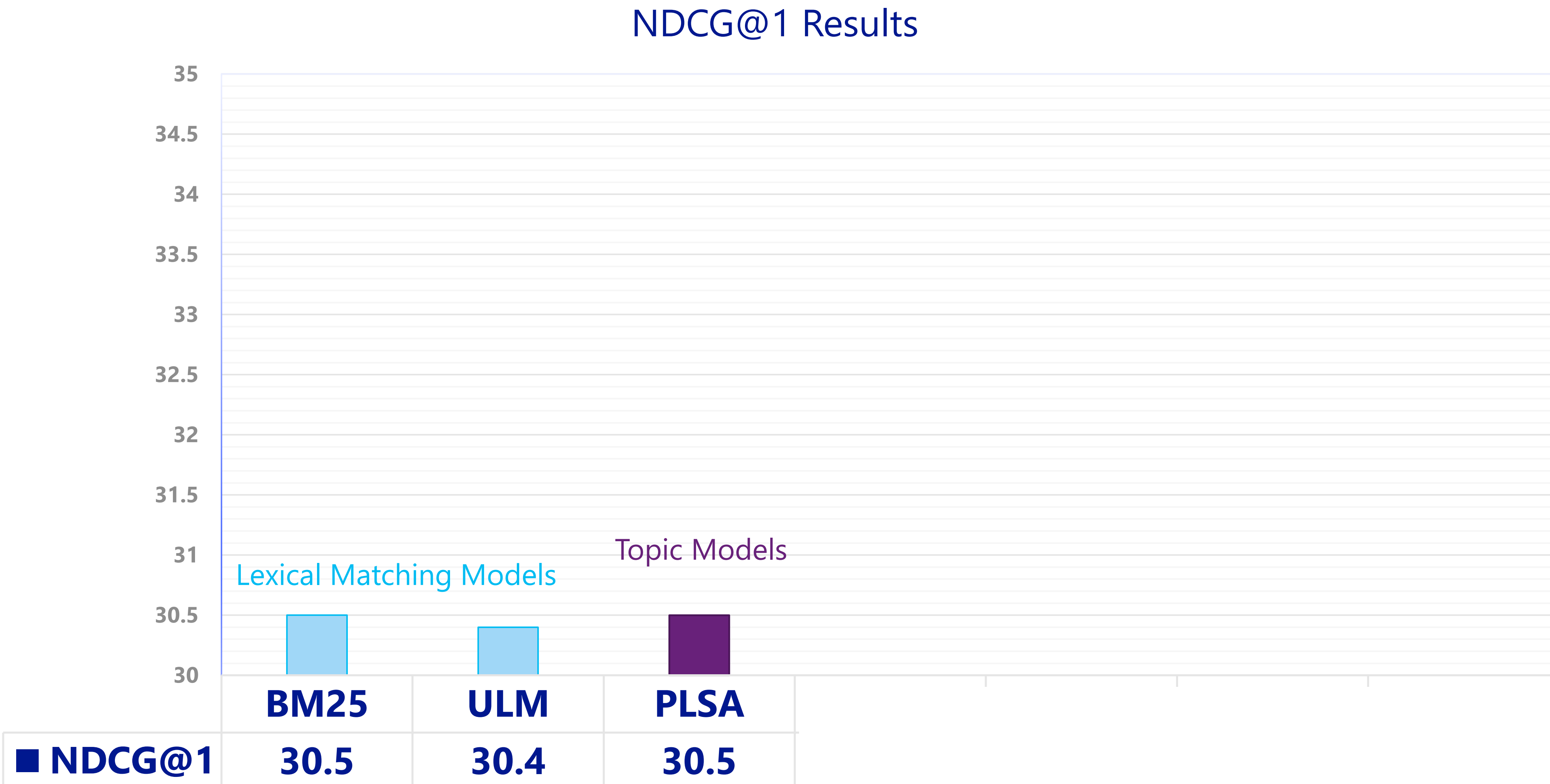
ULM : Zhai and Lafferty 2001



	BM25	ULM
NDCG@1	30.5	30.4

Main Experiment Results

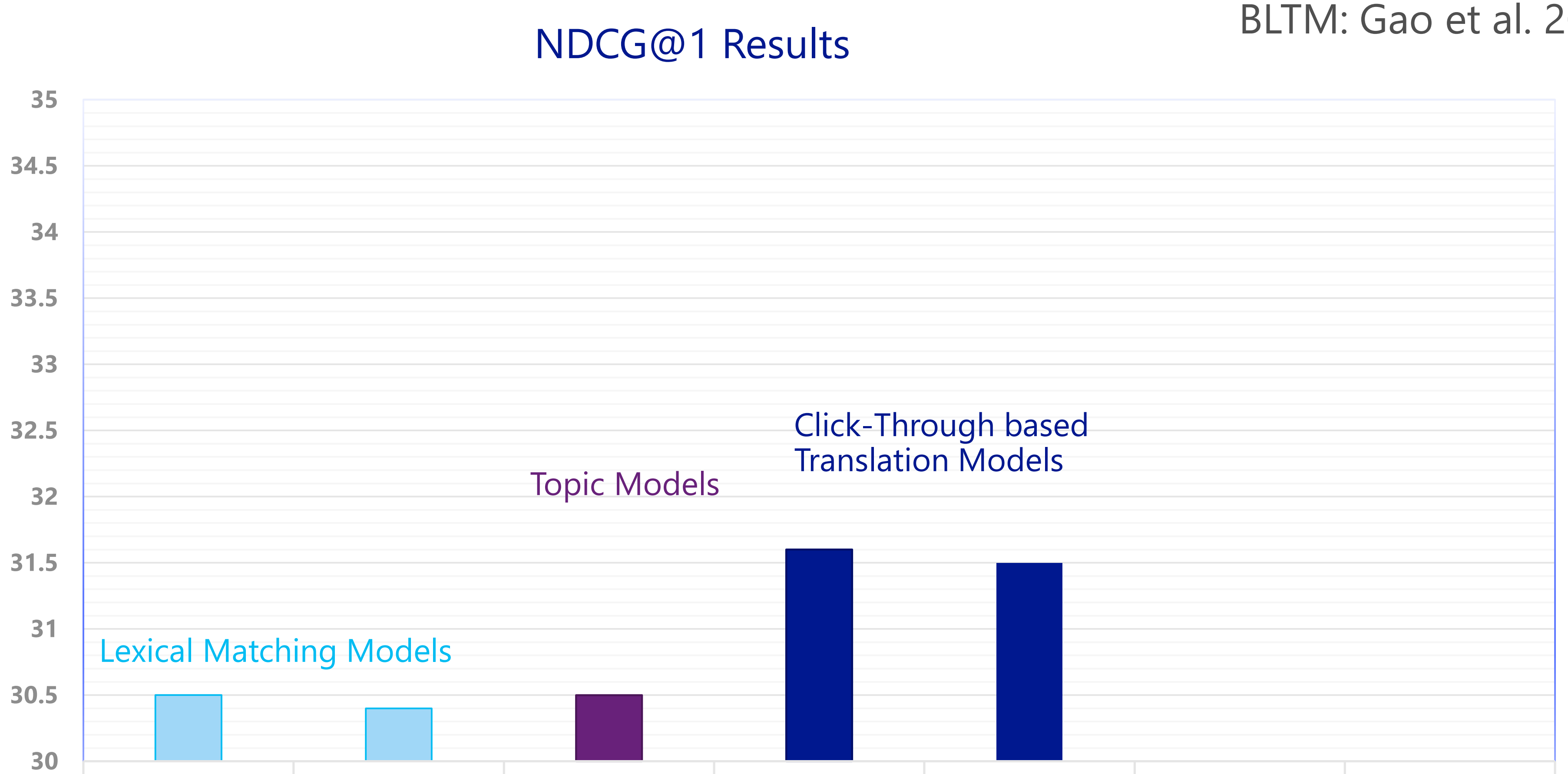
PLSA: Hofmann 1999



Main Experiment Results

WTM: Gao et al. 2010

BLTM: Gao et al. 2011

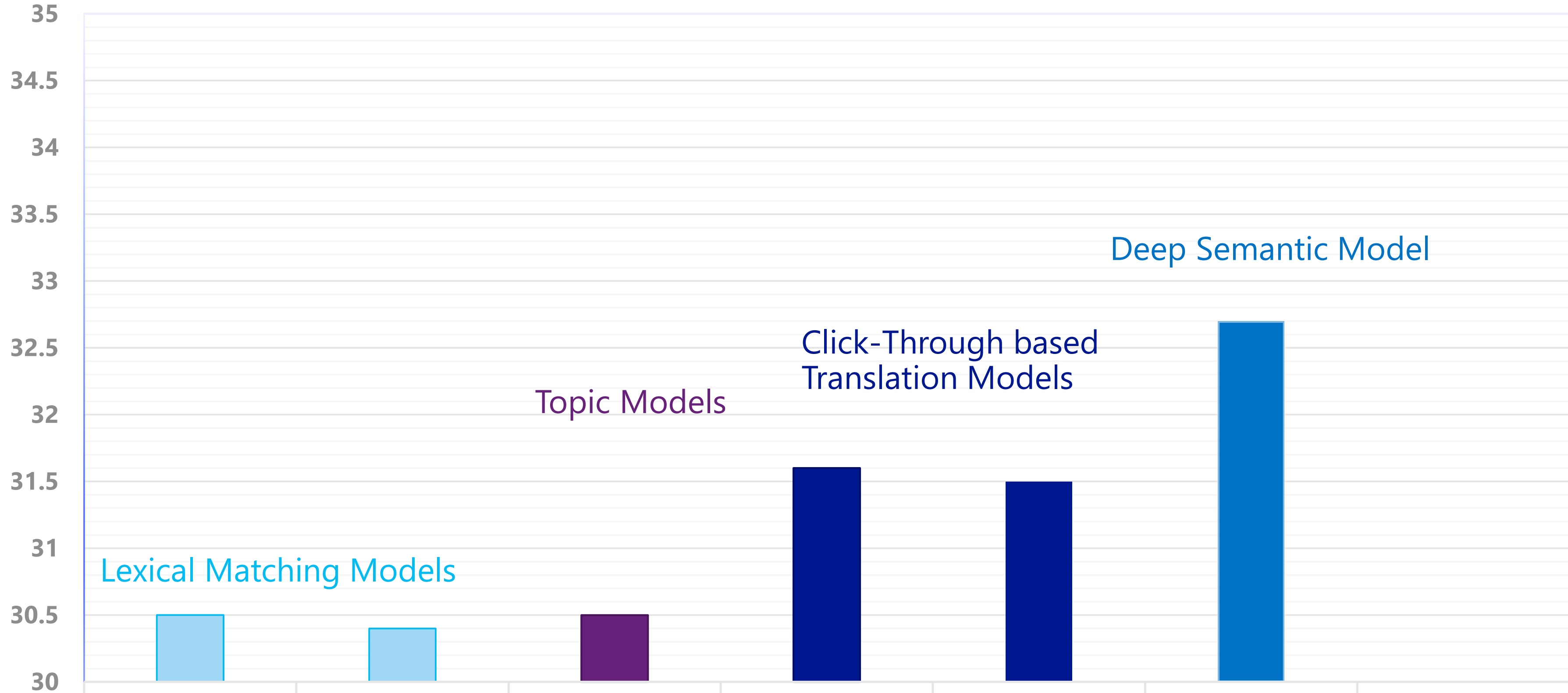


	BM25	ULM	PLSA	BLTM	WTM
NDCG@1	30.5	30.4	30.5	31.6	31.5

Main Experiment Results

DSSM: Huang et al. 2013

NDCG@1 Results

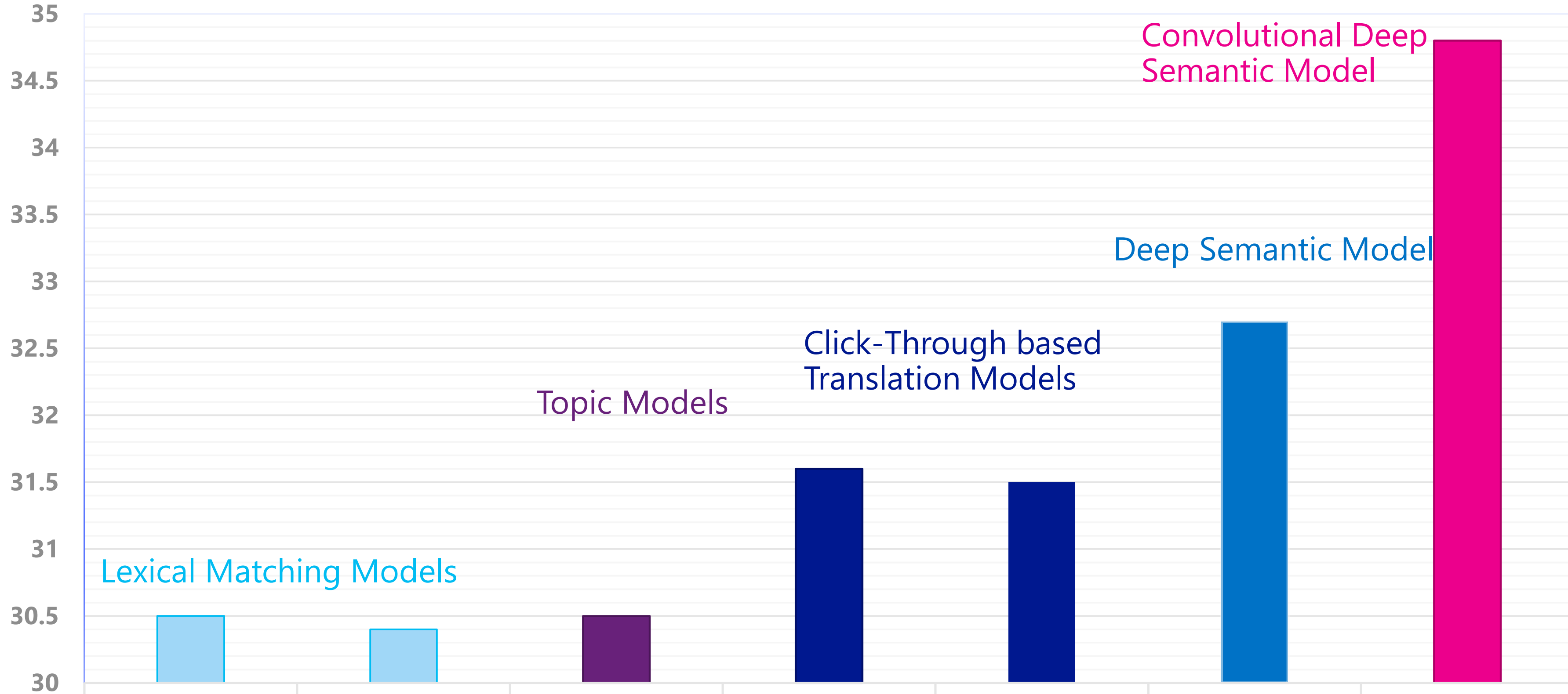


	BM25	ULM	PLSA	BLTM	WTM	DSSM
■ NDCG@1	30.5	30.4	30.5	31.6	31.5	32.7

Main Experiment Results

CDSSM: Shen et al. 2014

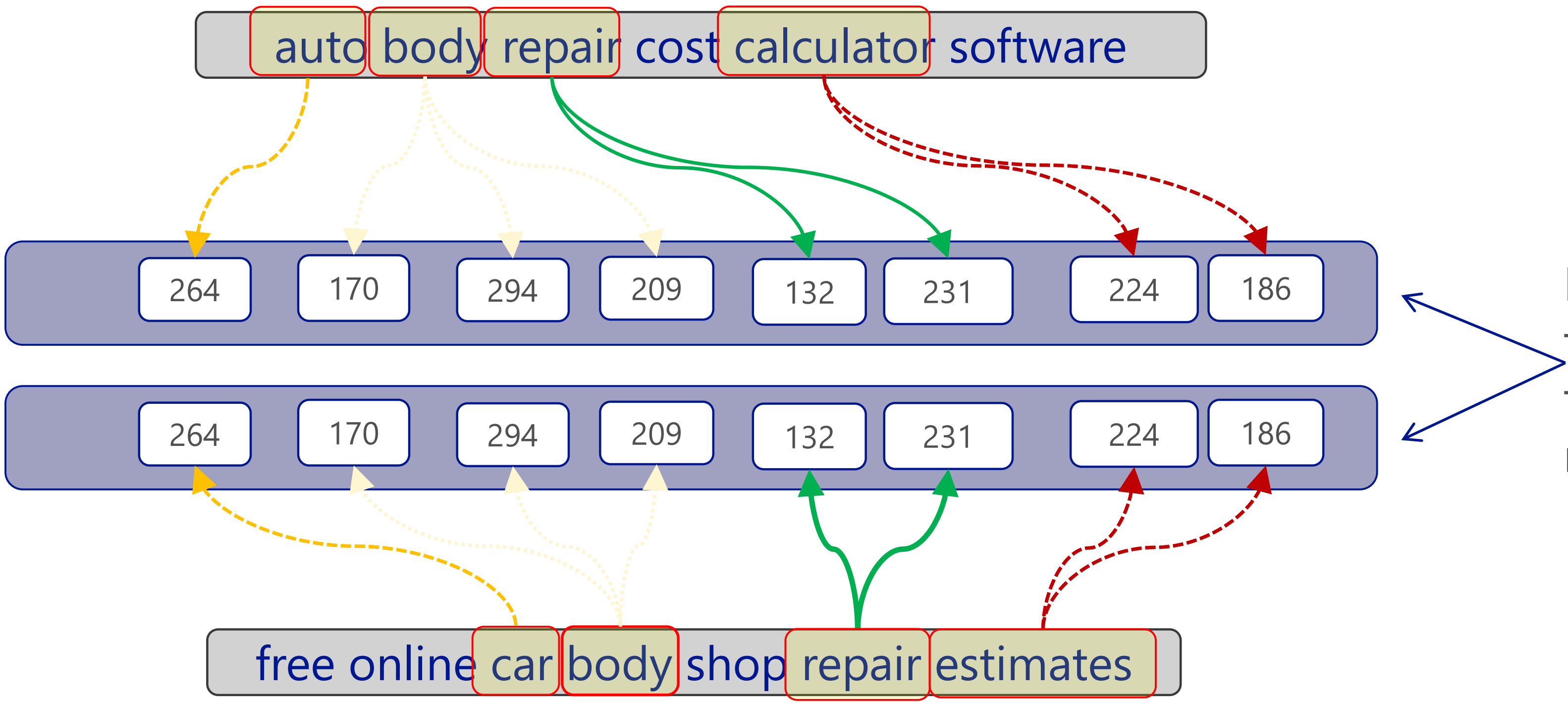
NDCG@1 Results



	BM25	ULM	PLSA	BLTM	WTM	DSSM	CDSSM
NDCG@1	**30.5**	**30.4**	**30.5**	**31.6**	**31.5**	**32.7**	**34.8**

Example: semantic matching

- Semantic matching of query and document

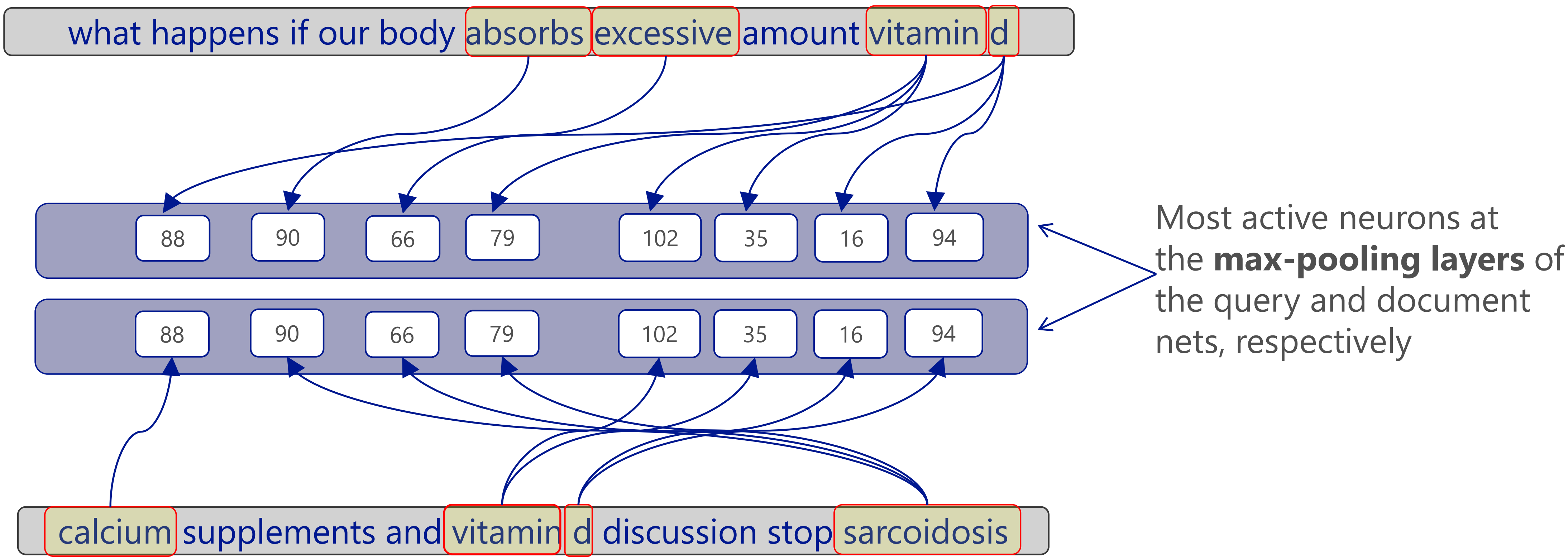


Most active neurons at the **max-pooling layers** of the query and document nets, respectively

More complex semantic matching example

sarcoidosis is a disease, a symptom is excessive amount of calcium in one's urine and blood. So medicines that increase the absorbing of calcium should be avoid. While **Vitamin d** is closely associated to **calcium absorbing**.

