Deep Learning and Continuous Representations for Natural Language Processing

Scott Wen-tau Yih, Xiaodong He and Jianfeng Gao

Microsoft Research, Redmond, WA

Tutorial presented at NAACL-HLT, May 31st, 2015

Tutorial Outline

Jianfeng Gao

- Part I: Background
- Part II: Deep learning in statistical machine translation (SMT)

Xiaodong He

- Part III: Learning semantic representations

Scott Yih

- Part IV: Natural language understanding
- Part V: Conclusion

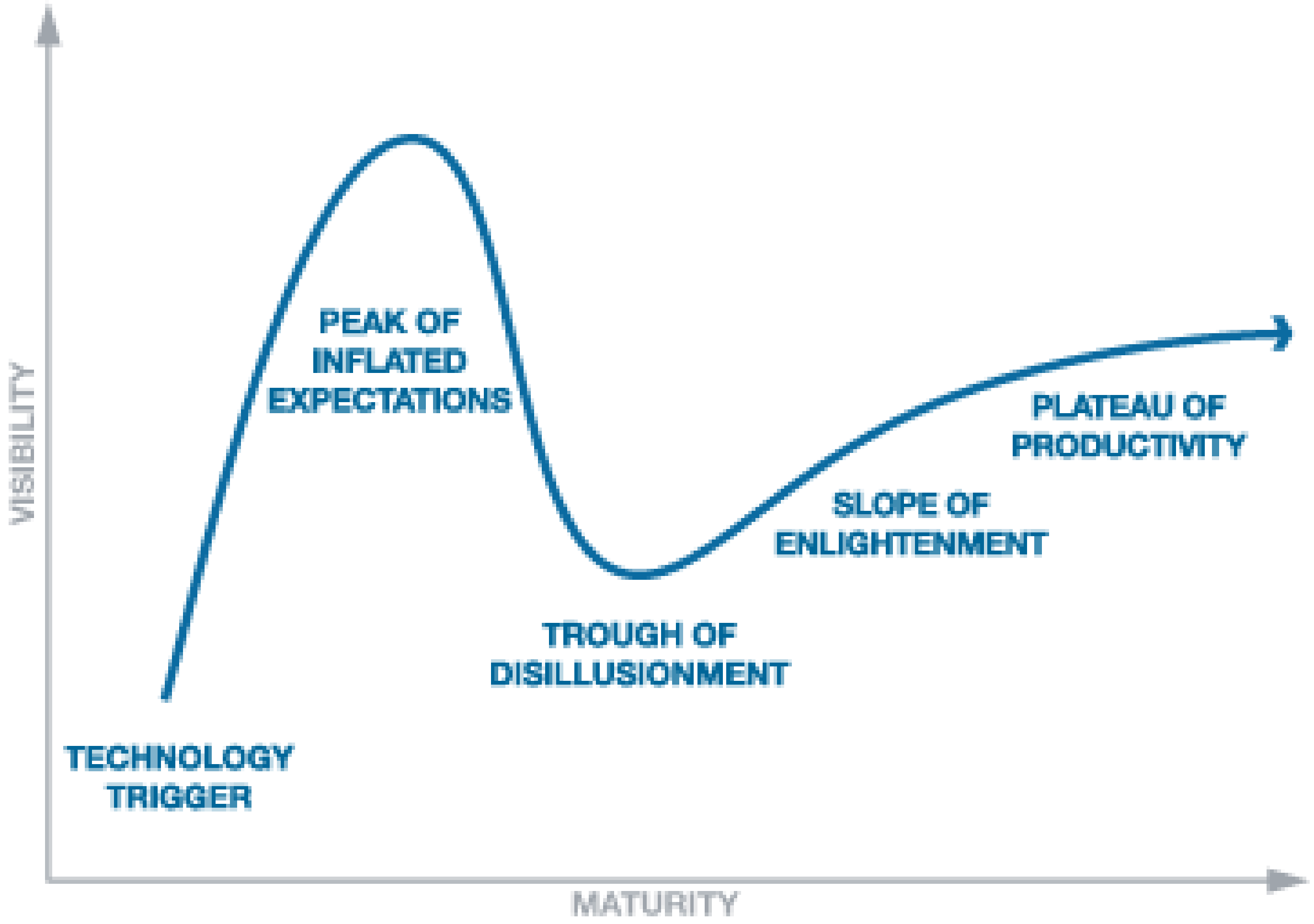
Part I

Background

Tutorial Outline

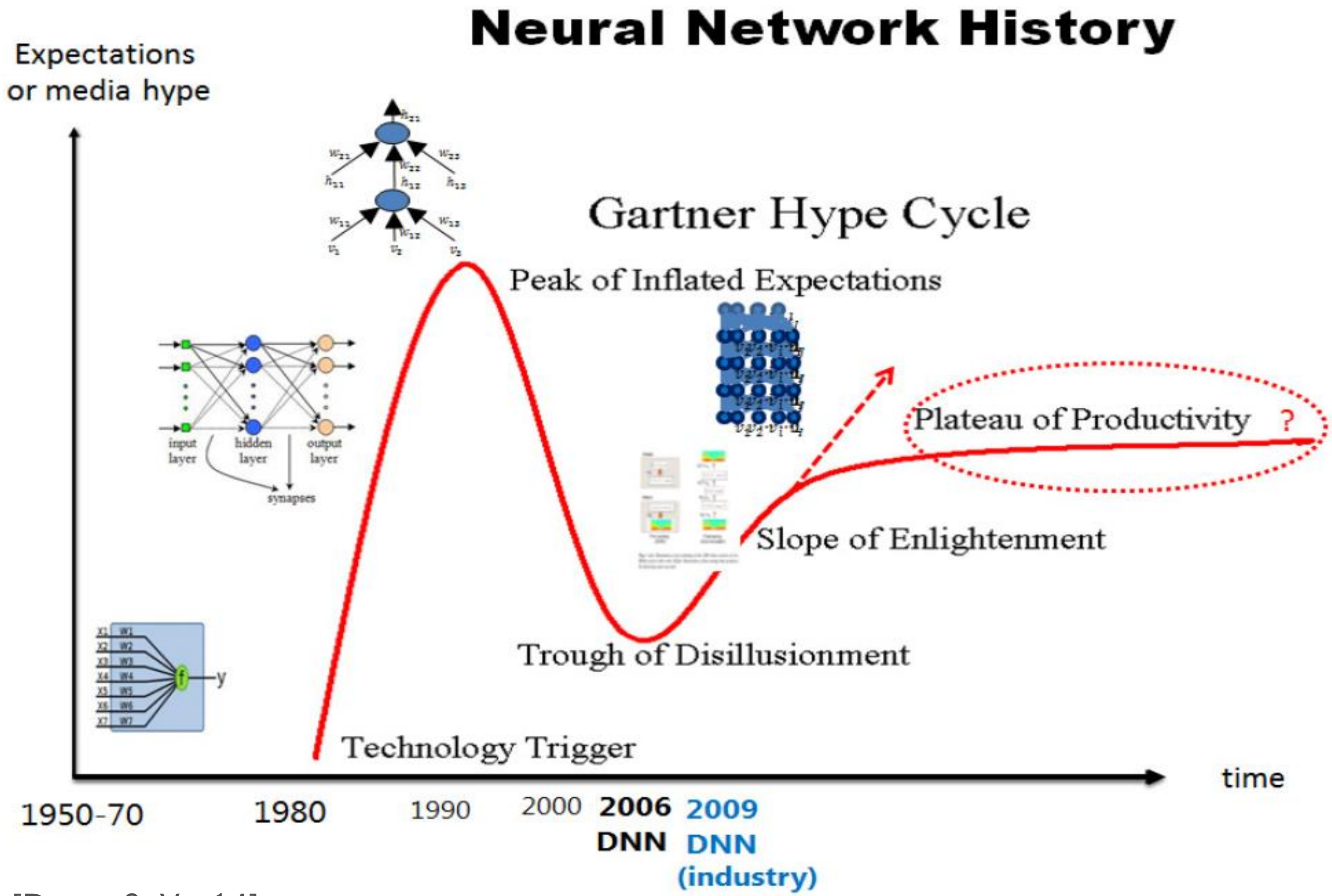
- Part I: Background
  - A brief history of deep neural networks (DNN)
  - An example of neural models for query classification
  - From classification to semantic similarity
- Part II: Deep learning in statistical machine translation (SMT)
- Part III: Learning semantic representations
- Part IV: Natural language understanding
- Part V: Conclusion