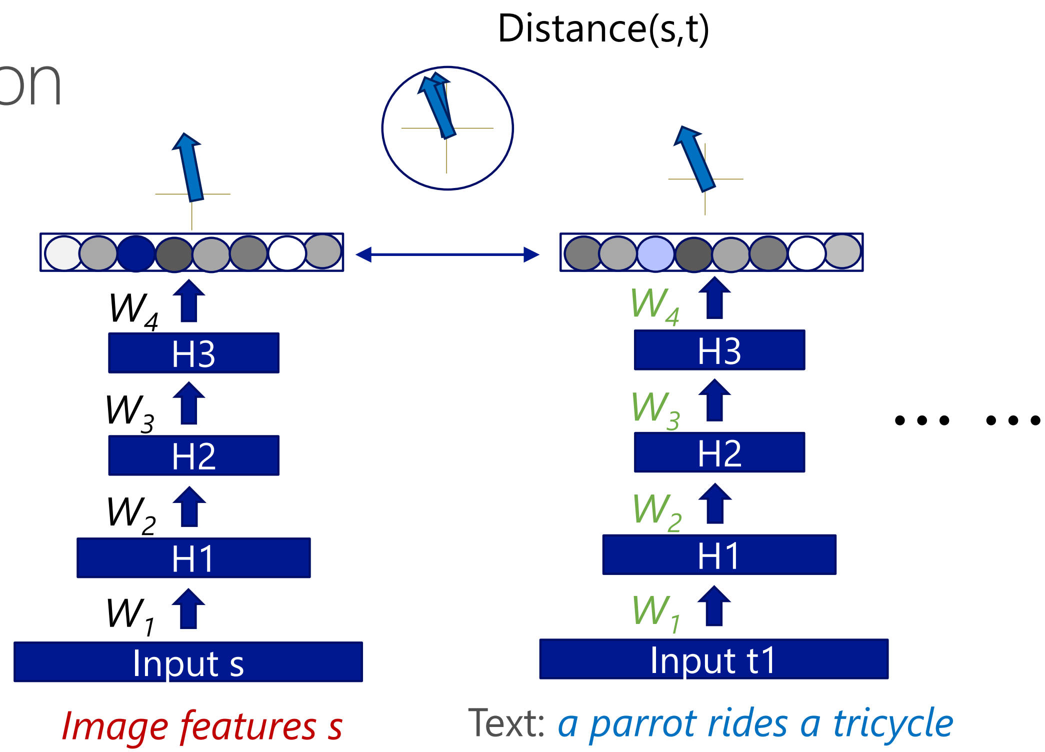
We observed that "sarcoidosis" in the document title and "absorbs" "excessive" and "vitamin (d)" in the query have high activations at neurons 90, 66, 79, indicating that the model knows that **"sarcoidosis" share similar semantic meaning with "absorbs" "excessive" "vitamin (d)", collectively**.

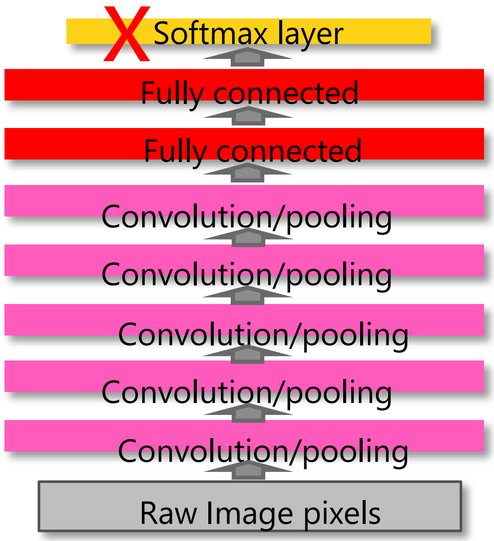


Go beyond text

DSSM for multi-modal representation learning

- Recall DSSM for text inputs: *s, t1, t2, t3, …*
- Now: replace text s by image **s**
- Using DNN/CNN features of image
- Can rank/generate text's given image or can rank images given text.





Evaluation: large scale image search

Training: 15M image/query pairs

Testing: 100K image/query pairs

Task: text query -> relevant image

Model	DCG%
Linear (e.g., DeViSE)	50.1%
Deep (img-txt DSSM)	53.9%

From captions to visual concepts and back

Detector Models, Deep Neural Net Features

Computer Vision System

street
signs
light
under
on
pole
red
stop
sign
building
bus
city
traffic

Language Model

Caption Generation System

a red stop sign sitting under a traffic light on a city street
a stop sign at an intersection on a street
a stop sign with two street signs on a pole on a sidewalk
a stop sign at an intersection on a city street
…
a stop sign
a red traffic light

DSSM Model

Global Semantic Ranking System

a stop sign at an intersection on a city street

Fang, Gupta, Iandola, Srivastava, Deng, Dollar, Gao, He, Mitchell, Platt, Zitnick, Zweig, "From captions to visual concepts and back," on arXiv

Evaluation: *How far are we from human?*

Training: 400K image/caption pairs as training data

Testing: 20K images, 5 annotators providing 5 captions per image

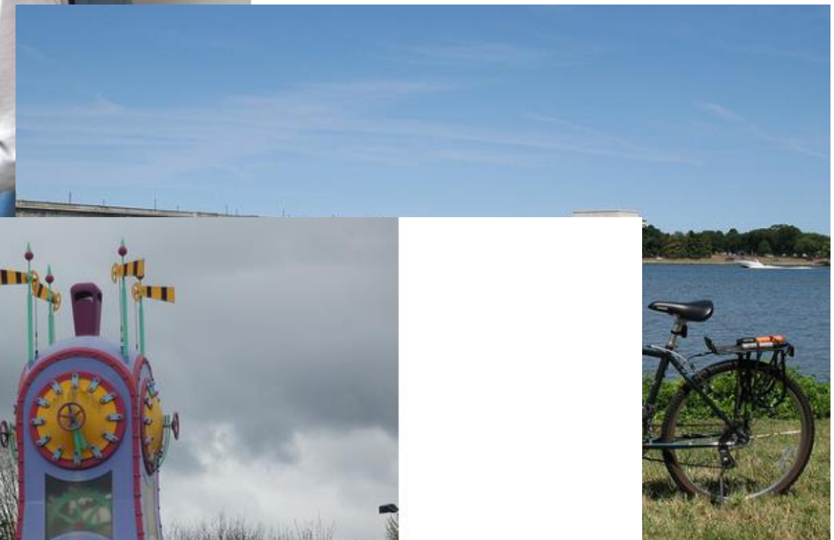
Hold 1 human as the control system

The other 4 annotations are gold reference for BLEU testing

Entry	BLEU % on 4-ref (higher the better)	Equal to or better than human annotation *
Human (control)	19.3	
Machine	21.1	23.3%

* The percentage that the judgers think the machine's output is equal to or better than the human's annotation.

http://blogs.technet.com/b/machinelearning/archive/2014/11/18/rapid-progress-in-automatic-image-captioning.aspx



next to a

next to a

Machine-generated (but turker prefered)	a group of motorc… to a motorcycle
Human-annotated (but turker not prefered)	two girls wearing skirts and one of … motorcycle while … nearby

Machine-generated (but turker prefered)	a man holding a tennis racquet on a tennis court
Human-annotated (but turker not prefered)	the man is on the tennis court playing a game

…rated (but …)	a clock tower in the middle of the street
Human-annotated (but turker not prefered)	a statue with a clock on it near a parking lot

Other models for sentence-level representation

Long short-term memory RNN (LSTM-RNN)

- Model long-span dependency (Hochreiter and Schmidhuber. Neural Computation, 1997)
- LSTM for IR (Palangi, et al., "Learning sequential semantic representations," to appear)
- LSTM for MT (Sutskever, et al., "Sequence to sequence learning with neural networks," NIPS14)

Recursive NN (ReNN)

- Model the hierarchical structure of nature language
- ReNN for parsing (Socher et al., "Parsing natural scenes and natural language with recursive neural networks", 2011)

Tensor product representation (TPR)

- Efficient representation of the structure of natural language
- Smolensky & Legendre: The Harmonic Mind, From Neural Computation to Optimality-Theoretic Grammar, MIT Press, 2006

Interim summary

Exciting advances in learning continuous semantic space

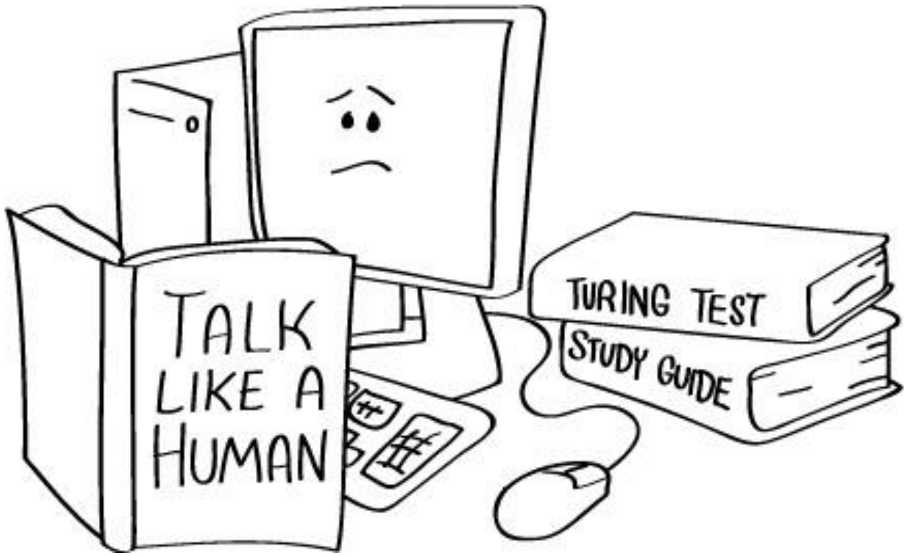
- deep models effectively learn semantic representation vectors
- leads to superior performance in a range of NL tasks
- facilitates cross-modality learning
  learning image and text vectors in an joint semantic space

Part IV

Natural Language Understanding

Natural Language Understanding

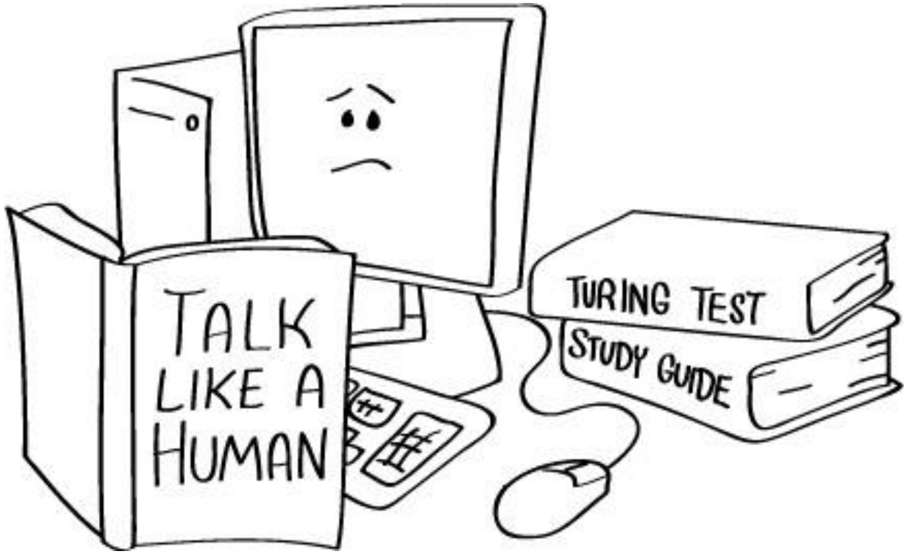
- Build an intelligent system that can interact with human using natural language
- Research challenge
  - **Meaning representation of text**
  - **Support useful inferential tasks**



http://csunplugged.org/turing-test

Natural Language Understanding

- Continuous Word Representations & Lexical Semantics
  - Language is compositional
  - Word is the basic semantic unit
- Knowledge Base Embedding
- Semantic Parsing & Question



http://csunplugged.org/turing-test

Continuous Word Representations

- A lot of popular methods for creating word vectors!
  - Vector Space Model [Salton & McGill 83]
  - Latent Semantic Analysis [Deerwester+ 90]
  - Brown Clustering [Brown+ 92]
  - Latent Dirichlet Allocation [Blei+ 01]
  - Deep Neural Networks [Collobert & Weston 08]
  - Word2Vec [Mikolov+ 13]
- Encode term co-occurrence information
- Measure semantic similarity well

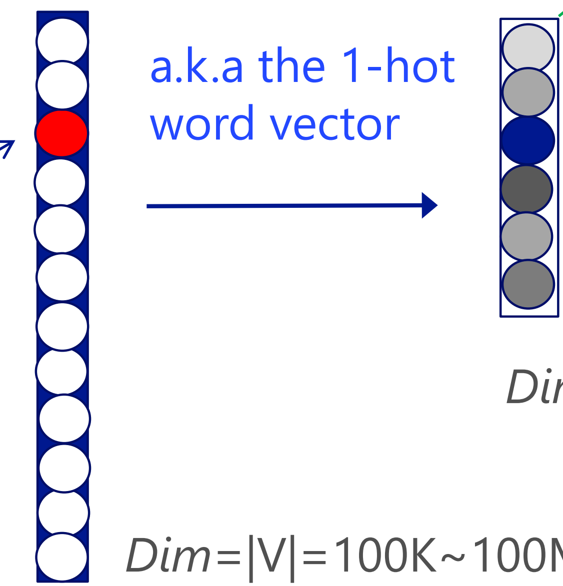
Semantic embedding

Project raw text into a continuous semantic space

e.g., word embedding

Captures the word meaning in a semantic space

$f(\boldsymbol{cat}) =$



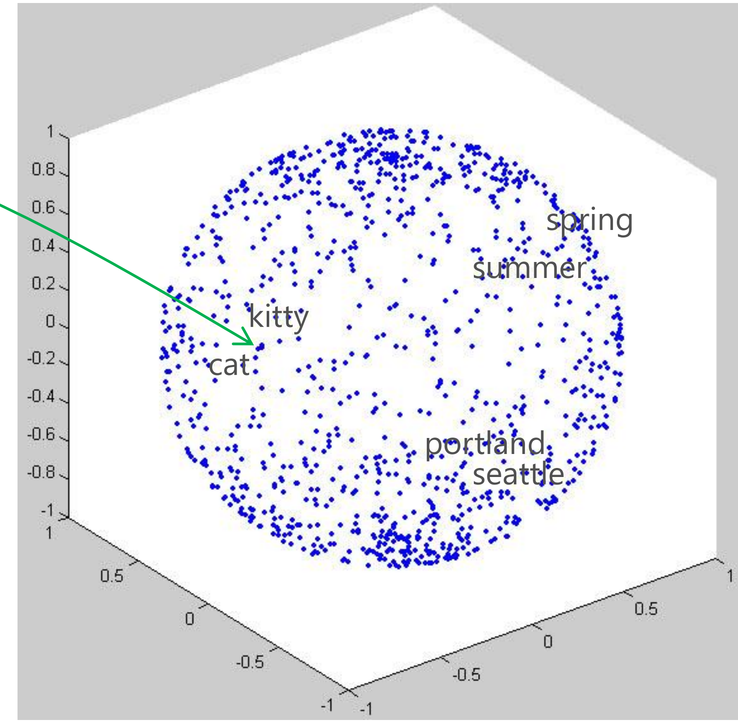
a.k.a the 1-hot word vector

word embedding vector in the semantic space

The index of "cat" in the vocabulary

Dim=100~1000

Dim=|V|=100K~100M



Deerwester, Dumais, Furnas, Landauer, Harshman, "Indexing by latent semantic analysis," JASIS 1990

SENNA word embedding

Scoring:

$$Score(w_1, w_2, w_3, w_4, w_5) = U^T \sigma(W[f_1, f_2, f_3, f_4, f_5] + b)$$

Training:

$$J = \max(0, 1 + S^- - S^+)$$

Update the model until $S^+ > 1 + S^-$

Where

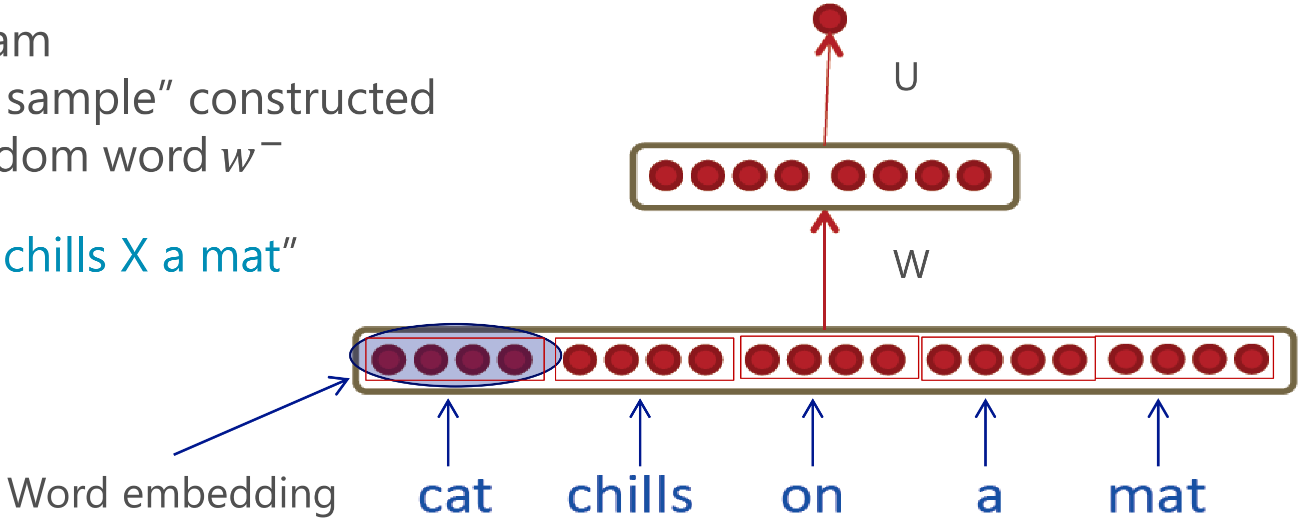
$$S^+ = Score(w_1, w_2, w_3, w_4, w_5)$$
$$S^- = Score(w_1, w_2, w^-, w_4, w_5)$$

And

$< w_1, w_2, w_3, w_4, w_5 >$ is a valid 5-gram

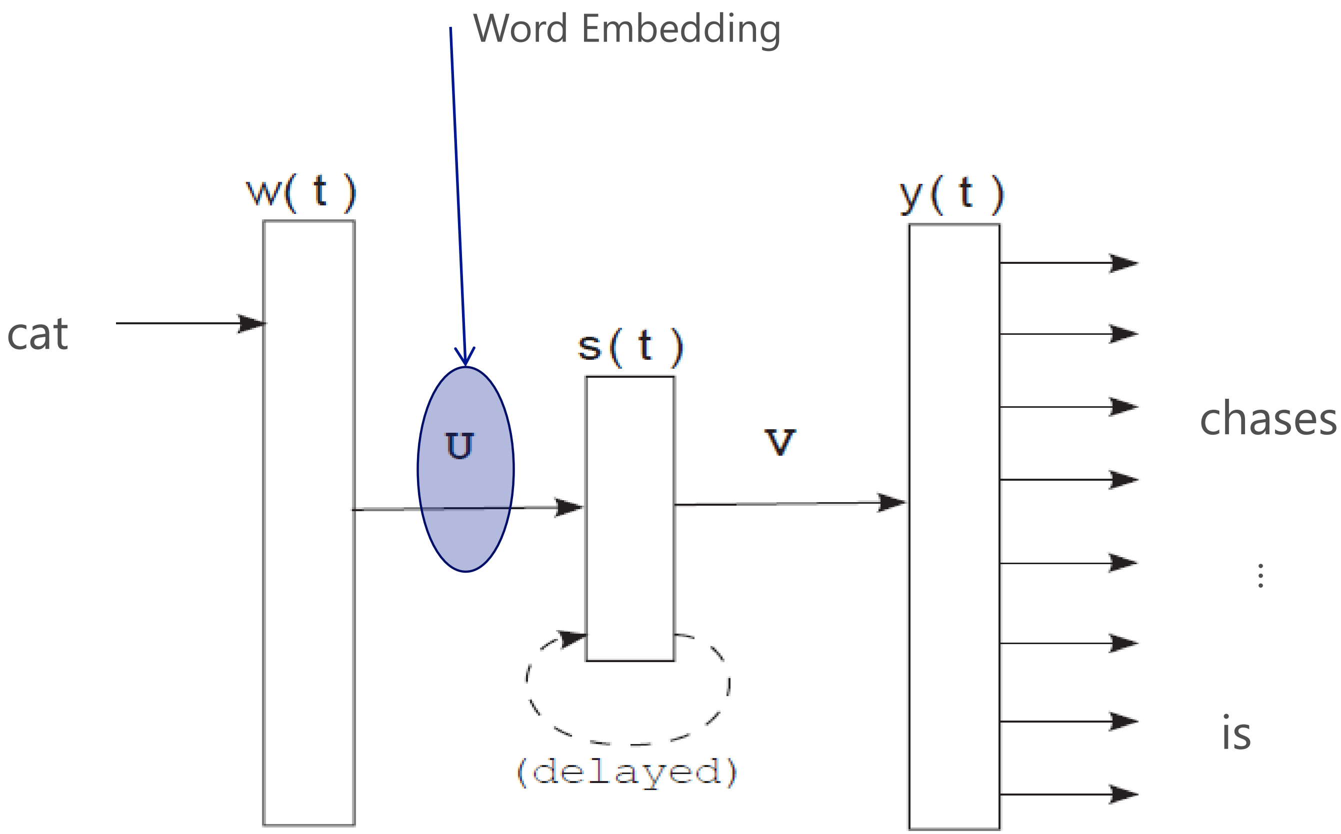
$< w_1, w_2, w^-, w_4, w_5 >$ is a "negative sample" constructed by replacing the word w_3 with a random word w^-

e.g., a negative example: "cat chills X a mat"



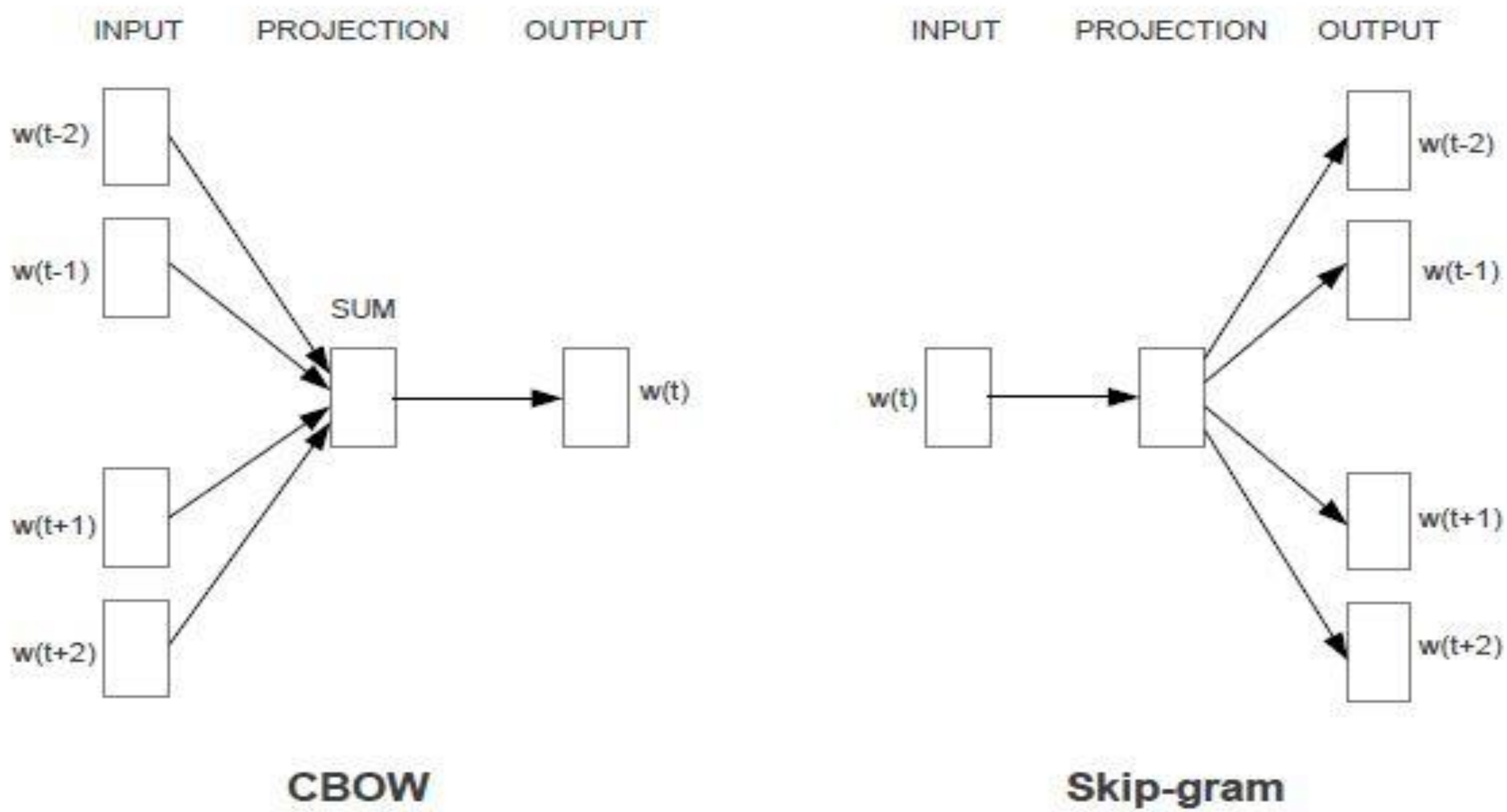
Collobert, Weston, Bottou, Karlen, Kavukcuoglu, Kuksa, "Natural Language Processing (Almost) from Scratch," JMLR 2011

RNN-LM base word embedding



Mikolov, Yih, Zweig, "Linguistic Regularities in Continuous Space Word Representations," NAACL 2013

CBOW/Skip-gram Word Embeddings



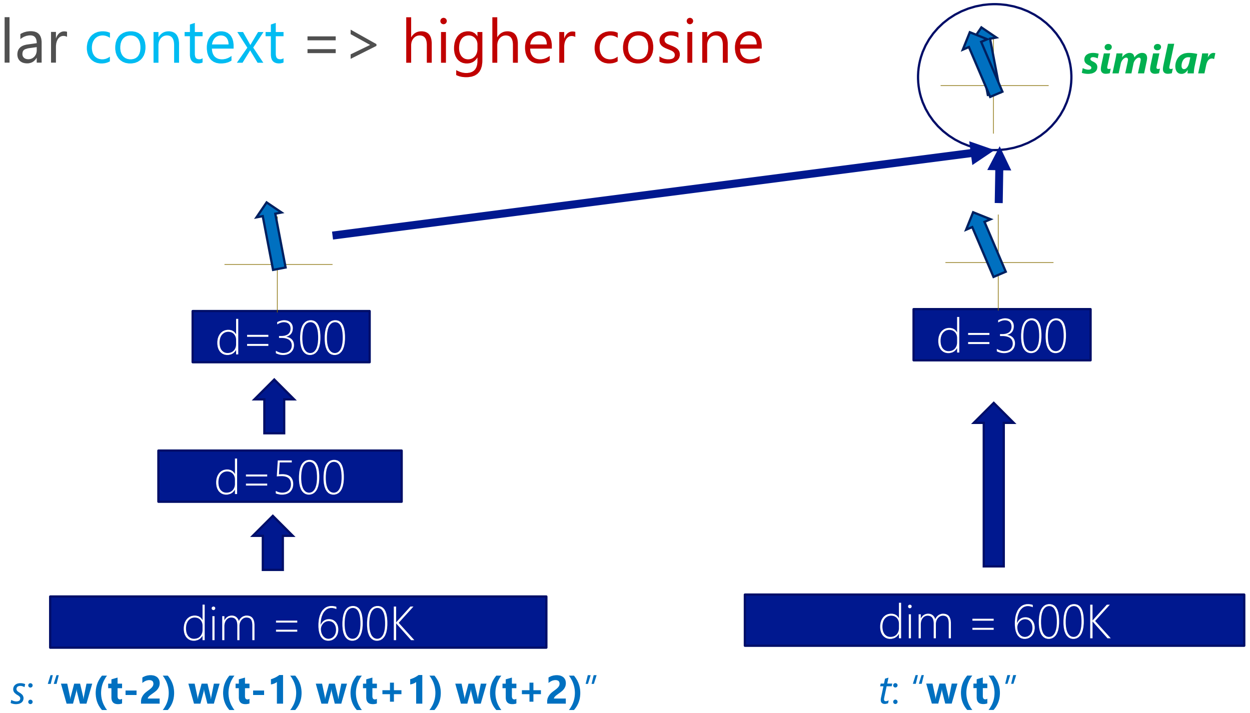
Continuous Bag-of-Words

The CBOW architecture (a) on the left, and the Skip-gram architecture (b) on the right. [Mikolov et al., 2013 ICLR].

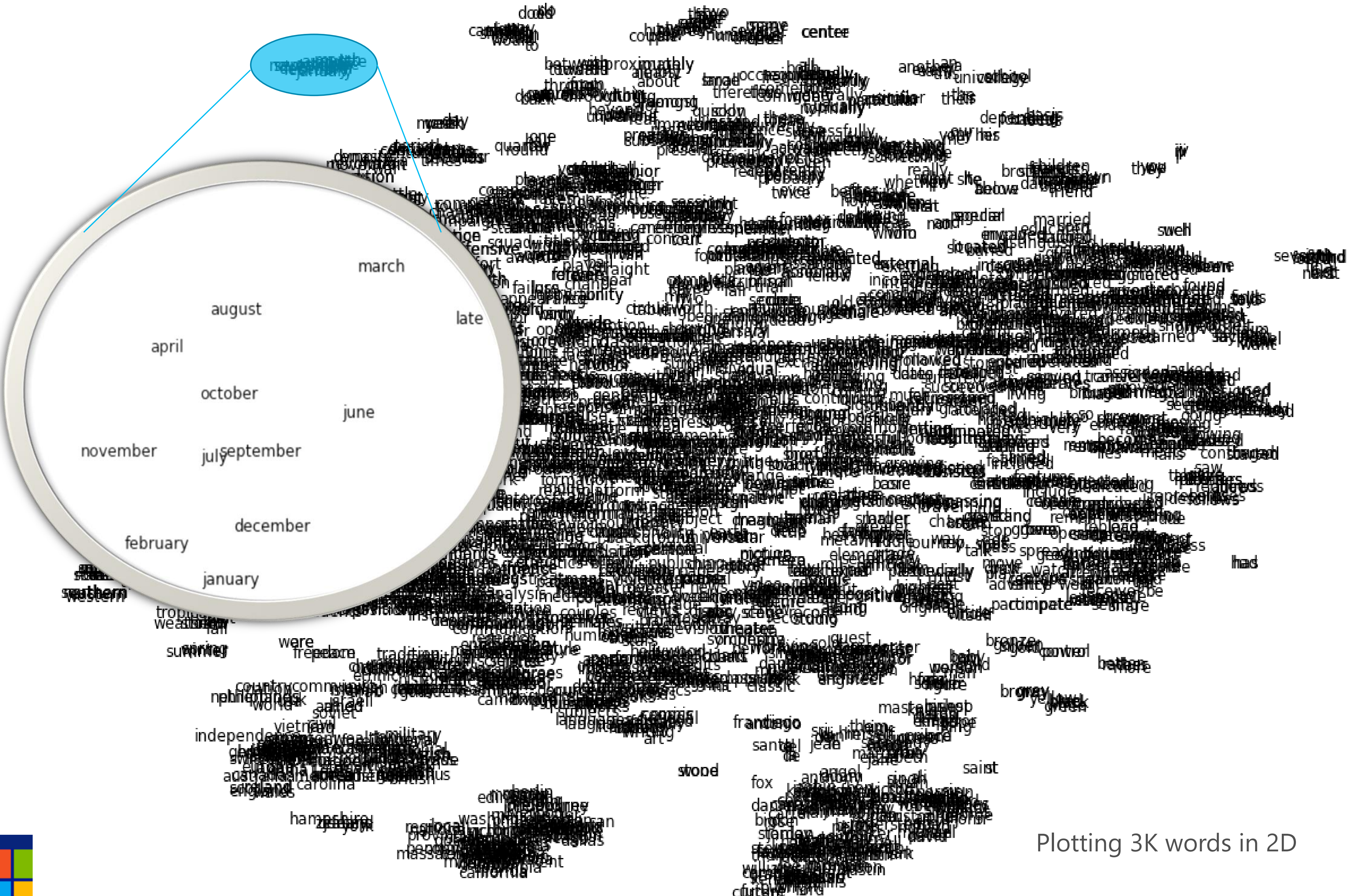
DSSM: learning words' meaning

- Learn a word's semantic meaning by means of its neighbors (context)
  - Construct context <-> word training pair for DSSM
  - Similar words with similar context => higher cosine
- Training Condition:
  - 600K vocabulary size
  - 1B words from Wikipedia
  - 300-dimentional vector

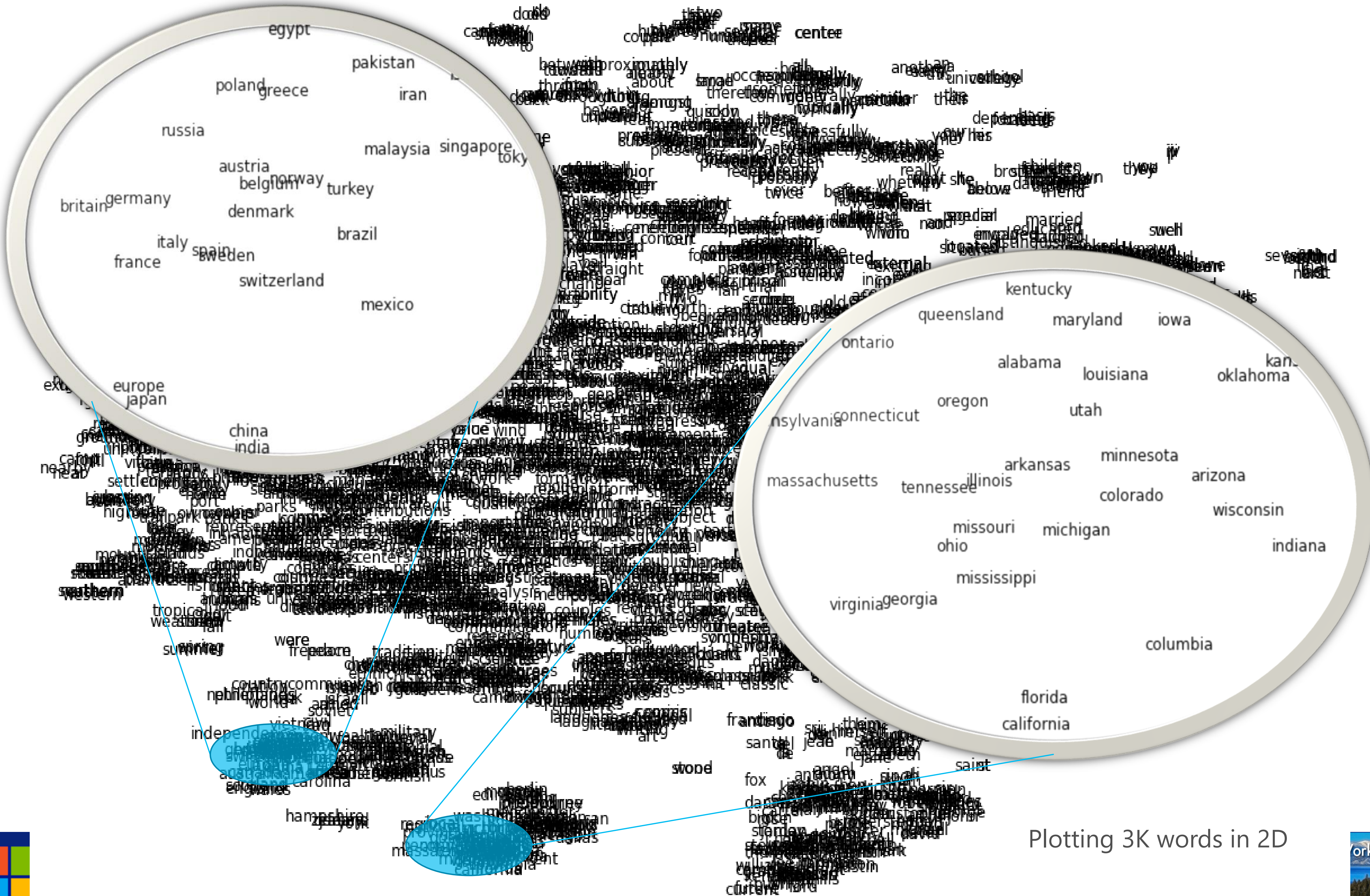
You shall know a word by the company it keeps (J. R. Firth 1957: 11)



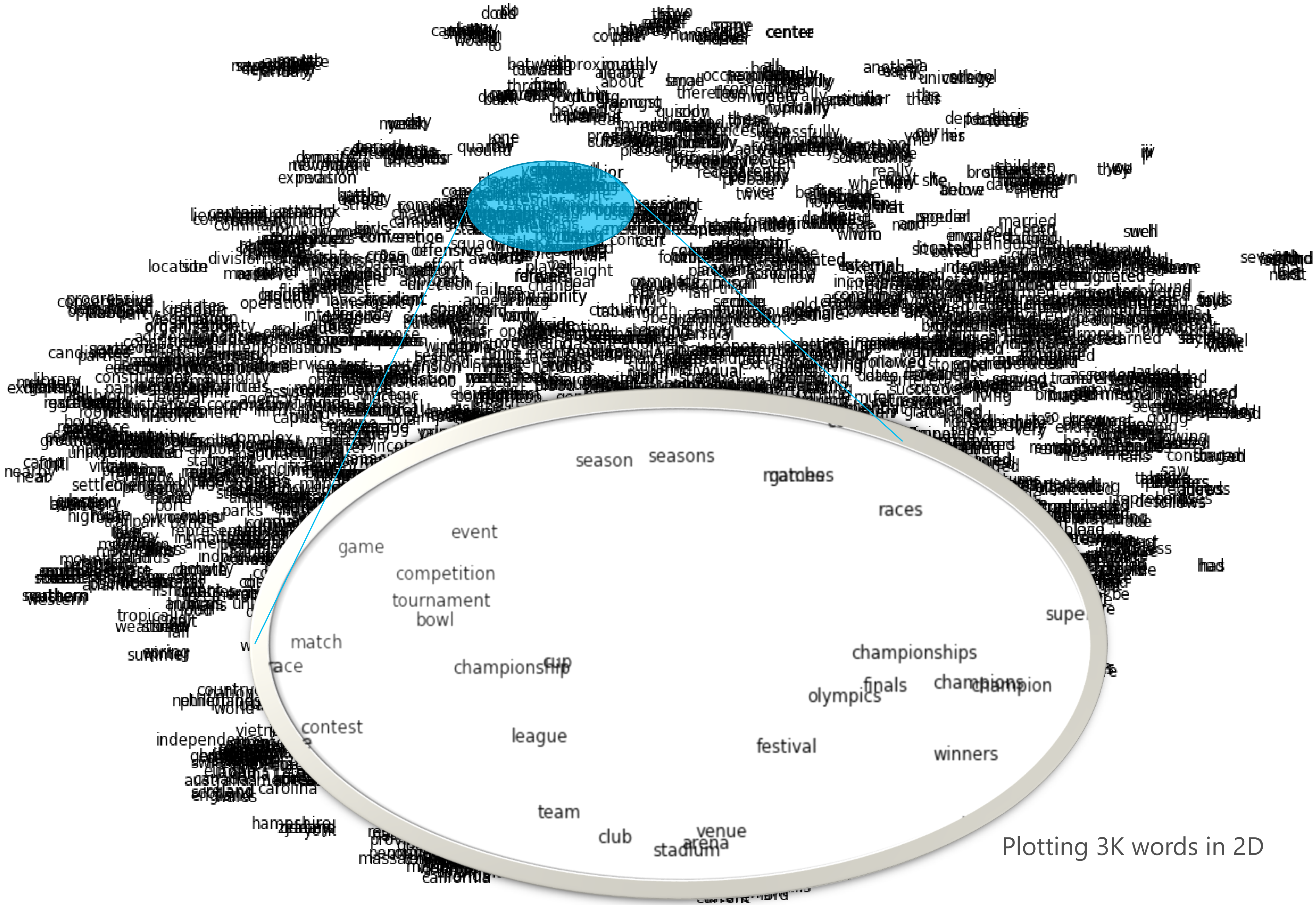
[Song, He, Gao, Deng, 2014]



Plotting 3K words in 2D

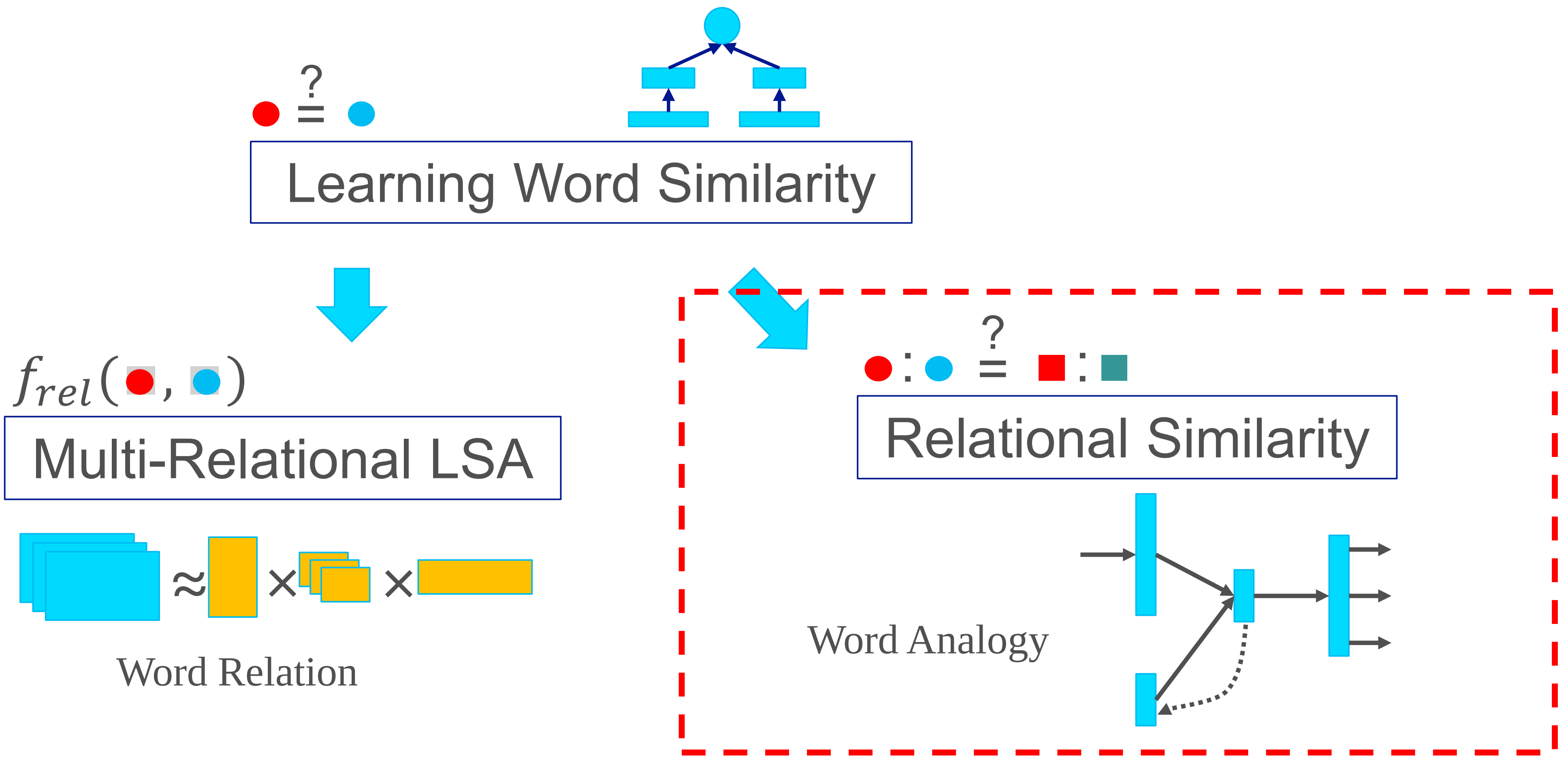


Plotting 3K words in 2D



Plotting 3K words in 2D

Relational Similarity (Word Analogy)



king : queen $\stackrel{?}{=}$ man : woman

Measuring Relational Similarity

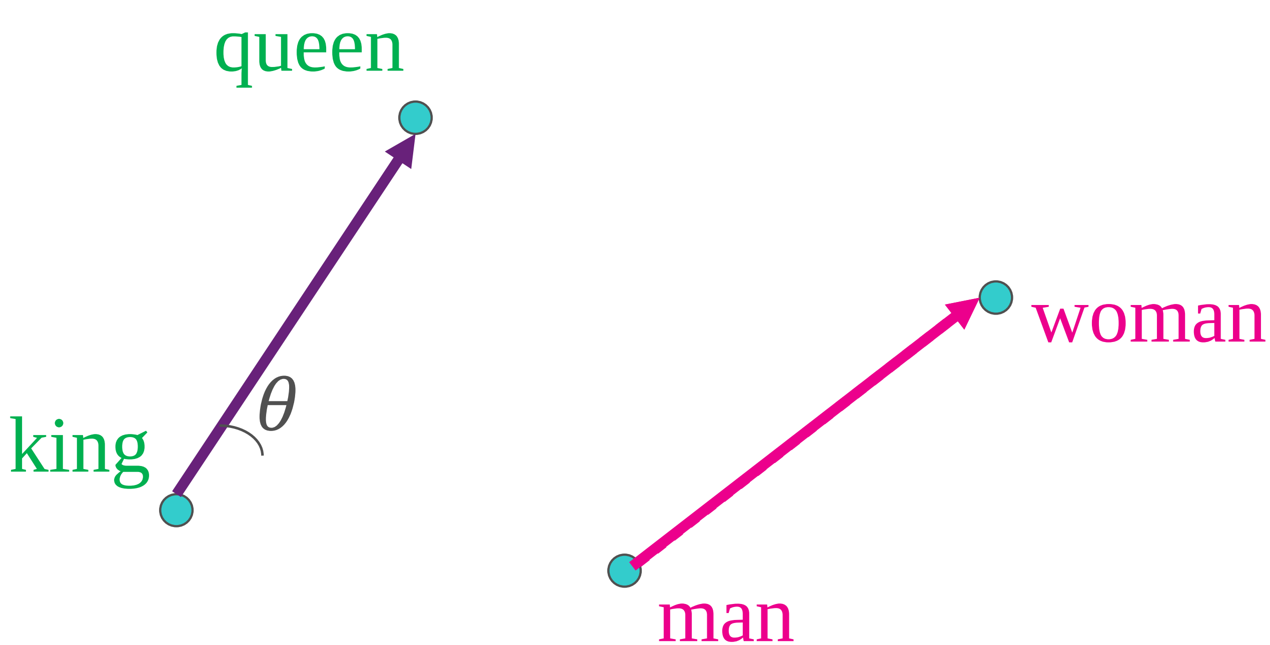
- Determine whether two pairs of words have the same relation (the "analogy" problem) [Bejar et al. '91]
  - (silverware : fork) vs. (clothing : shirt) [singular collective]
  - (coast : ocean) vs. (sidewalk : road) [contiguity]
  - (psychology : mind) vs. (astronomy : stars) [knowledge]

- Why it's useful?