Gartner hype cycle

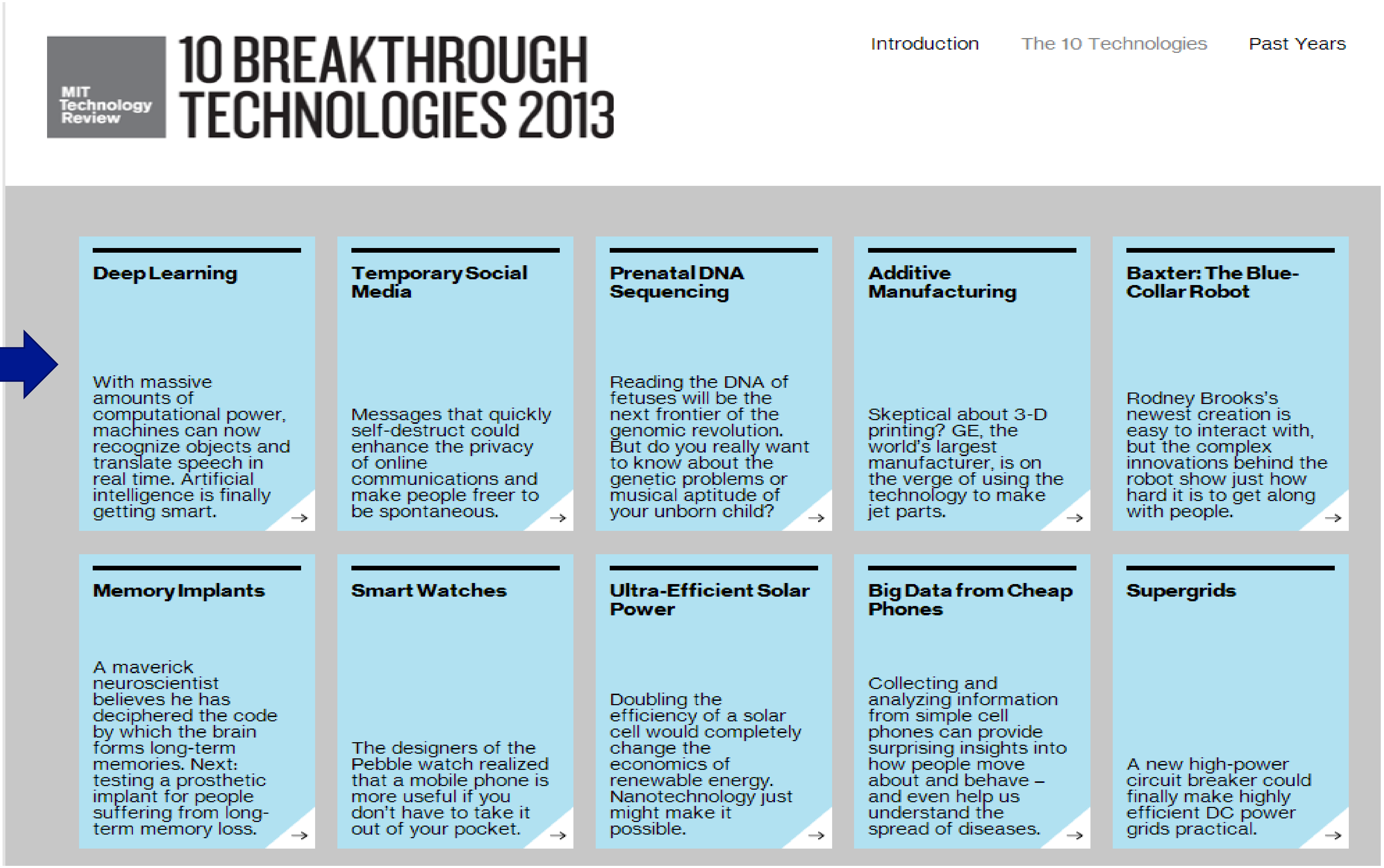


Gartner hype cycle

A brief history of deep neural networks (DNN)



[Deng & Yu 14]



MIT Technology Review

10 BREAKTHROUGH TECHNOLOGIES 2013

Deep Learning

With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart.

Temporary Social Media

Messages that quickly self-destruct could enhance the privacy of online communications and make people freer to be spontaneous.

Prenatal DNA Sequencing

Reading the DNA of fetuses will be the next frontier of the genomic revolution. But do you really want to know about the genetic problems or musical aptitude of your unborn child?

Additive Manufacturing

Skeptical about 3-D printing? GE, the world's largest manufacturer, is on the verge of using the technology to make jet parts.

Baxter: The Blue-Collar Robot

Rodney Brooks's newest creation is easy to interact with, but the complex innovations behind the robot show just how hard it is to get along with people.

Memory Implants

A maverick neuroscientist believes he has deciphered the code by which the brain forms long-term memories. Next: testing a prosthetic implant for people suffering from long-term memory loss.

Smart Watches

The designers of the Pebble watch realized that a mobile phone is more useful if you don't have to take it out of your pocket.

Ultra-Efficient Solar Power

Doubling the efficiency of a solar cell would completely change the economics of renewable energy. Nanotechnology just might make it possible.

Big Data from Cheap Phones

Collecting and analyzing information from simple cell phones can provide surprising insights into how people move about and behave – and even help us understand the spread of diseases.