  Building a general "relational similarity" model is a more efficient way to learn a model for any arbitrary relation [Turney, 2008]

Unexpected Finding: Directional Similarity

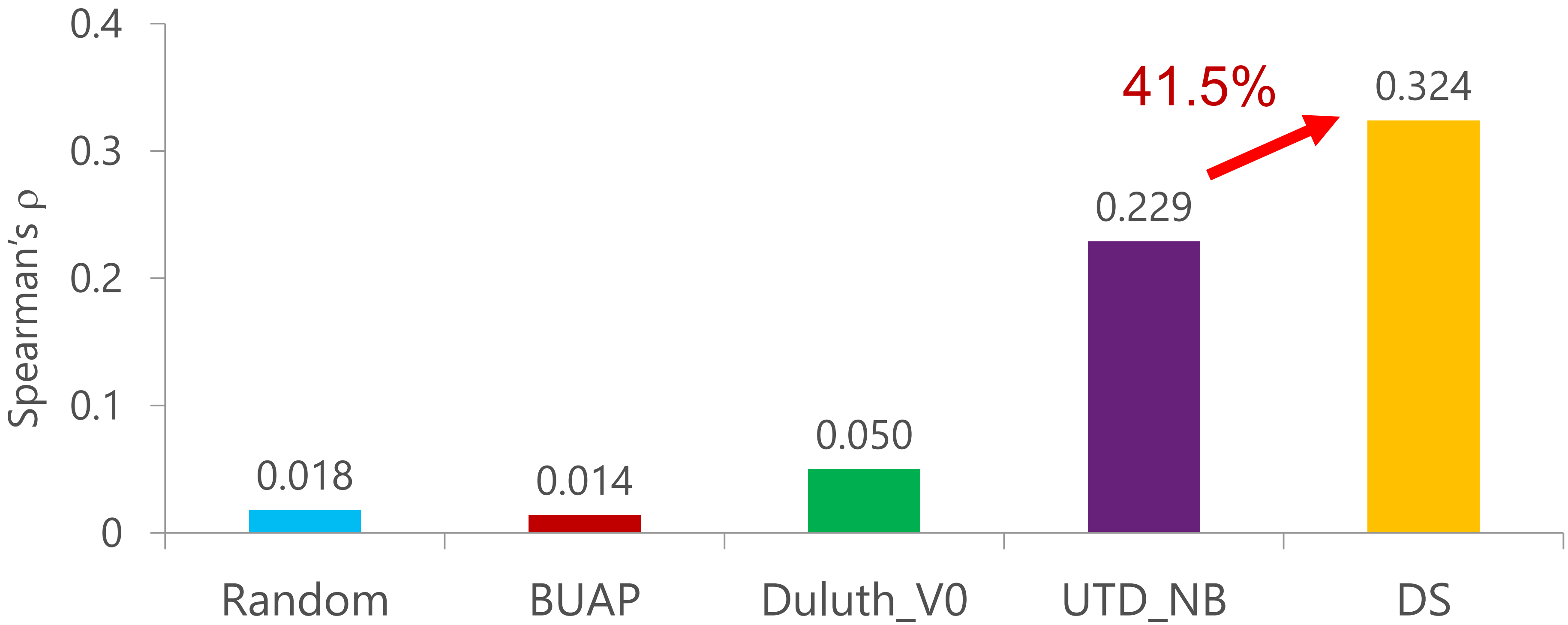
- Word embedding taken from recurrent neural network language model (RNN-LM) [Mikolov 2011]



- Relational similarity is derived by the cosine score

Experimental Results

- SemEval-2012 Task 2 – Relational Similarity
  - Rank word pairs of 69 testing relations
  - Evaluate model by its correlation to human judgments



Similar Results Observed on Other Datasets

- MSR syntactic test set [Mikolov+ 2013]
  - see : saw = return : returned
  - better : best = rough : roughest

- Semantic-Syntactic word relationship [Mikolov+ 2013]
  - Athens : Greece = Oslo : Norway
  - brother : sister = grandson : granddaughter
  - apparent : apparently = rapid : rapidly

Evaluation on Word Analogy

The dataset contains 19,544 word analogy questions:

Semantic questions, e.g.,: “Athens is to Greece as Berlin is to ?”

Syntactic questions, e.g.,: “dance is to dancing as fly is to ?”

Model	**Dim**	**Size**	**Accuracy Avg.(sem+syn)**
SG	300	1B	61.0%
CBOW	300	1.6B	36.1%
vLBL	300	1.5B	60.0%
ivLBL	300	1.5B	64.0%
GloVe	300	1.6B	70.3%
DSSM	300	1B	71.9%

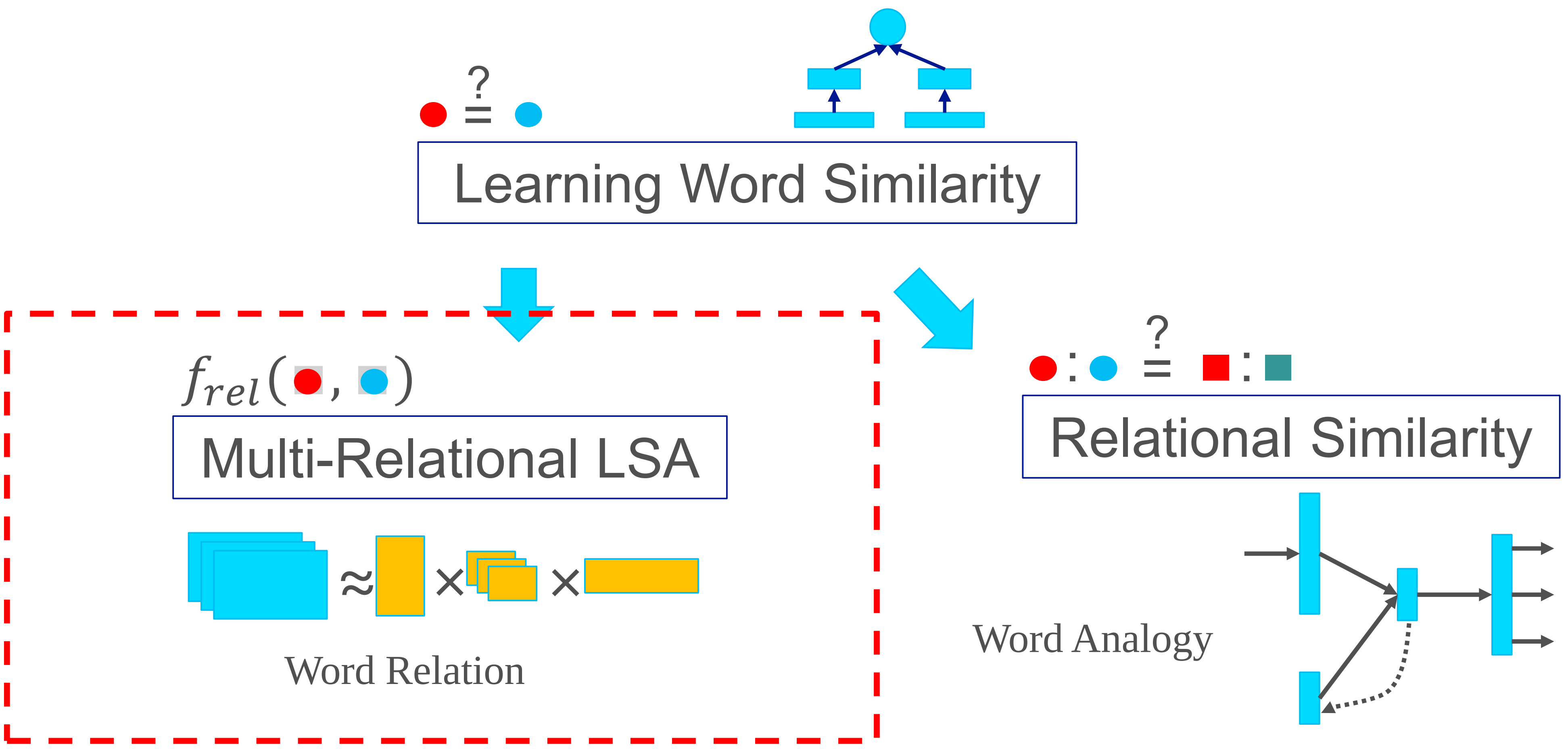
(i)vLBL results are from (Mnih et al., 2013); skip-gram (SG) and CBOW results are from (Mikolov et al., 2013a,b); GloVe are from (Pennington, Socher, and Manning, EMNLP2014)

Discussion

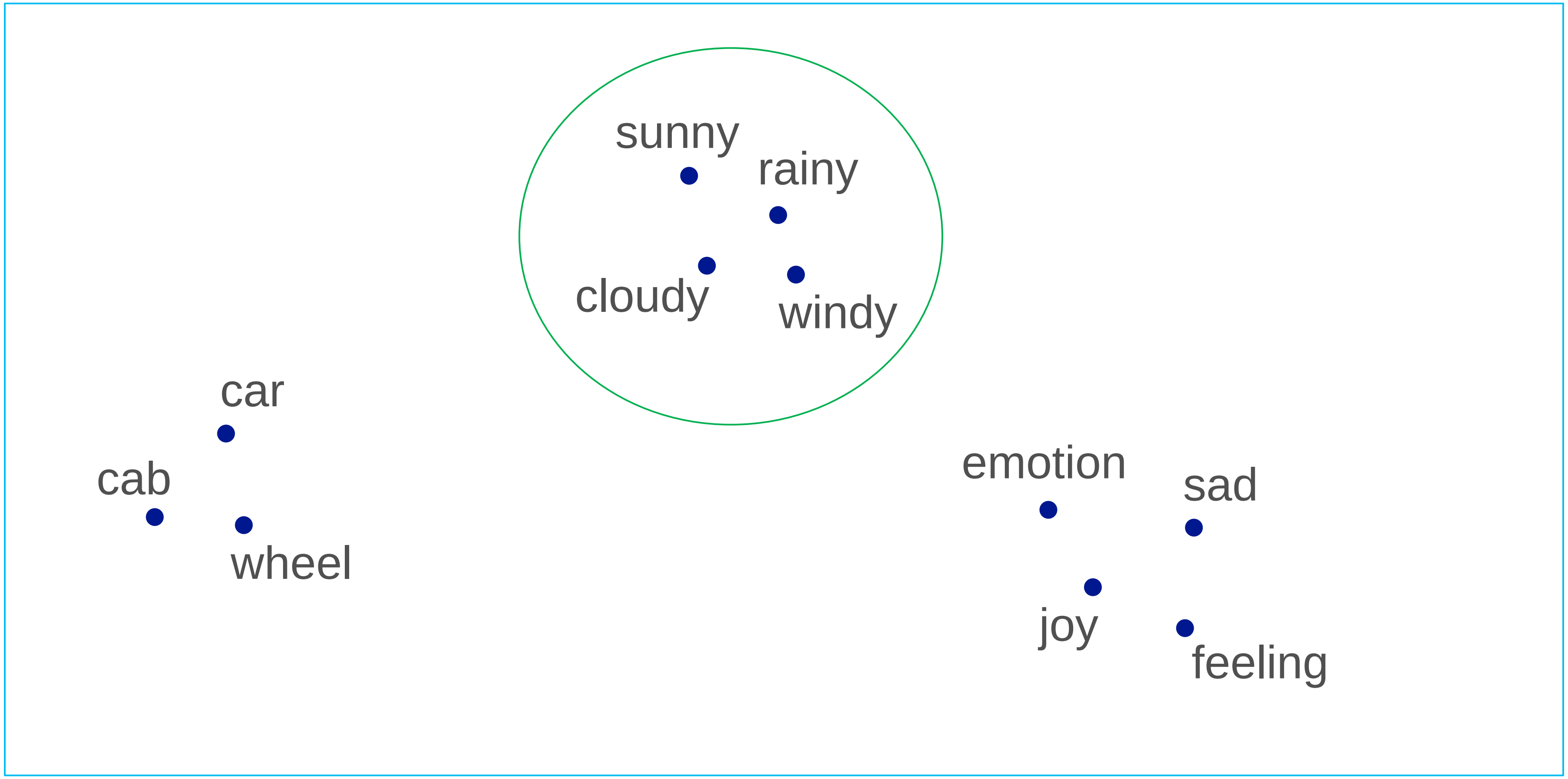
- Directional Similarity cannot handle symmetric relations
  - good : bad = bad : good
- Vector arithmetic = Similarity arithmetic
  [Levy & Goldberg CoNLL-14]
- Find the closest x to $king - man + woman$ by

$$\arg\max_x(\cos(x, king - man + woman)) =$$
$$\arg\max_x(\cos(x, king) - \cos(x, man) + \cos(x, woman))$$

Lexical Semantics (Word Relations)



Continuous Semantic Representations



Semantics Needs More Than Similarity

Tomorrow will be rainy.

Tomorrow will be sunny.

$\boxed{similar(\text{rainy, sunny})?}$

$\boxed{antonym(\text{rainy, sunny})?}$

Leverage Linguistic Knowledge Bases

- Can't we just use the existing linguistic KBs?
  - Knowledge in these resources is never complete
  - Often lack of degree of relations
- Create a continuous semantic representation that
  - Leverages existing rich linguistic knowledge bases
  - Discovers new relations
  - Enables us to measure the degree of multiple relations (not just similarity)

Roadmap

- Background:
  Latent Semantic Analysis (LSA)
- Two opposite relations:
  Polarity Inducing Latent Semantic Analysis (PILSA)
- More relations:
  Multi-Relational Latent Semantic Analysis (MRLSA)

Latent Semantic Analysis [Deerwester+ 1990]

- Data representation
  - Encode single-relational data in a matrix
    - Co-occurrence (e.g., from a general corpus)
    - Synonyms (e.g., from a thesaurus)
- Factorization
  - Apply SVD to the matrix to find latent components
- Measuring degree of relation
  - Cosine of latent vectors

Encode Synonyms in Matrix

- Input: Synonyms from a thesaurus
- Joyfulness: joy, gladden
- Sad: sorrow, sadden

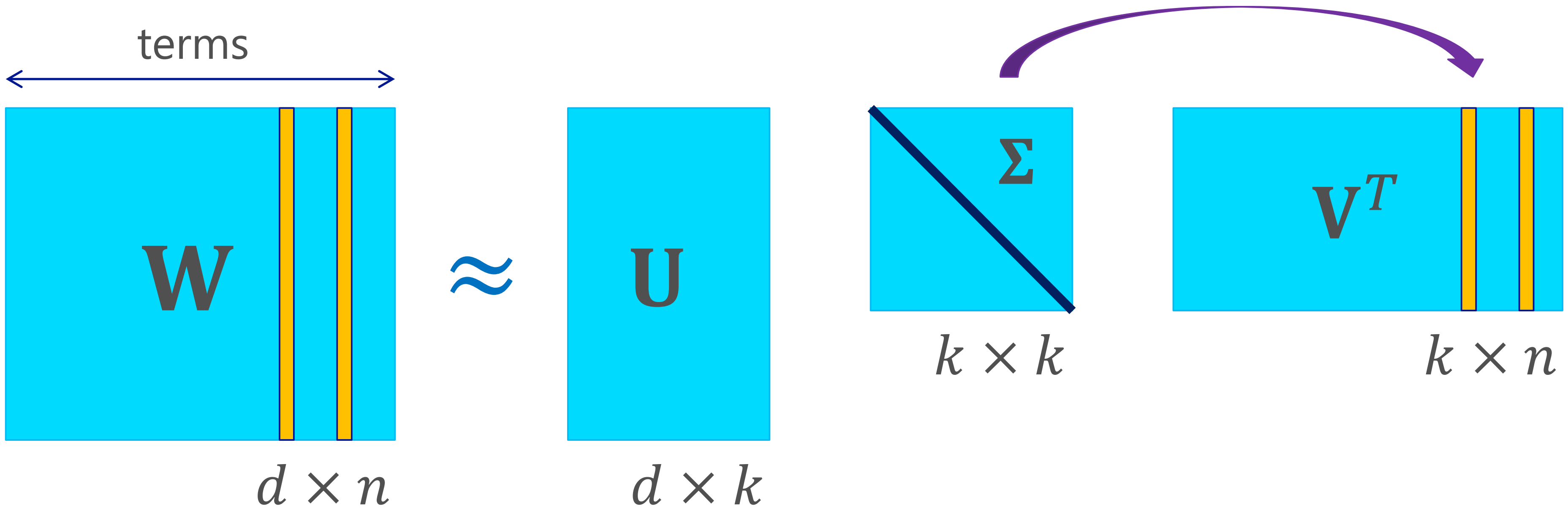
Target word: row-vector

Term: column-vector

	joy	gladden	sorrow	sadden	goodwill
Group 1: “joyfulness”	1	1	0	0	0
Group 2: “sad”	0	0	1	1	0
Group 3: “affection”	0	0	0	0	1

Cosine Score

Mapping to Latent Space via SVD



- SVD generalizes the original data
- Uncovers relationships not explicit in the thesaurus
- Term vectors projected to $\boldsymbol{k}$-dim latent space
- Word similarity: cosine of two column vectors in $\mathbf{\Sigma V}^T$

Problem: Handling Two Opposite Relations
Synonyms & Antonyms

- LSA cannot distinguish antonyms [Landauer 2002]
  - "Distinguishing synonyms and antonyms is still perceived as a difficult open problem."
    [Poon & Domingos 09]
- Idea: Change the data representation

Polarity Inducing LSA [Yih, Zweig & Platt 2012]

- Data representation
  - Encode two opposite relations in a matrix using "polarity"
    - Synonyms & antonyms (e.g., from a thesaurus)
- Factorization
  - Apply SVD to the matrix to find latent components
- Measuring degree of relation
  - Cosine of latent vectors

Encode Synonyms & Antonyms in Matrix

- Joyfulness: joy, gladden
- Sad: sorrow, sadden

Target word: row-vector

	joy	gladden	sorrow	sadden	goodwill
Group 1: “joyfulness”	1	1	0	0	0
Group 2: “sad”	0	0	1	1	0
Group 3: “affection”	0	0	0	0	1

Encode Synonyms & Antonyms in Matrix

- Joyfulness: joy, gladden
- Sad: sorrow, sadden

Inducing polarity

Target word: row-vector

	joy	gladden	sorrow	sadden	goodwill
Group 1: “joyfulness”	1	1	-1	-1	0
Group 2: “sad”	-1	-1	1	1	0
Group 3: “affection”	0	0	0	0	1

Encode Synonyms & Antonyms in Matrix

- Joyfulness: joy, gladden
- Sad: sorrow, sadden

Target word: row-vector

Inducing polarity

	joy	gladden	sorrow	sadden	goodwill
Group 1: "joyfulness"	1	1	-1	-1	0
Group 2: "sad"	-1	-1	1	1	0
Group 3: "affection"	0	0	0	0	1

Cosine Score: $+\,Synonyms$

Encode Synonyms & Antonyms in Matrix

- Joyfulness: joy, gladden
- Sad: sorrow, sadden

Inducing polarity

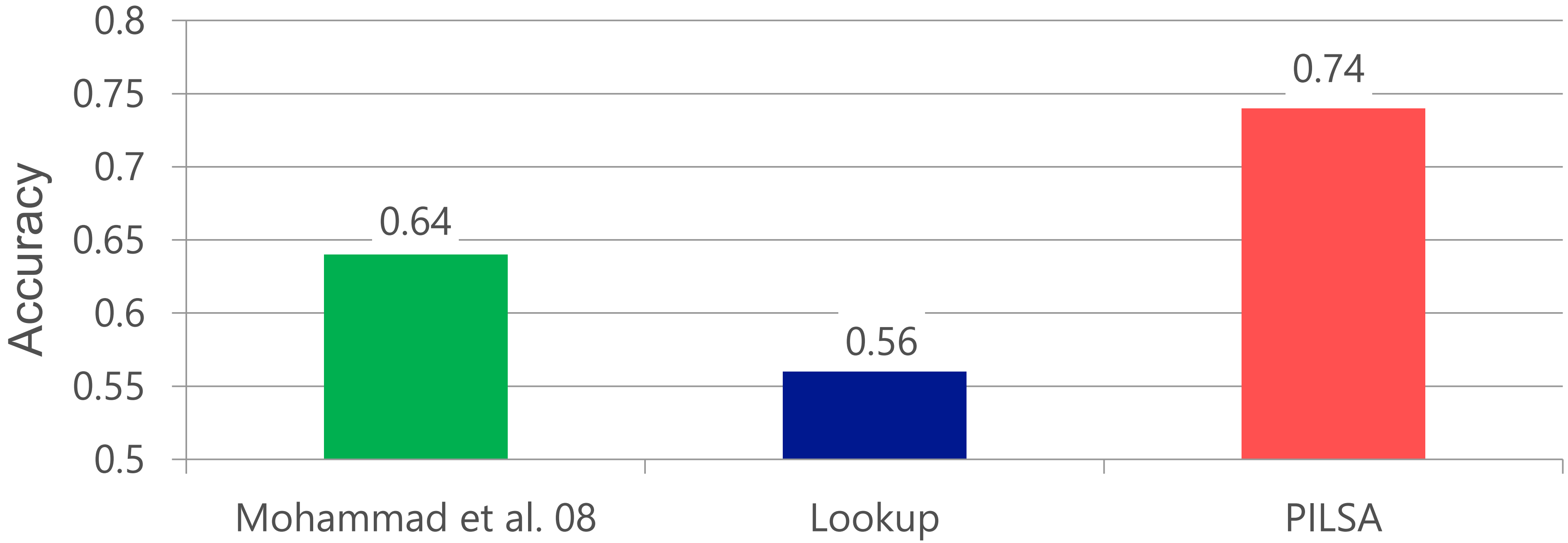
Target word: row-vector

	joy	gladden	sorrow	sadden	goodwill
Group 1: “joyfulness”	1	1	-1	-1	0
Group 2: “sad”	-1	-1	1	1	0
Group 3: “affection”	0	0	0	0	1

Cosine Score: $-\,Antonyms$

Results – GRE Antonym Test

- Task: GRE closest-opposite questions
  - Which is the closest opposite of *adulterate*?
    (a) renounce (b) forbid (c) purify (d) criticize (e) correct



Problem: How to Handle More Relations?

- Limitation of the matrix representation
  - Each entry captures a particular type of relation between two entities, or
  - Two opposite relations with the polarity trick
- Encoding other binary relations
  - Is-A (hyponym) – ostrich *is a* bird
  - Part-whole – engine is a *part of* car

Problem: How to Handle More Relations?

- Limitation of the matrix representation
  - Each entry captures a particular type of relation between two entities, or
  - Two opposite relations with the polarity trick
- Encoding other binary relations
  - Is-A (hyponym) – ostrich *is a* bird
  - Part-whole – engine is a *part of* car
- Idea: Encode multiple relations in a 3-way tensor (3-dim array)!

Multi-Relational LSA [Chang, Yih & Meek 2013]

- Data representation
  - Encode multiple relations in a tensor
    - Synonyms, antonyms, hyponyms (is-a), …
      (e.g., from a linguistic knowledge base)
- Factorization
  - Apply tensor decomposition to the tensor to find latent components
- Measuring degree of relation
  - Cosine of latent vectors after projection

Encode Multiple Relations in Tensor

- Represent word relations using a tensor
  - Each slice encodes a relation between terms and target words.

	joy	gladden	sadden	feeling
joyfulness	1	1	0	0
gladden	1	1	0	0
sad	0	0	1	0
anger	0	0	0	0

Synonym layer

	joy	gladden	sadden	feeling
joyfulness	0	0	0	0
gladden	0	0	1	0
sad	1	0	0	0
anger	0	0	0	0

Antonym layer

Construct a tensor with two slices

Encode Multiple Relations in Tensor

- Can encode multiple relations in the tensor

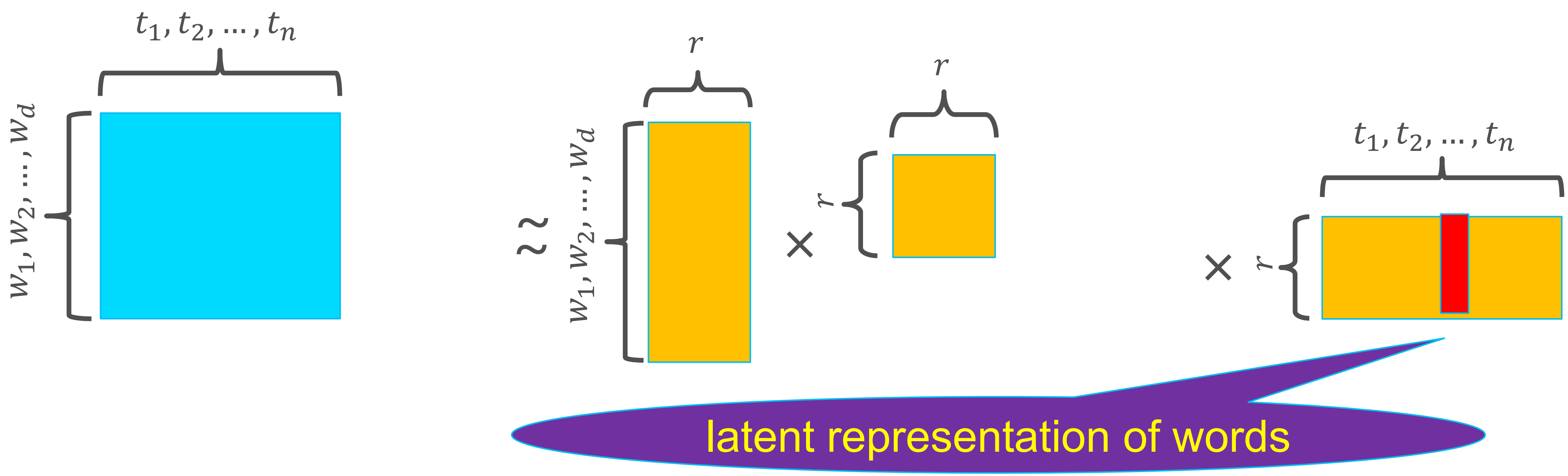
1	1	0	0
1	1	0	0
0	0	1	0
0	0	0	0

	joy	gladden	sadden	feeling
joyfulness	0	0	0	1
gladden	0	0	0	0
sad	0	0	0	1
anger	0	0	0	1

Hyponym layer

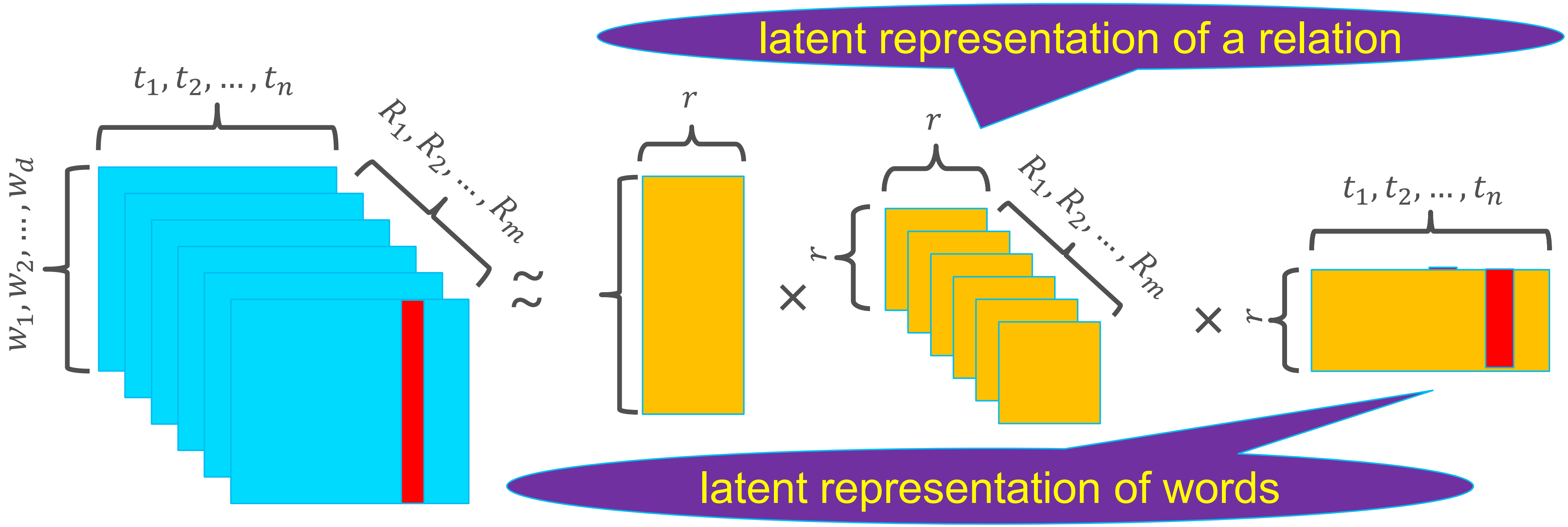
Tensor Decomposition – Analogy to SVD

- Derive a low-rank approximation to generalize the data and to discover unseen relations
- Apply Tucker decomposition and reformulate the results



Tensor Decomposition – Analogy to SVD

- Derive a low-rank approximation to generalize the data and to discover unseen relations
- Apply Tucker decomposition and reformulate the results



Experiment: Data for Building MRLSA Model

- Encarta Thesaurus
  - Record synonyms and antonyms of target words
  - Vocabulary of 50k terms and 47k target words
- WordNet
  - Has synonym, antonym, hyponym, hypernym relations
  - Vocabulary of 149k terms and 117k target words
- Goals:
  - MRLSA generalizes LSA to model multiple relations
  - Improve performance by combing heterogeneous data

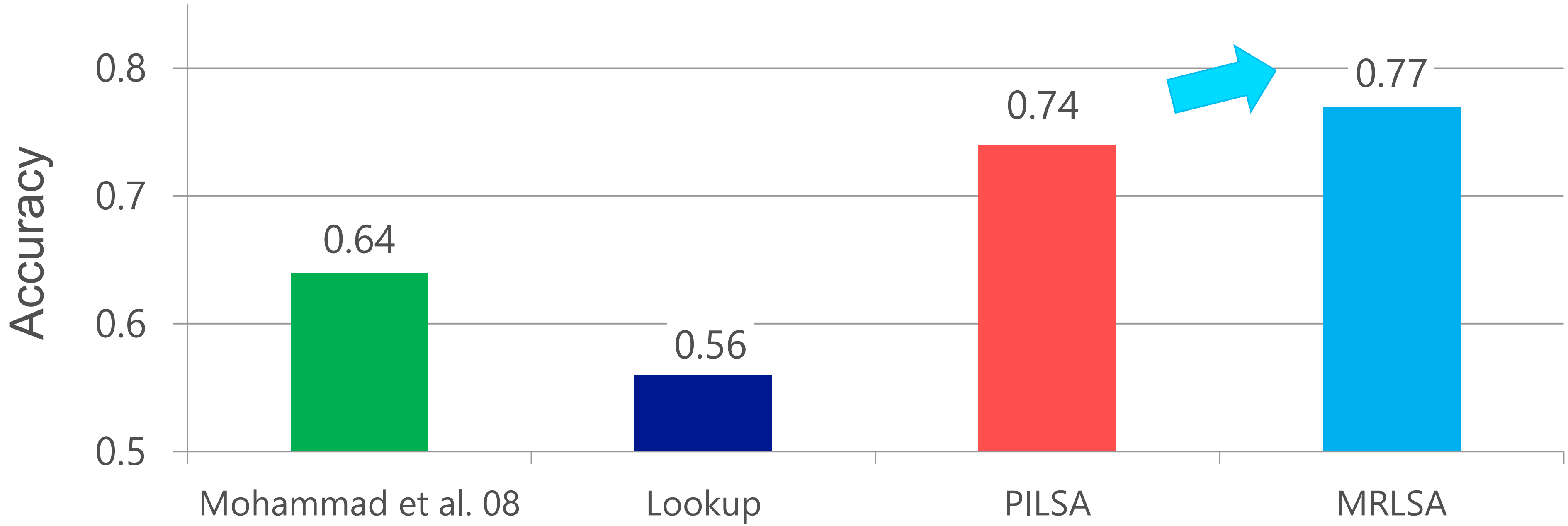
Example Antonyms Output by MRLSA

Target	High Score Words
inanimate	alive, living, bodily, in-the-flesh, incarnate
alleviate	exacerbate, make-worse, in-flame, amplify, stir-up
relish	detest, abhor, abominate, despise, loathe

* *Words in blue are antonyms listed in the Encarta thesaurus.*

Results – GRE Antonym Test

- Task: GRE closest-opposite questions
  - **Which is the closest opposite of *adulterate*?**
    (a) renounce (b) forbid (c) purify (d) criticize (e) correct

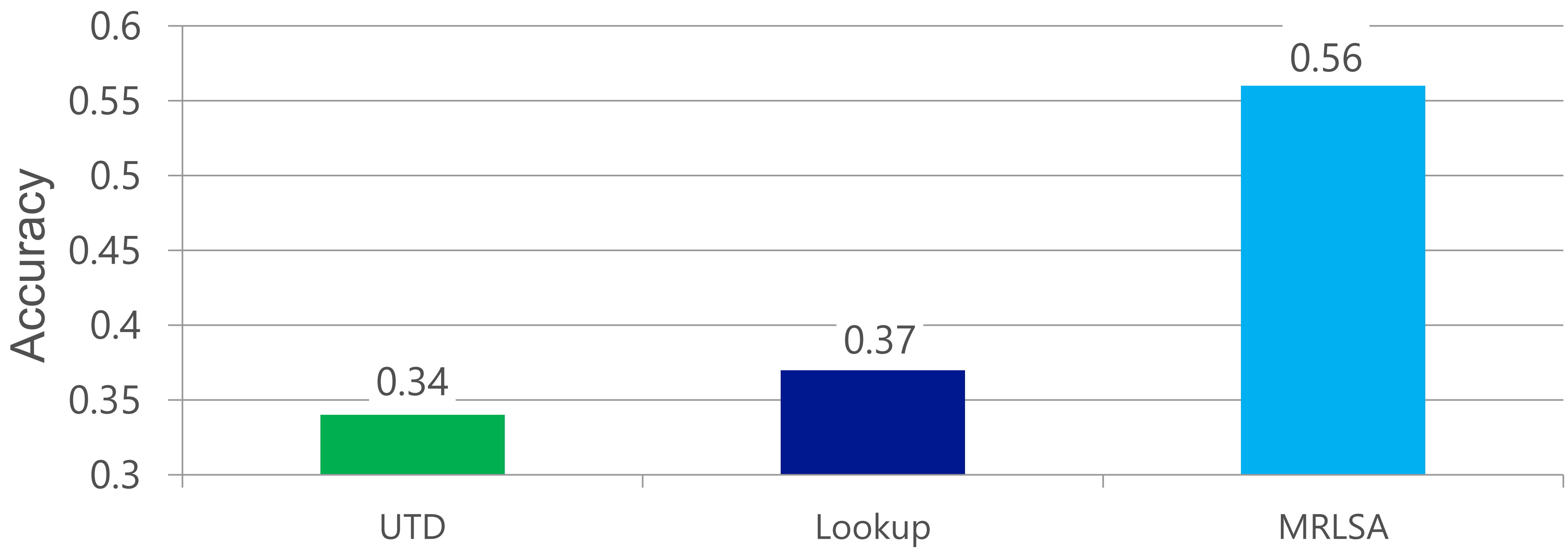


Example Hyponyms Output by MRLSA

Target	High Score Words
bird	ostrich, gamecock, nighthawk, amazon, parrot
automobile	minivan, wagon, taxi, minicab, gypsy cab
vegetable	buttercrunch, yellow turnip, romaine, chipotle, chilli

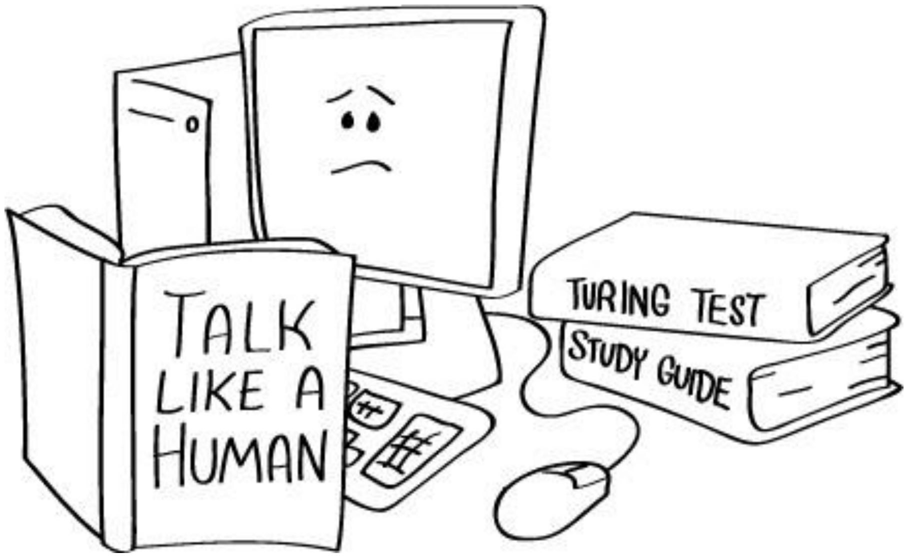
Results – Relational Similarity (SemEval-2012)

- Task: Class-Inclusion Relation (*Y* *is-a* kind of *X*)
  - Most/least illustrative word pairs
    (a) art:abstract (b) song:opera (c) footwear:boot (d) hair:brown



Natural Language Understanding

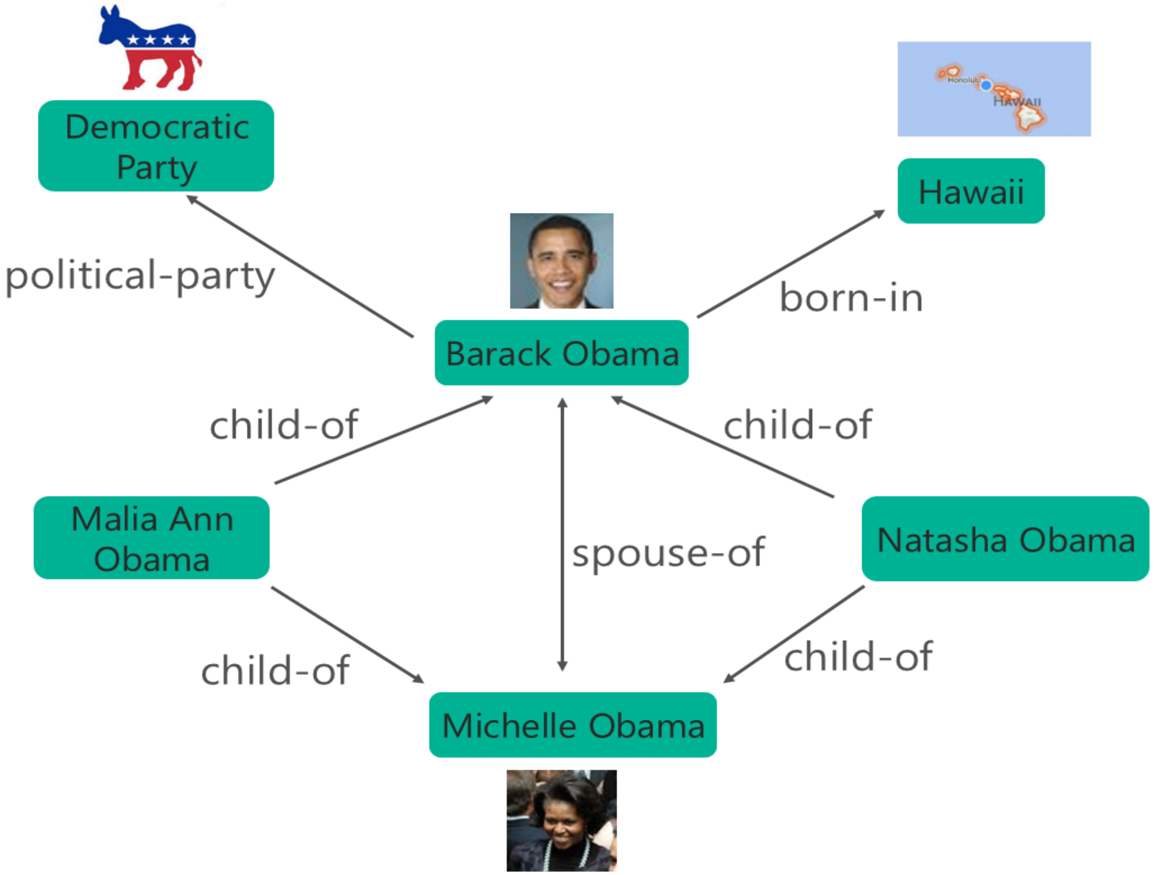
- Continuous Word Representations & Lexical Semantics
- Knowledge Base Embedding
- Semantic Parsing & Question Answering

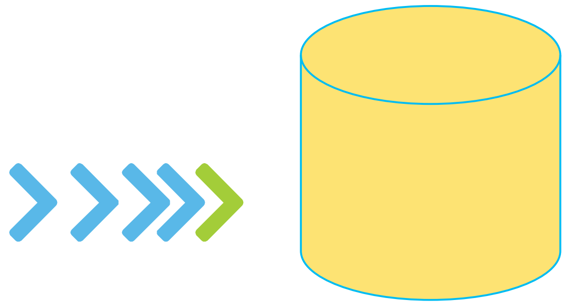


http://csunplugged.org/turing-test

Knowledge Base

- Captures world knowledge by storing properties of millions of entities, as well as relations among them





Freebase
DBpedia
YAGO
NELL
OpenIE/ReVerb

KB Applications in NLP & IR

- Question Answering

  "What are the names of Obama's daughters?"

- Information Extraction
  - *"Hathaway was born in Brooklyn, New York."*

- Web Search
  - Identify entities and relationships in queries

Reasoning with Knowledge Base

- Knowledge base is never complete!
  - Extract previously unknown facts from new corpora
  - Predict new facts via inference

- Modeling multi-relational data
  - Statistical relational learning [Getoor & Taskar, 2007]
  - Path ranking methods (e.g., random walk) [e.g., Lao+ 2011]
  - Knowledge base embedding
    - Very efficient
    - Better prediction accuracy

Knowledge Base Embedding

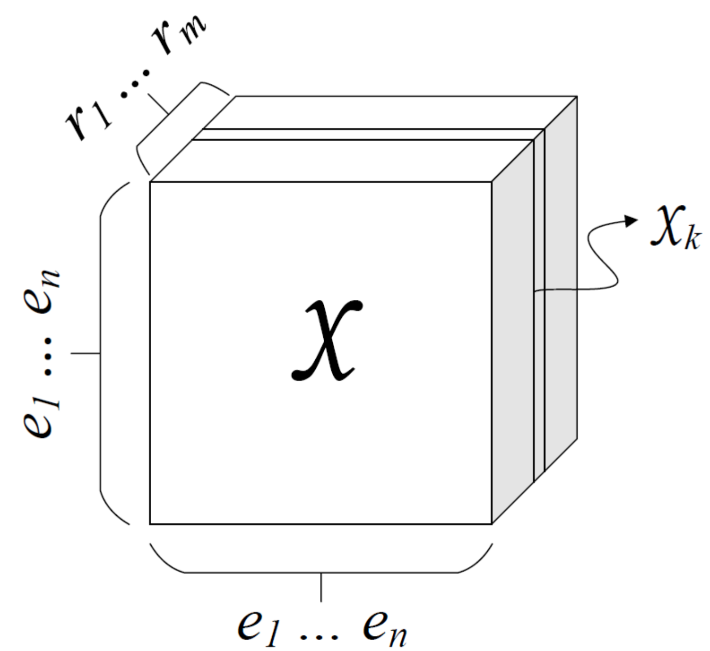
- Each entity in a KB is represented by an $\boldsymbol{R^d}$ vector
- Predict whether $(\boldsymbol{e_1}, \boldsymbol{r}, \boldsymbol{e_2})$ is true by $\boldsymbol{f_r}(\boldsymbol{v_{e_1}}, \boldsymbol{v_{e_2}})$

- Recent work on KB embedding
  - Tensor decomposition
    - RESCAL [Nickel+, ICML-11], TRESCAL [Chang+, EMNLP-14]
  - Neural networks
    - SME [Bordes+, AISTATS-12], NTN [Socher+, NIPS-13], TransE [Bordes+, NIPS-13]

Knowledge Base Representation (1/2)

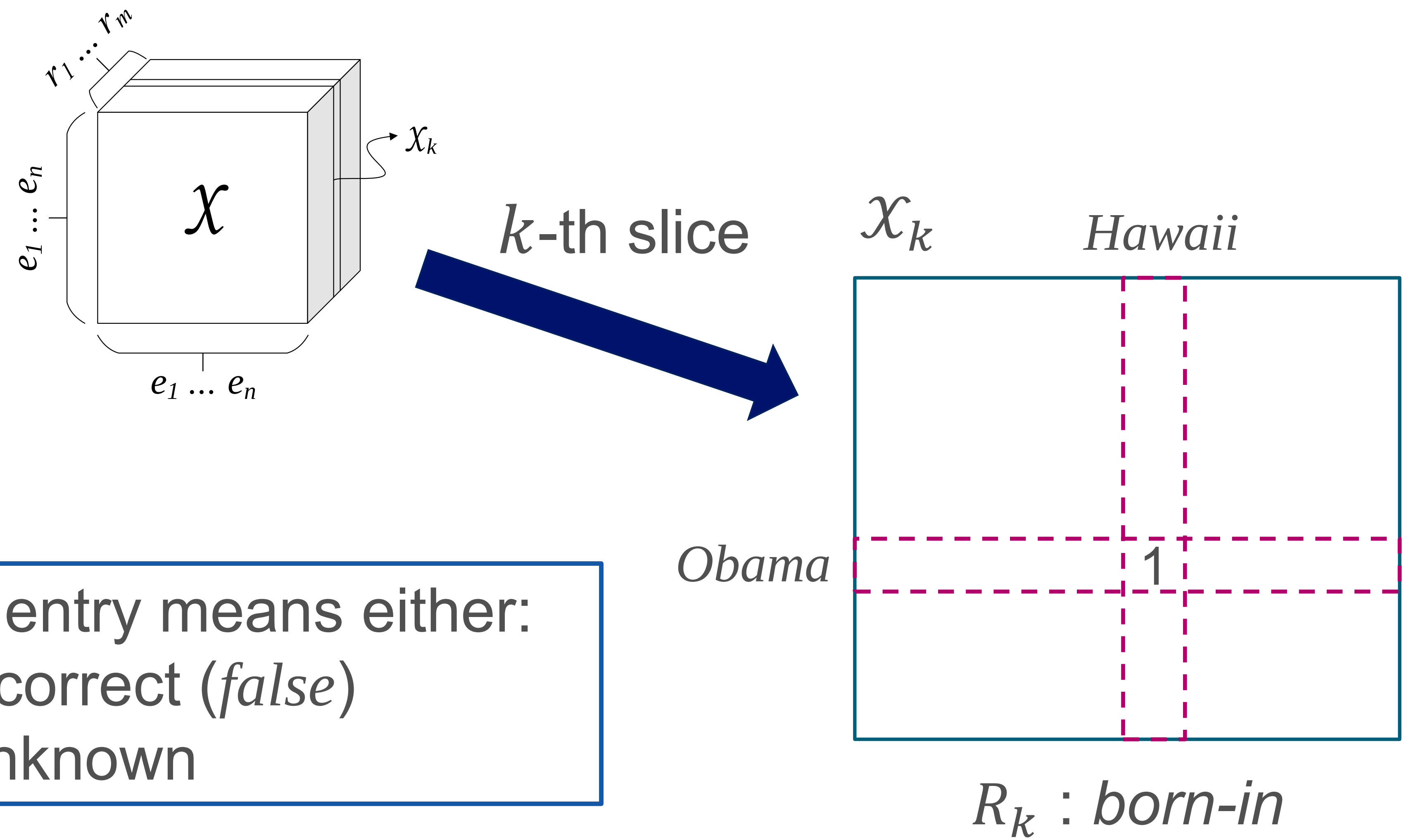
- Collection of **subj-pred-obj** triples – (e_1, r, e_2)

Subject	Predicate	Object
Obama	Born-in	Hawaii
Bill Gates	Nationality	USA
Bill Clinton	Spouse-of	Hillary Clinton
Satya Nadella	Work-at	Microsoft
…	…	…



n: # entities, m: # relations

Knowledge Base Representation (2/2)

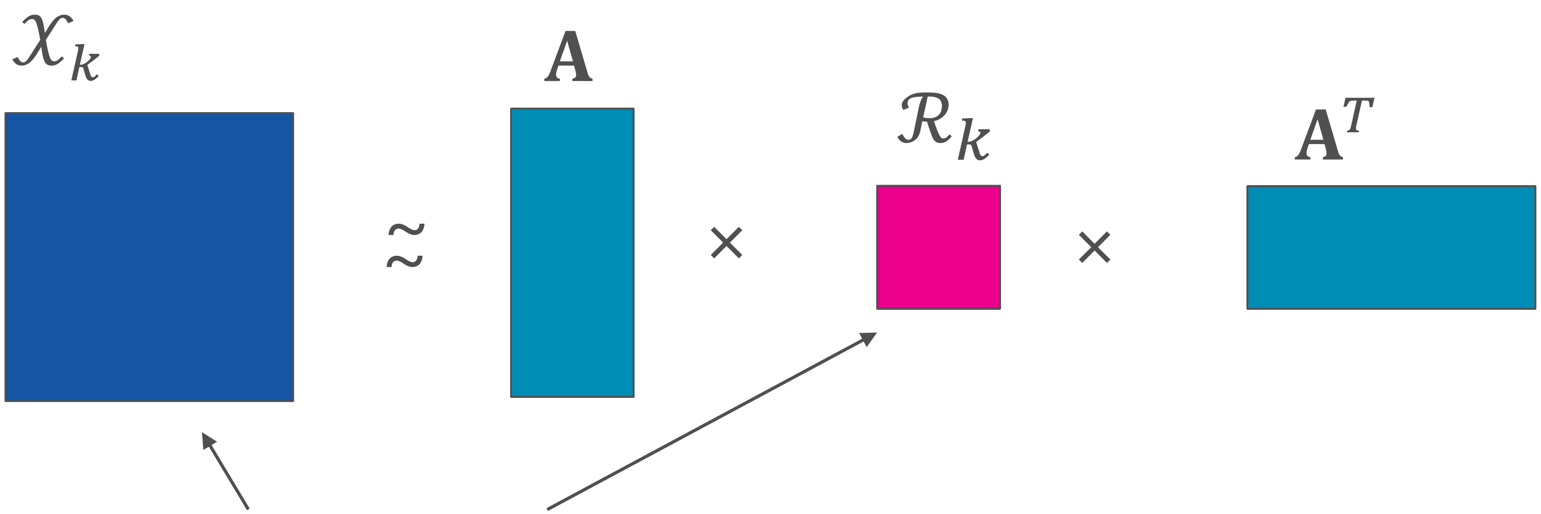


A zero entry means either:

- Incorrect (*false*)
- Unknown

Tensor Decomposition Objective

- Objective: $\underbrace{\frac{1}{2}\left(\sum_k \|\mathcal{X}_k - \mathbf{A}\mathcal{R}_k\mathbf{A}^T\|_F^2\right)}_{\textit{Reconstruction Error}} + \underbrace{\frac{1}{2}\left(\|A\|_F^2 + \sum_k \|\mathcal{R}_k\|_F^2\right)}_{\textit{Regularization}}$



k-th relation

RESCAL [Nickel+, ICML-11]

Measure the Degree of a Relationship

$$f_{\text{born-in}}(\text{Obama}, \text{Hawaii})$$

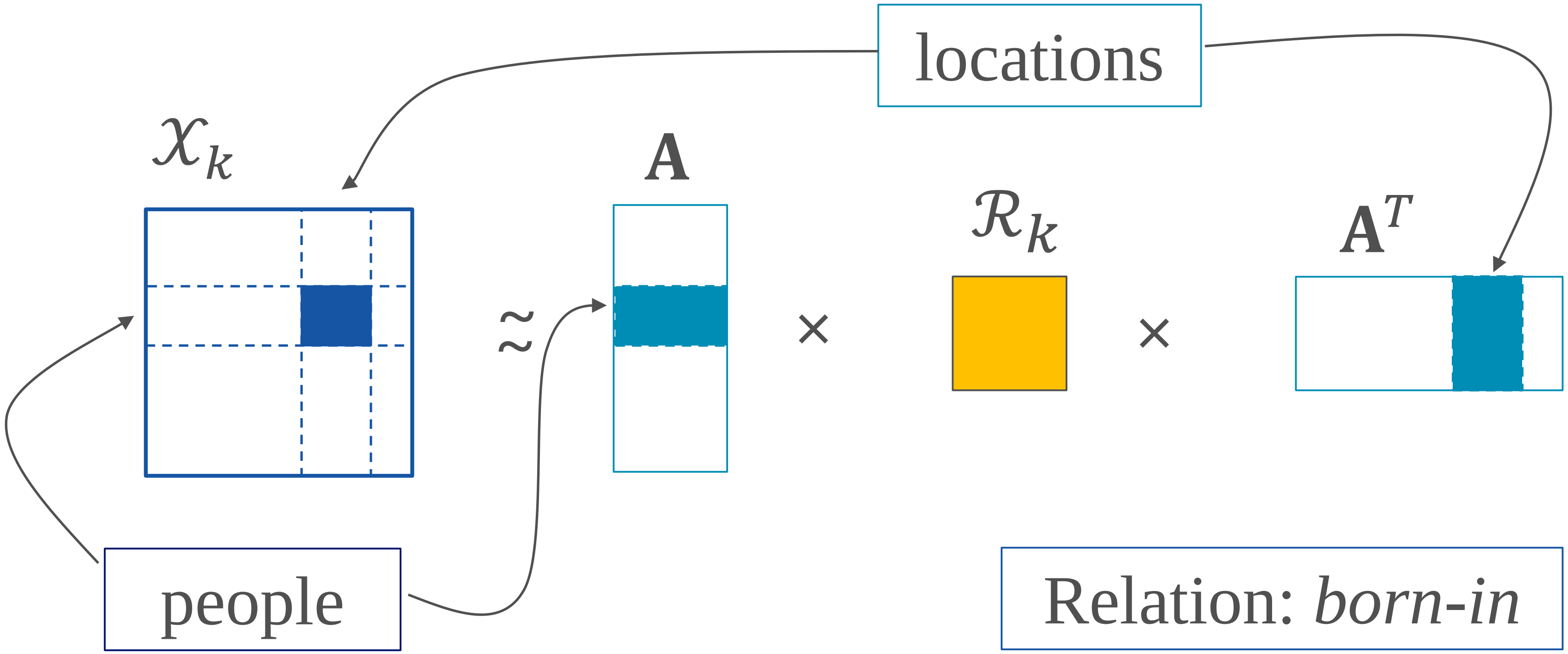
$$= \mathbf{A}_{\text{Obama},:}\, \mathcal{R}_{\text{born-in}}\, \mathbf{A}^{\mathrm{T}}_{\text{Hawaii},:}$$

Typed Tensor Decomposition – TRESCAL [Chang+ EMNLP-14]

- Relational domain knowledge
  - Type information and constraints
  - Only legitimate entities are included in the loss
- Benefits of leveraging type information
  - Faster model training time
  - Highly scalable to large KB
  - Higher prediction accuracy

Typed Tensor Decomposition Objective

- Reconstruction error: $\frac{1}{2}\sum_k \|\mathcal{X}_k - \mathbf{A}\mathcal{R}_k\mathbf{A}^T\|_F^2$



Typed Tensor Decomposition Objective

- Reconstruction error: $\frac{1}{2}\sum_{k} \| \mathcal{X}'_k - \mathbf{A}_{k_l} \mathcal{R}_k \mathbf{A}^T_{k_r} \|_F^2$

$$\mathcal{X}'_k \approx \mathbf{A}_{k_l} \times \mathcal{R}_k \times \mathbf{A}^T_{k_r}$$

Training Procedure – Alternating Least-Squares (ALS) Method

Fix $\mathcal{R}_k$, update $\mathbf{A}$

Fix $\mathbf{A}$, update $\mathcal{R}_k$

Training Procedure – Alternating Least-Squares (ALS) Method

$$\mathbf{A} \leftarrow \left[\sum_k \mathcal{X}_k' \mathbf{A}_{k_r} \mathcal{R}_k^{\mathrm{T}} + \mathcal{X}_k'^{\mathrm{T}} \mathbf{A}_{k_l} \mathcal{R}_k\right] \left[\sum_k B_{k_r} + C_{k_l} + \lambda \mathbf{I}\right]^{-1}$$

where $B_{k_r} = \mathcal{R}_k \mathbf{A}_{k_r}^{\mathrm{T}} \mathbf{A}_{k_r} \mathcal{R}_k^{\mathrm{T}}$, $C_{k_l} = \mathcal{R}_k^{\mathrm{T}} \mathbf{A}_{k_l}^{\mathrm{T}} \mathbf{A}_{k_l} \mathcal{R}_k$.

$$\mathbf{vec}(\mathcal{R}_k) \leftarrow \left(\mathbf{A}_{k_r}^{\mathrm{T}} \mathbf{A}_{k_r} \otimes \mathbf{A}_{k_l}^{\mathrm{T}} \mathbf{A}_{k_l} + \lambda \mathbf{I}\right)^{-1} \times \mathbf{vec}\left(\mathbf{A}_{k_l}^{\mathrm{T}} \mathcal{X}_k' \mathbf{A}_{k_r}\right)$$

Complexity Analysis

- Without Type information (RESCAL): $O(nr^2 + pr)$
  - n: # entities
  - p: # non-zero entries
  - r: # dimensions of projected entity vectors

- With Type information (TRESCAL): $O(\bar{n}r^2 + pr)$
  - $\bar{n}$: average # entities satisfying the type constraint
  - $\bar{n}/n \cong 0.06$

Experiments – KB Completion

- KB – Never Ending Language Learning (NELL)
  - Training: version 165
  - Developing: new facts between v.166 and v.533
  - Testing: new facts between v.534 and v.745

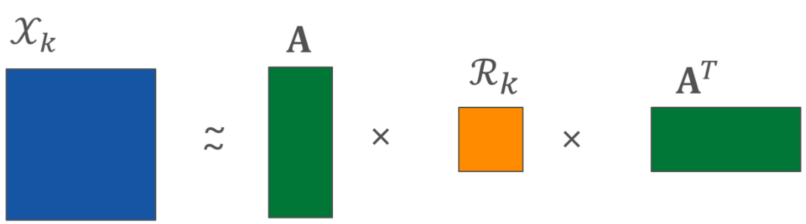
- Data statistics of the training set

# Entities	753k
# Relation Types	229
# Entity Types	300
# Entity-Relation Triples	1.8M

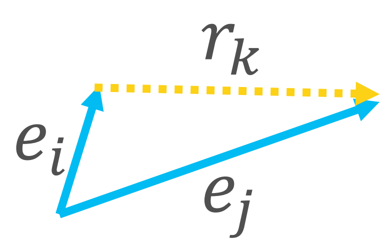
Tasks & Baselines

- Entity Retrieval: $(e_i, r_k, ?)$
  - One positive entity with 100 negative entities
- Relation Retrieval: $(e_i, ?, e_j)$
  - Positive entity pairs with equal number of negative pairs

- Baselines:

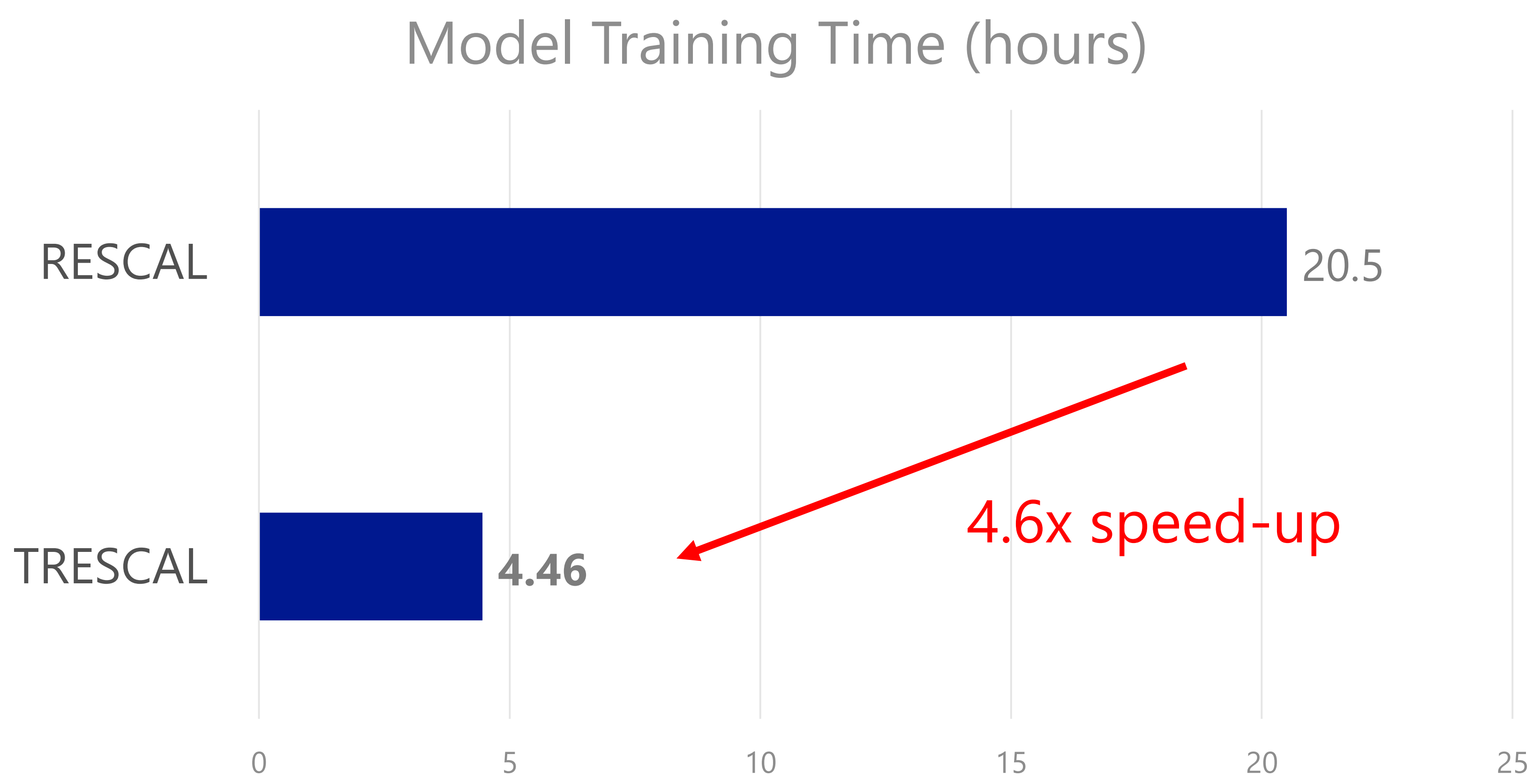


RESCAL
[Nickel+, ICML-11]



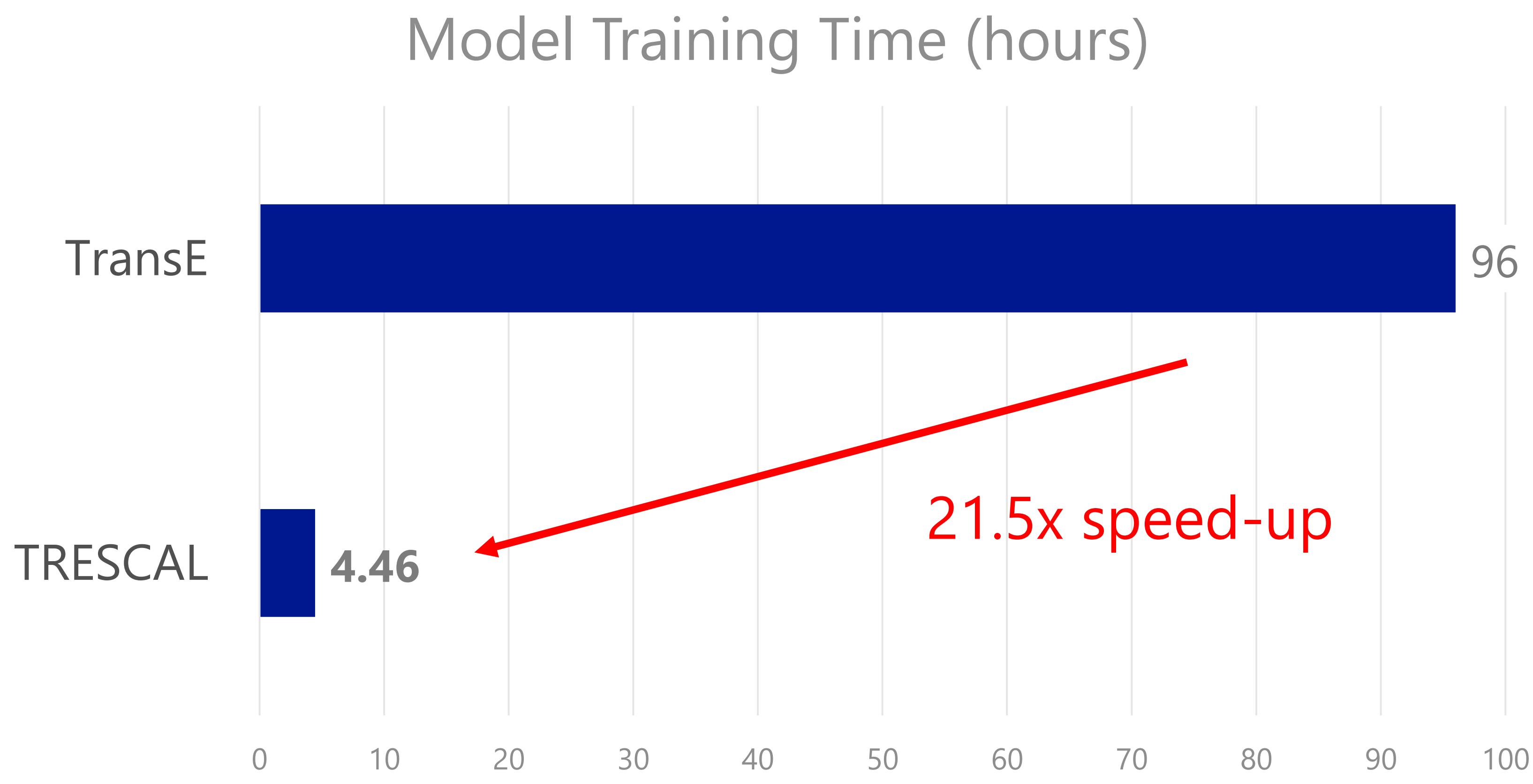
TransE
[Bordes+, NIPS-13]

Training Time Reduction



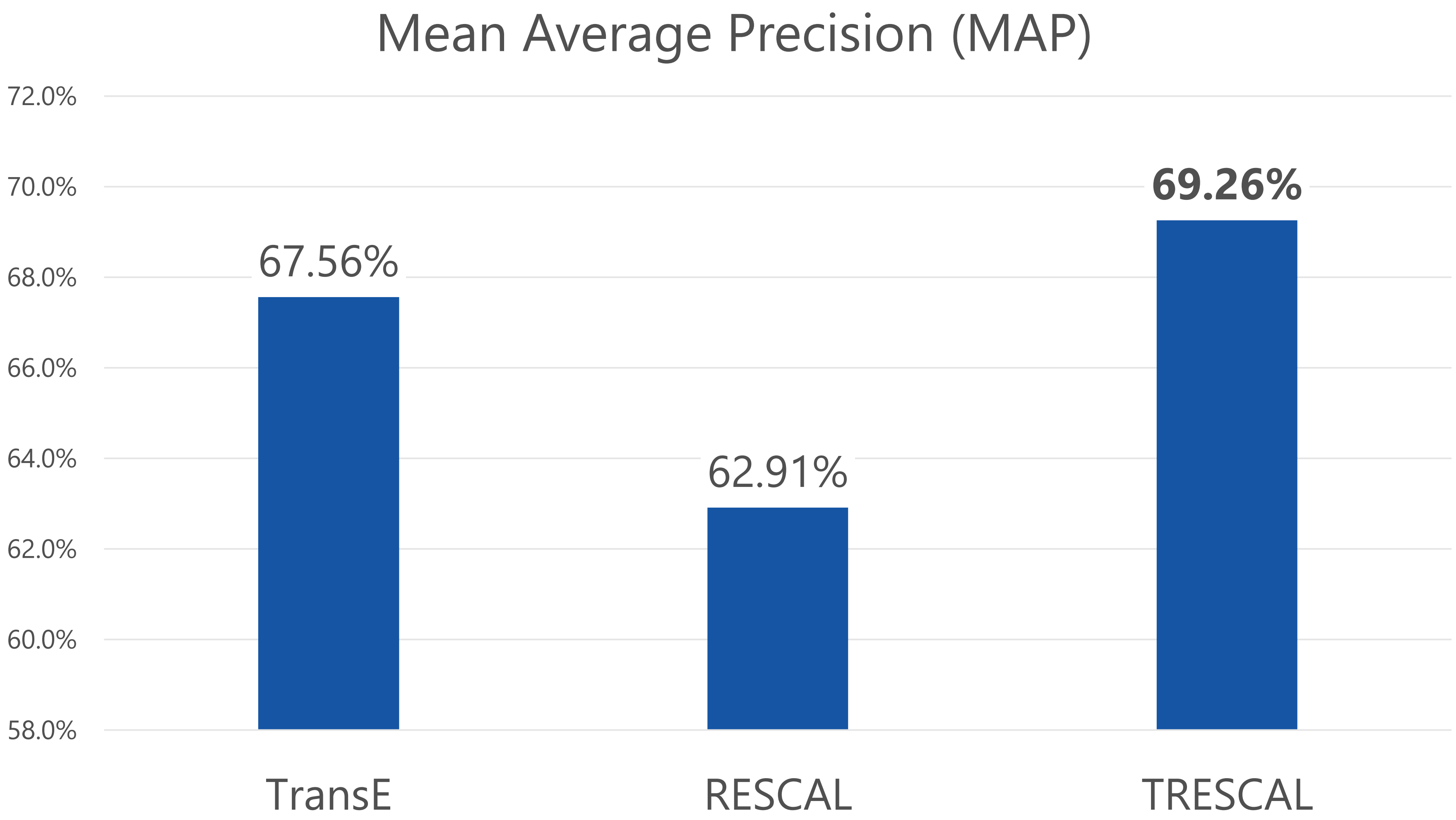
- Both models finish training in 10 iterations.
- TRESCAL filters 96% entity triples with incompatible types.

Training Time Reduction

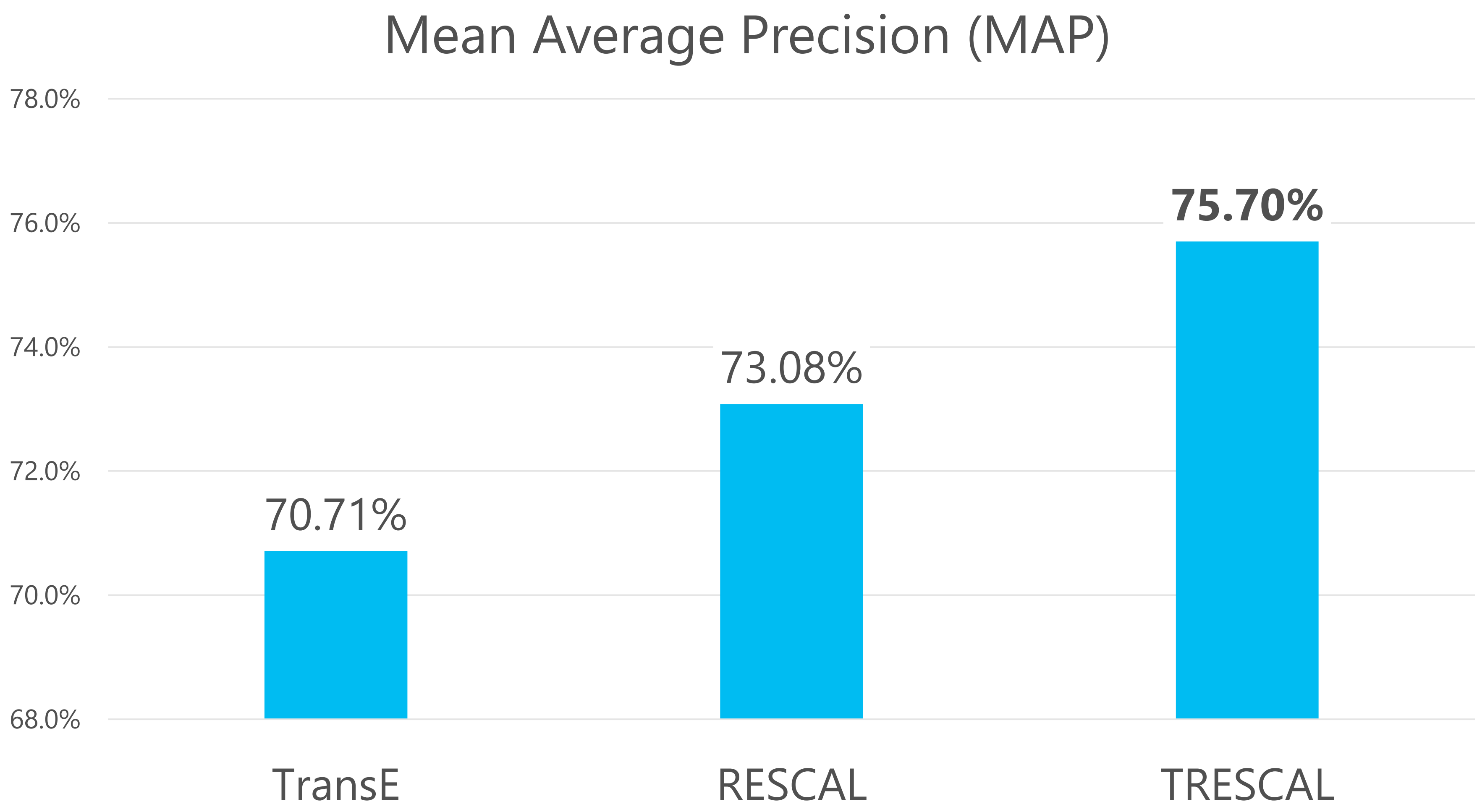


- # iterations for TransE is set to 500 (the default value).

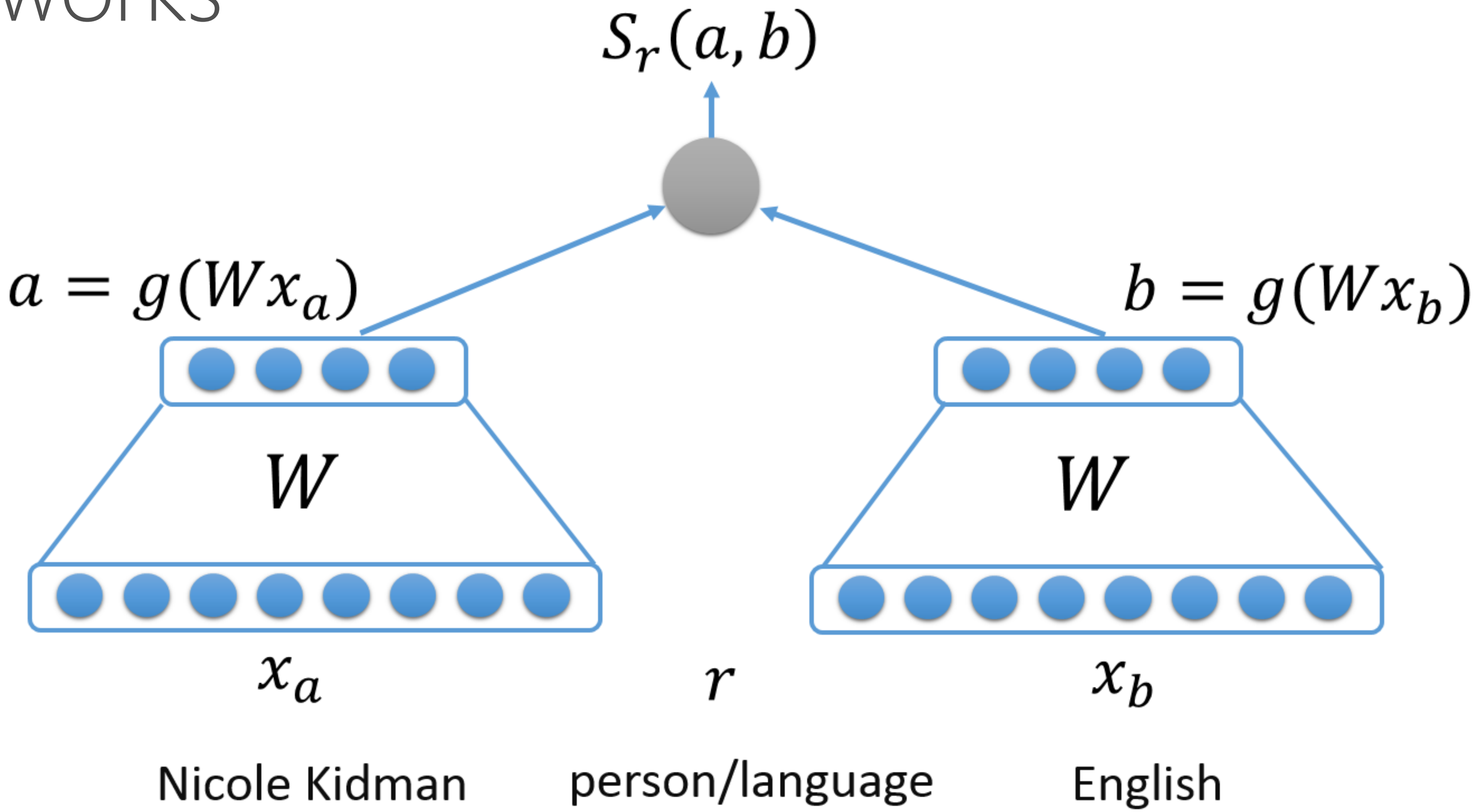
Entity Retrieval $(e_i, r_k, ?)$



Relation Retrieval $(e_i, ?, e_j)$



Embedding Relationships using Neural Networks



Relation Operators

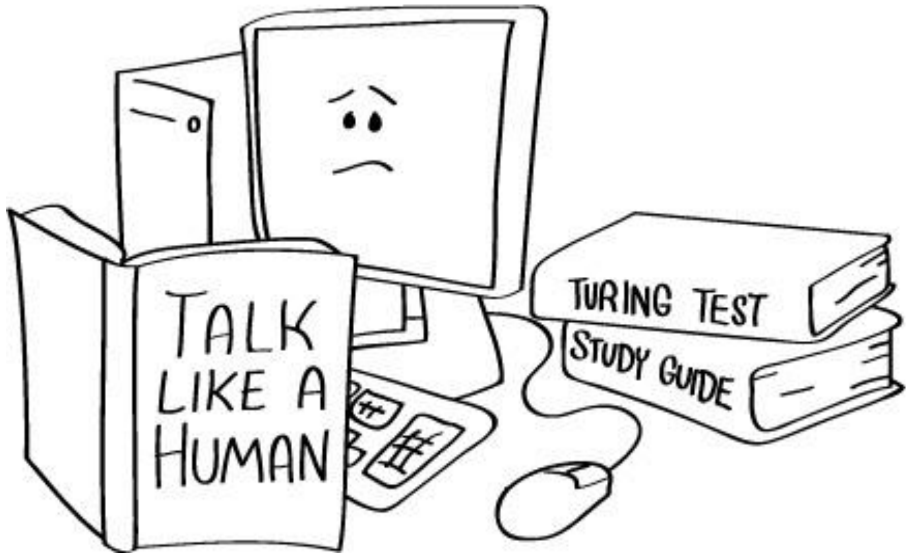
Relation representation	Scoring Function $S_r(a, b)$	# Parameters
Vector (TransE) (Bordes+ 2013)	$\|\|a - b + V_r\|\|_{1,2}$	$O(n_r \times k)$
Matrix (Bilinear) (Bordes+ 2012, Collobert & Weston 2008)	$a^T M_r b$ $u^T f(M_{r1} a + M_{r2} b)$	$O(n_r \times k^2)$
Tensor (NTN) (Socher+ 2013)	$u^T f(a^T T_r b + M_{r1} a + M_{r2} b)$	$O(n_r \times k^2 \times d)$
Diagonal Matrix (RelDot) (Yang+ 2014)	$a^T diag(M_r) b$	$O(n_r \times k)$

Empirical Comparisons of NN-based KB Embedding Methods [Yang+ NIPS-LS-2014]

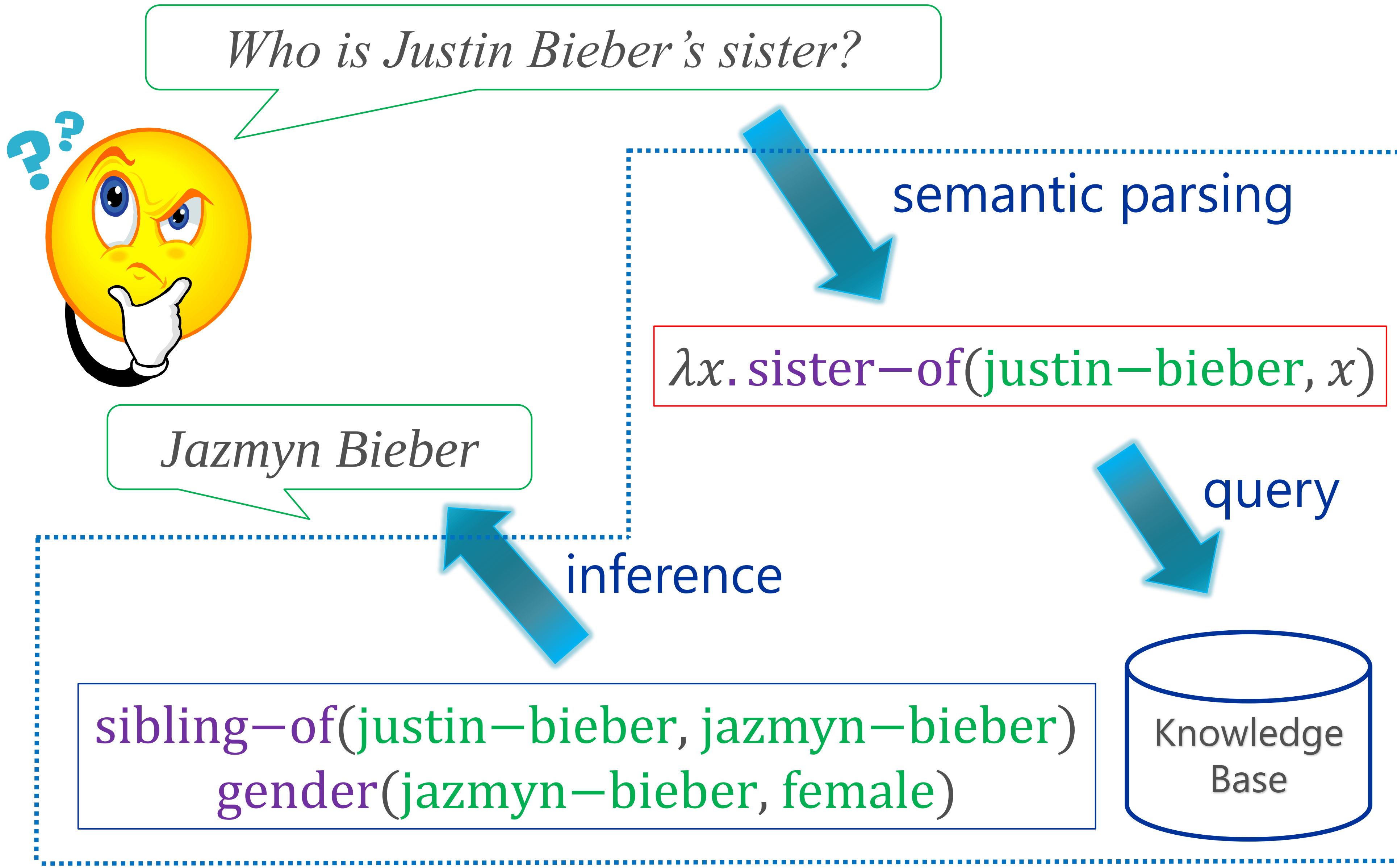
- Models with fewer parameters tend to perform better.
- The bilinear operator ($\boldsymbol{a}^T \boldsymbol{M}_r \boldsymbol{b}$) plays an important role in capturing entity interact.
- With the same model complexity, multiplicative operations are superior to additive operations in modeling relations.
- Initializing entity vectors with pre-trained phrase vectors can significantly boost performance.

Natural Language Understanding

- Continuous Word Representations & Lexical Semantics
- Knowledge Base Embedding
- Semantic Parsing & Question Answering



http://csunplugged.org/turing-test


Who is Justin Bieber's sister?
semantic parsing
λx. sister−of(justin−bieber, x)
query
Knowledge Base
inference
sibling−of(justin−bieber, jazmyn−bieber)
gender(jazmyn−bieber, female)
Jazmyn Bieber

Key Challenge – Language Mismatch

- Lots of ways to ask the same question
  - *"What was the date that Minnesota became a state?"*
  - *"Minnesota became a state on?"*
  - *"When was the state Minnesota created?"*
  - *"Minnesota's date it entered the union?"*
  - *"When was Minnesota established as a state?"*
  - *"What day did Minnesota officially become a state?"*

- Need to map them to the predicate defined in KB
  - location.dated_location.date_founded

Recent Work

- Most approaches rely on lexical matching
  - Paraphrase [Berant&Liang, ACL-2014]
    question → canonical question → Logical form
  - CCG as intermediate representation [Reddy+, TACL 2014]

- Continuous-space methods
  - Subgraph embeddings [Bordes+, EMNLP-2014]
  - Compositional entity/relation matching [Yih+, ACL-2014]

Single-Relation Semantic Parsing [Yih+, ACL-14]

- Most common questions in the search query logs
  - *"How old is Kirk Douglas, the actor?"*
  - *"What county is St. Elizabeth MO in?"*
  - *"What year was the 8 track invented?"*
  - *"Who owns the Texas Rangers?"*

- Foundation for answering complicated questions
  - *"Name a director of movies starred by Tom Hanks."*

Key Ideas & Related Work

- Simple Context-Free Grammar
  - Separate a question into a relation pattern and an entity mention
  - Match pattern/mention and KB relation/entity using convolutional neural networks

- Inspired by Paralex [Fader et al. 2013]
  - 35M question paraphrase pairs from WikiAnswers
  - Learn weighted lexical matching rules

Task & Problem Definition

Input

- A KB as a collection of triples (r, e_1, e_2)
- A single-relation question, describing a relation and one of its entity arguments

"When were DVD players invented?"

Output

- An entity that has the relation with the given entity

High-level Approach: Semantic Parsing

$$Q = \text{“}When\ were\ DVD\ players\ invented?\text{”}$$

$$Q \rightarrow P \wedge M$$
$$P \rightarrow when\ were\ \mathrm{X}\ invented$$
$$M \rightarrow DVD\ players$$
$$when\ were\ X\ invented \rightarrow \mathrm{be{-}invent{-}in}_2$$
$$DVD\ players \rightarrow \mathrm{dvd{-}player}$$

$$\lambda x.\,\mathrm{be{-}invent{-}in}(\mathrm{dvd{-}player}, x)$$

Procedure: Enumerate All Hypotheses

Q = "*When were DVD players invented*?"

$P \rightarrow when\ \mathrm{X}\ players\ invented$
$M \rightarrow were\ DVD$

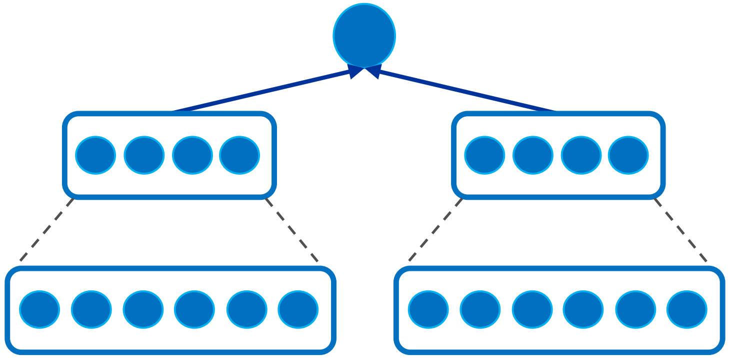
Procedure: Enumerate All Hypotheses

$Q =$ “*When were DVD players invented?*”

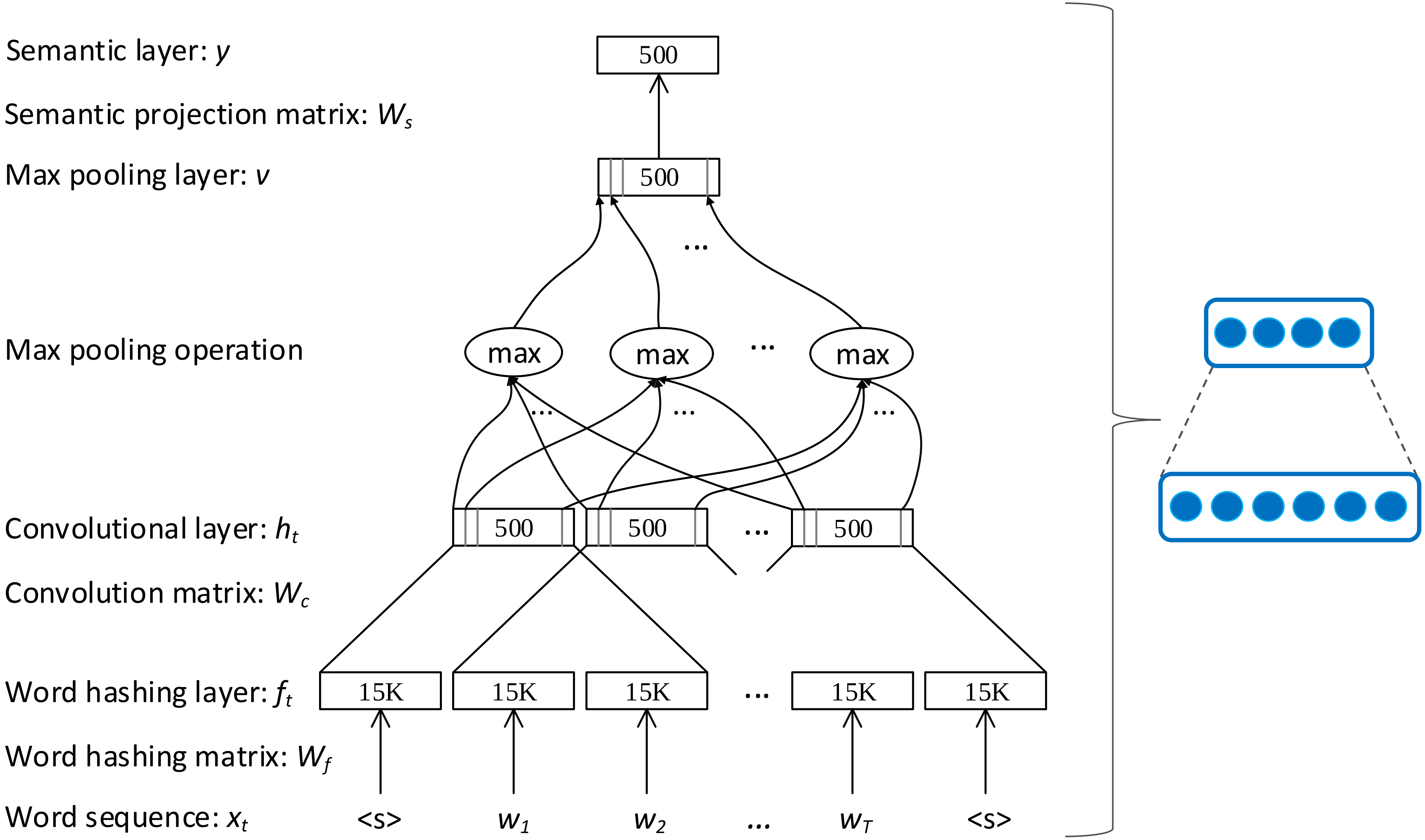
$P \rightarrow when\ were\ \mathrm{X}\ invented$
$M \rightarrow DVD\ players$

$Prob(\text{be}-\text{invent}-\text{in}_2 | when\ were\ \mathrm{X}\ invented) = 0.5$
$Prob(\text{dvd}-\text{player} | DVD\ players) = 0.7$
$Prob(\lambda x.\ \text{be}-\text{invent}-\text{in}(\text{dvd}-\text{player}, x) | Q) = 0.35$

Semantic Matching via
Deep Semantic Similarity Model !



Convolutional Deep Structured Semantic Model



Experiments: Data

Knowledge base: ReVerb [Fader et al., 2011]

Relation	Entity Argument #1	Entity Argument #2
be-official-language	chinese-and-english	hong-kong
be-second-largest-city-in	arequipa	peru
be-tallest-mountain-in	ararat	armenia
have-population-of	city-of-vancouver	587,891
provide	microsoft	office-software
use-for	laser	lasik
…	…	…

Experiments: Data

Paralex dataset [Fader et al., 2013]

- 1.8M (question, single-relation queries)

  When were DVD players invented?

  $\lambda x.\, \mathrm{be{-}invent{-}in}(\mathrm{dvd{-}player}, x)$

- 1.2M (relation pattern, relation)

  When were X invented?

  $\mathrm{be{-}invent{-}in}_2$

- 160k (mention, entity)

  Saint Patrick day

  $\mathrm{st{-}patrick{-}day}$

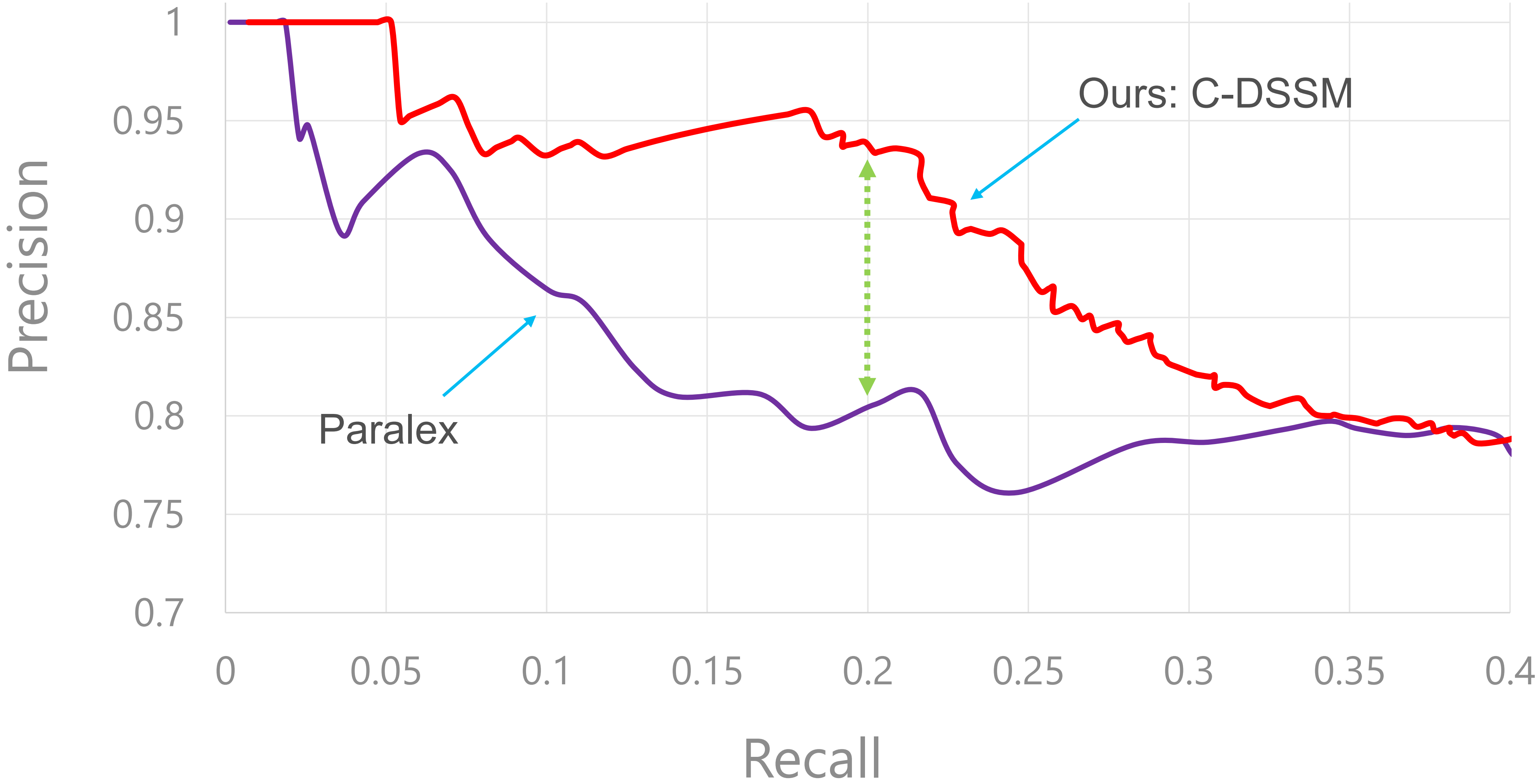
Experiments: Task – Question Answering

- Same test questions in the Paralex dataset
- 698 questions from 37 clusters

- *What language do people in Hong Kong use?*
  $\text{be-speak-in}(\text{english}, \text{hong-kong})$
  $\text{be-predominant-language-in}(\text{cantonese}, \text{hong-kong})$

- *Where do you find Mt Ararat?*
  $\text{be-highest-mountain-in}(\text{ararat}, \text{turkey})$
  $\text{be-mountain-in}(\text{ararat}, \text{armenia})$

Experiments: Results



Cherries

- What is the national anthem in the France?
  PARALEX: be-currency-in.r euro.e france.e
  CNNSM: be-national-anthem-of.r la-marseillaise.e france.e

- What is the title of france national anthem?
  PARALEX: be-national-dog-of.r poodles.e france.e
  CNNSM: be-national-anthem-of.r la-marseillaise.e france.e

- What is the name of the national anthem of France?
  PARALEX: be-national-language-in.r french.e france.e
  CNNSM: be-national-anthem-of.r la-marseillaise.e france.e

More Cherries

- What is the largest city in Peru?
  PARALEX: be-city-in.r cabana.e peru.e
  CNNSM: be-largest-city-in.r lima.e peru.e

- When was Apple Computer founded?
  PARALEX: be-founder-of.r steve-jobs.e apple.e
  CNNSM: be-found-on.r apple-computer.e april-1-,-1976.e

- What is the plural form of the word bacterium?
  PARALEX: be-plural-form-of.r virii.e virus.e
  CNNSM: be-plural-form-of.r bacterium.e bacterium.e

Some Lemmons

- Where does cassava grow?

  PARALEX: grow-in.r cassava.e tropical-and-subtropical-regions
  CNNSM: be-grow-by.r cassava.e poor-farmer.e

- Where in the world are watermelon grown?

  PARALEX: be-grow.r japanese-farmer.e square-watermelon.e

  CNNSM: be-grow-in.r watermelon.e different-shape.e

- What is the official theme song of France?

  PARALEX: be-theme-song-for.r marseillaise.e french-revolution.e

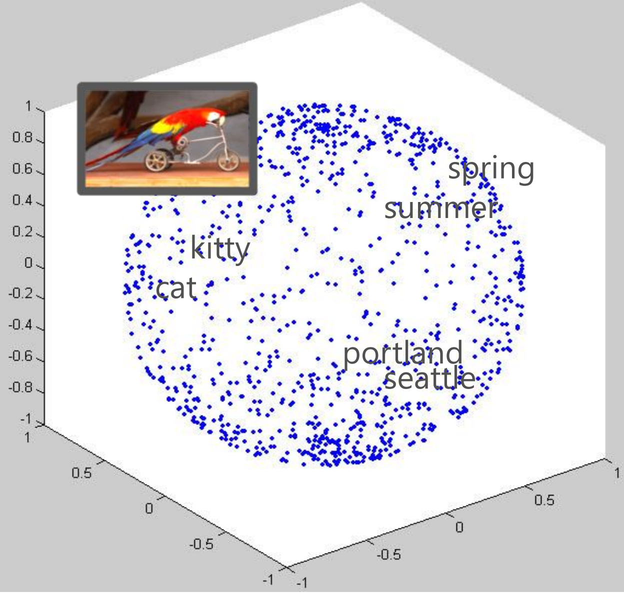
  CNNSM: be-recurrent-theme-in.r song.e mailbox.e

Interim summary

Continuous-space representations are effective for several natural language semantic tasks

- Continuous Word Representations & Lexical Semantics
- Knowledge Base Embedding
- Semantic Parsing & Question Answering

Summary



Great progress in deep learning

breakthrough in speech, image, and language

Exciting advances in learning continuous semantic space

deep models effectively learn semantic representation vectors

- leads to superior performance in a range of NL tasks

learning image and text vectors in an joint semantic space

- facilitates exciting cross-modality scenarios

Learning knowledge-base embedding for entities and relationships

Deep learning for semantic parsing & question answering

Look forward

Building an universal semantic space for all modalities

speech, vision, text, social graph …

Building an universal intelligence space, too

knowledge, reasoning, …

Acquiring intelligence from ambient signals automatically

Deep learning meets big data!

big capacity to digest big data

efficient computation even for small labs: one GPU machine, 10000 cores, learn a billion sentences in one day …

Thank You

Q/A & discussions

Xiaodong He and Scott Wen-tau Yih
xiaohe@microsoft.com, scottyih@microsoft.com

References

- Auli, M., Galley, M., Quirk, C. and Zweig, G., 2013. Joint language and translation modeling with recurrent neural networks. In EMNLP.
- Auli, M., and Gao, J., 2014. Decoder integration and expected bleu training for recurrent neural network language models. In ACL.
- Bejar, I., Chaffin, R. and Embretson, S. 1991. Cognitive and psychometric analysis of analogical problem solving. Recent research in psychology.
- Bengio, Y., 2009. Learning deep architectures for AI. Foundumental Trends Machine Learning, vol. 2.
- Bengio, Y., Courville, A., and Vincent, P. 2013. Representation learning: A review and new perspectives. IEEE Trans. PAMI, vol. 38, pp. 1798-1828.
- Bengio, Y., Ducharme, R., and Vincent, P., 2000. A Neural Probabilistic Language Model, in NIPS.
- Berant, J., and Liang, P. 2014. Semantic parsing via paraphrasing. In ACL.
- Blei, D., Ng, A., and Jordan M. 2001. Latent dirichlet allocation. In NIPS.
- Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J. and Yakhnenko, O. 2013. Translating Embeddings for Modeling Multi-relational Data. In NIPS.
- Bordes, A., Chopra, S., and Weston, J. 2014. Question answering with subgraph embeddings. In EMNLP.
- Bordes, A., Glorot, X., Weston, J. and Bengio Y. 2012. Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing. In AISTATS.
- Brown, P., deSouza, P. Mercer, R., Della Pietra, V., and Lai, J. 1992. Class-based n-gram models of natural language. Computational Linguistics 18 (4).
- Chang, K., Yih, W., and Meek, C. 2013. Multi-Relational Latent Semantic Analysis. In EMNLP.
- Chang, K., Yih, W., Yang, B., and Meek, C. 2014. Typed Tensor Decomposition of Knowledge Bases for Relation Extraction. In EMNLP.
- Collobert, R., and Weston, J. 2008. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In ICML.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P., 2011. Natural language processing (almost) from scratch. in JMLR, vol. 12.
- Dahl, G., Yu, D., Deng, L., and Acero, 2012. A. Context-dependent, pre-trained deep neural networks for large vocabulary speech recognition, IEEE Trans. Audio, Speech, & Language Proc., Vol. 20 (1), pp. 30-42.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T., and Harshman, R. 1990. Indexing by latent semantic analysis. J. American Society for Information Science, 41(6): 391-407
- Deng, L., Seltzer, M., Yu, D., Acero, A., Mohamed, A., and Hinton, G., 2010. Binary Coding of Speech Spectrograms Using a Deep Auto-encoder, in Interspeech.
- Deng, L., Tur, G, He, X, and Hakkani-Tur, D. 2012. Use of kernel deep convex networks and end-to-end learning for spoken language understanding, Proc. IEEE Workshop on Spoken Language Technologies.
- Deng, L., Yu, D. and Acero, A. 2006. Structured speech modeling, IEEE Trans. on Audio, Speech and Language Processing, vol. 14, no. 5, pp. 1492-1504.
- Deng, L., Yu, D., and Platt, J. 2012. Scalable stacking and learning for building deep architectures, Proc. ICASSP.
- Deoras, A., and Sarikaya, R., 2013. Deep belief network based semantic taggers for spoken language understanding, in INTERSPEECH.
- Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., and Makhoul, J., 2014. Fast and Robust Neural Network Joint Models for Statistical Machine Translation, ACL.

References

- Fader, A., Zettlemoyer, L., and Etzioni, O. 2013. Paraphrase-driven learning for open question answering. In ACL.
- Fang, H., Gupta, S., Iandola, F., Srivastava, R., Deng, L., Dollár, P., Gao, J., He, X., Mitchell, M., Platt, J., Zitnick, L., Zweig, G., "From Captions to Visual Concepts and Back," arXiv:1411.4952
- Firth, J. R. 1957. *Papers in Linguistics 1934–1951*, Oxford University Press, 1957
- Frome, A., Corrado, G., Shlens, J., Bengio, S., Dean, J., Ranzato, M., and Mikolov, T., 2013. DeViSE: A Deep Visual-Semantic Embedding Model, Proc. NIPS.
- Gao, J., He, X., Yih, W-t., and Deng, L. 2014a. Learning continuous phrase representations for translation modeling. In ACL.
- Gao, J., He, X., and Nie, J-Y. 2010. Clickthrough-based translation models for web search: from word models to phrase models. In CIKM.
- Gao, J., Pantel, P., Gamon, M., He, X., and Deng, L. 2014. Modeling interestingness with deep neural networks. In EMNLP
- Gao, J., Toutanova, K., Yih., W-T. 2011. Clickthrough-based latent semantic models for web search. In SIGIR.
- Gao, J., Yuan, W., Li, X., Deng, K., and Nie, J-Y. 2009. Smoothing clickthrough data for web search ranking. In SIGIR.
- Gao, J., and He, X. 2013. Training MRF-based translation models using gradient ascent. In NAACL-HLT.
- Getoor, L., and Taskar, B. editors. 2007. Introduction to Statistical Relational Learning. The MIT Press.
- Graves, A., Jaitly, N., and Mohamed, A., 2013a. Hybrid speech recognition with deep bidirectional LSTM, Proc. ASRU.
- Graves, A., Mohamed, A., and Hinton, G., 2013. Speech recognition with deep recurrent neural networks, Proc. ICASSP.
- He, X. and Deng, L., 2013. Speech-Centric Information Processing: An Optimization-Oriented Approach, in Proceedings of the IEEE.
- He, X. and Deng, L., 2012. Maximum Expected BLEU Training of Phrase and Lexicon Translation Models , ACL.
- He, X., Deng, L., and Chou, W., 2008. Discriminative learning in sequential pattern recognition, Sept. IEEE Sig. Proc. Mag.
- Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B., 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition, IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82-97.
- Hinton, G., and Salakhutdinov, R., 2010. Discovering binary codes for documents by learning deep generative models. Topics in Cognitive Science.
- Hu, Y., Auli, M., Gao, Q., and Gao, J. 2014. Minimum translation modeling with recurrent neural networks. In EACL.
- Huang, E., Socher, R., Manning, C, and Ng, A. 2012. Improving word representations via global context and multiple word prototypes, Proc. ACL.
- Huang, P., He, X., Gao, J., Deng, L., Acero, A., and Heck, L. 2013. Learning deep structured semantic models for web search using clickthrough data. In CIKM.
- Hutchinson, B., Deng, L., and Yu, D., 2012. A deep architecture with bilinear modeling of hidden representations: Applications to phonetic recognition, Proc. ICASSP.
- Hutchinson, B., Deng, L., and Yu, D., 2013. Tensor deep stacking networks, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 35, pp. 1944 - 1957.
- Jurgens, D., Mohammad, S., Turney, P. and Holyoak, K. 2012. SemEval-2012 Task 2: Measuring degrees of relational similarity. In SemEval.
- Kiros, R., Zemel, R., and Salakhutdinov, R. 2013. Multimodal Neural Language Models, Proc. NIPS Deep Learning Workshop.
- Koehn, P. 2009. Statistical Machine Translation. Cambridge University Press.
- Krizhevsky, A., Sutskever, I, and Hinton, G., 2012. ImageNet Classification with Deep Convolutional Neural Networks, NIPS.
- Landauer. T., 2002. On the computational basis of learning and cognition: Arguments from LSA. Psychology of Learning and Motivation, 41:43–84.

References

- Lao, N., Mitchell, T., and Cohen, W. 2011. Random walk inference and learning in a large scale knowledge base. In EMNLP.
- Le, H-S, Oparin, I., Allauzen, A., Gauvain, J-L., Yvon, F., 2013. Structured output layer neural network language models for speech recognition, IEEE Transactions on Audio, Speech and Language Processing.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. 1998. Gradient-based learning applied to document recognition, Proceedings of the IEEE, Vol. 86, pp. 2278-2324.
- Levy, O., and Goldberg, Y. 2014. Linguistic Regularities in Sparse and Explicit Word Representations. In CoNLL.
- Li, P., Hastie, T., and Church, K.. 2006. Very sparse random projections, in Proc. SIGKDD.
- Mesnil, G., He, X., Deng, L., and Bengio, Y., 2013. Investigation of Recurrent-Neural-Network Architectures and Learning Methods for Spoken Language Understanding, in Interspeech.
- Mikolov, T. 2012. Statistical Language Models based on Neural Networks, Ph.D. thesis, Brno University of Technology.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. 2013. Efficient estimation of word representations in vector space, Proc. ICLR.
- Mikolov, T., Kombrink,. S., Burget, L., Cernocky, J., Khudanpur, S., 2011. Extensions of Recurrent Neural Network LM. ICASSP.
- Mikolov, T., Yih, W., Zweig, G., 2013. Linguistic Regularities in Continuous Space Word Representations. In NAACL-HLT.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. 2013. Distributed Representations of Words and Phrases and their Compositionality. In NIPS.
- Mohamed, A., Yu, D., and Deng, L. 2010. Investigation of full-sequence training of deep belief networks for speech recognition, Proc. Interspeech.
- Mohammad, S., Dorr, Bonnie., and Hirst, G. 2008. Computing word pair antonymy. In EMNLP.
- Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. 2011. Multimodal deep learning, Proc. ICML.
- Nickel, M., Tresp, V., and Kriegel, H. 2011. A three-way model for collective learning on multi-relational data. In ICML.
- Reddy, S., Lapata, M., and Steedman, M. 2014. Large-scale semantic parsing without question-answer pairs. Transactions of the Association for Computational Linguistics (TACL).

References

- Sainath, T., Mohamed, A., Kingsbury, B., and Ramabhadran, B. 2013. Convolutional neural networks for LVCSR, Proc. ICASSP.
- Salakhutdinov R., and Hinton, G., 2007 Semantic hashing. in Proc. SIGIR Workshop Information Retrieval and Applications of Graphical Models
- Salton, G. and McGill, M. 1983. Introduction to Modern Information Retrieval. McGraw Hill.
- Sarikaya, R., Hinton, G., and Ramabhadran, B., 2011. Deep belief nets for natural language call-routing, in Proceedings of the ICASSP.
- Schwenk, H., Dchelotte, D., Gauvain, J-L., 2006. Continuous space language models for statistical machine translation, in COLING-ACL
- Seide, F., Li, G., and Yu, D. 2011. Conversational speech transcription using context-dependent deep neural networks, Proc. Interspeech
- Shen, Y., He, X., Gao, J., Deng, L., and Mesnil, G. 2014. Learning Semantic Representations Using Convolutional Neural Networks for Web Search, in Proceedings of WWW.
- Shen, Y., He, X., Gao, J., Deng, L., and Mesnil, G. 2014. A convolutional latent semantic model for web search. CIKM
- Socher, R., Chen, D., Manning, C., and Ng, A. 2013. Reasoning With Neural Tensor Networks For Knowledge Base Completion. In NIPS.
- Socher, R., Huval, B., Manning, C., Ng, A., 2012. Semantic compositionality through recursive matrix-vector spaces. In EMNLP.
- Socher, R., Lin, C., Ng, A., and Manning, C. 2011. Learning continuous phrase representations and syntactic parsing with recursive neural networks, Proc. ICML.
- Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C., Ng A., and Potts. C. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, Proc. EMNLP
- Song, X. He, X., Gao. J., and Deng, L. 2014. Learning Word Embedding Using the DSSM. MSR Tech Report.
- Song, Y., Wang, H., and He, X., 2014. Adapting Deep RankNet for Personalized Search. Proc. WSDM.
- Tur, G., Deng, L., Hakkani-Tur, D., and He, X., 2012. Towards Deeper Understanding Deep Convex Networks for Semantic Utterance Classification, in ICASSP.
- Turney P. 2008. A uniform approach to analogies, synonyms, antonyms, and associations. In COLING.
- Wright, S., Kanevsky, D., Deng, L., He, X., Heigold, G., and Li, H., 2013. Optimization Algorithms and Applications for Speech and Language Processing, in IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, no. 11.
- Xu, P., and Sarikaya, R., 2013. Convolutional neural network based triangular crf for joint intent detection and slot filling, in IEEE ASRU.
- Yang, B., Yih, W., He, X., Gao, J., and Deng L. 2014. In NIPS-2014 Workshop Learning Semantics.
- Yao, K., Zweig, G., Hwang, M-Y. , Shi, Y., Yu, D., 2013. Recurrent neural networks for language understanding, submitted to Interspeech.
- Yann, D., Tur, G., Hakkani-Tur, D., Heck, L., 2014. Zero-Shot Learning and Clustering for Semantic Utterance Classification Using Deep Learning, in ICLR.
- Yih, W., Toutanova, K., Platt, J., and Meek, C. 2011. Learning discriminative projections for text similarity measures. In CoNLL.
- Yih, W., Zweig, G., Platt, J. 2012. Polarity Inducing Latent Semantic Analysis. In EMNLP-CoNLL.
- Yih, W., He, X., Meek, C. 2014. Semantic Parsing for Single-Relation Question Answering, in ACL.
- Zeiler, M. and Fergus, R. 2013. Visualizing and understanding convolutional networks, arXiv:1311.2901, pp. 1-11.


2014 Spoken Language Technology Workshop
December 7-10, 2014 • South Lake Tahoe, Nevada, USA
IEEE
IEEE
Signal Processing Society