Supergrids

A new high-power circuit breaker could finally make highly efficient DC power grids practical.

Geoff Hinton

The universal translator on "Star Trek" comes true…

The New York Times

Scientists See Promise in Deep-Learning Programs

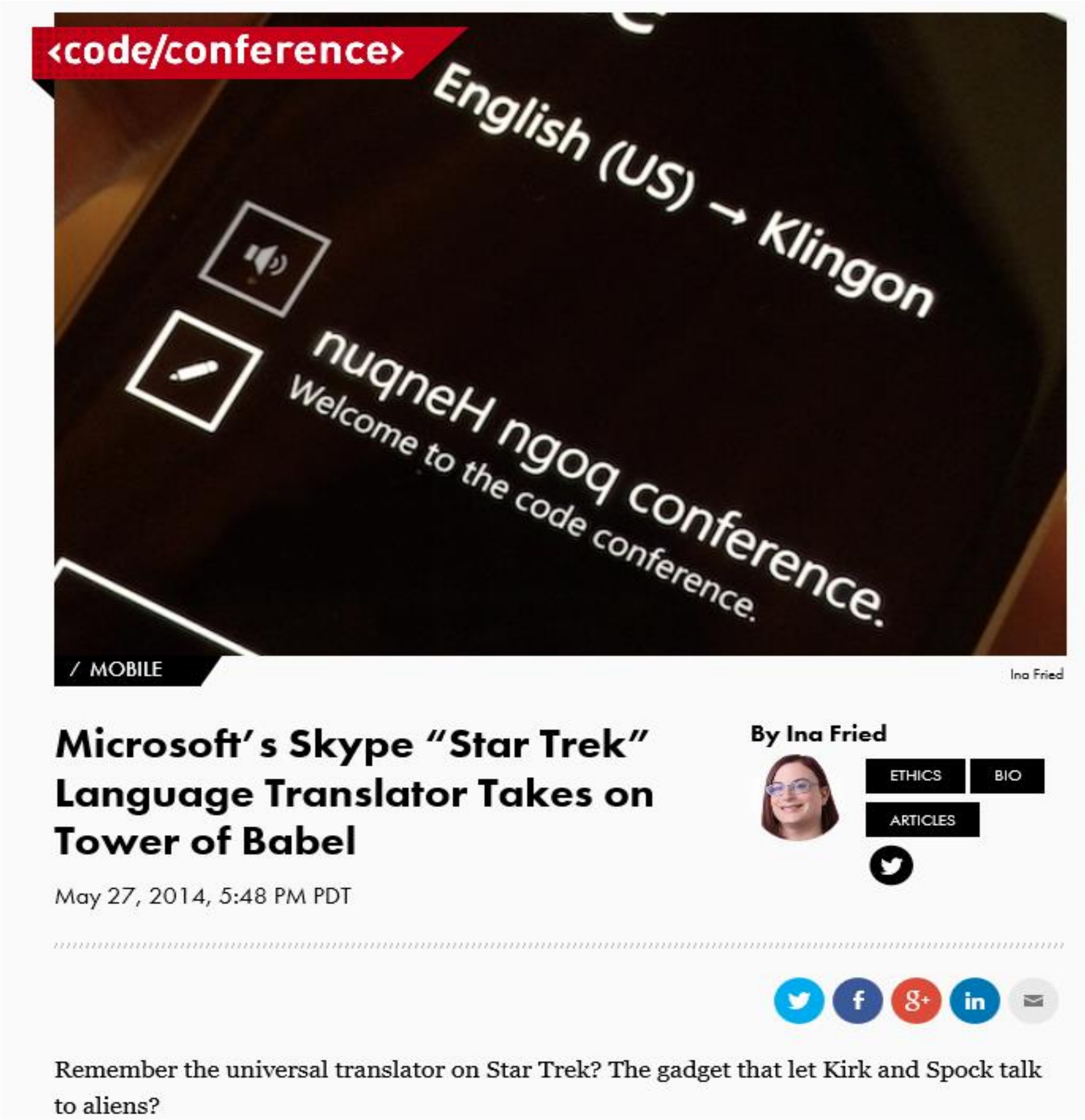
John Markoff November 23, 2012

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 Chinese.

Skype to get 'real-time' translator

Analysts say the translation feature could have wide ranging applications



Microsoft's Skype "Star Trek" Language Translator Takes on Tower of Babel

By Ina Fried

May 27, 2014, 5:48 PM PDT

Remember the universal translator on Star Trek? The gadget that let Kirk and Spock talk to aliens?

Impact of deep learning in speech technology

讯飞语点

您需要做什么?

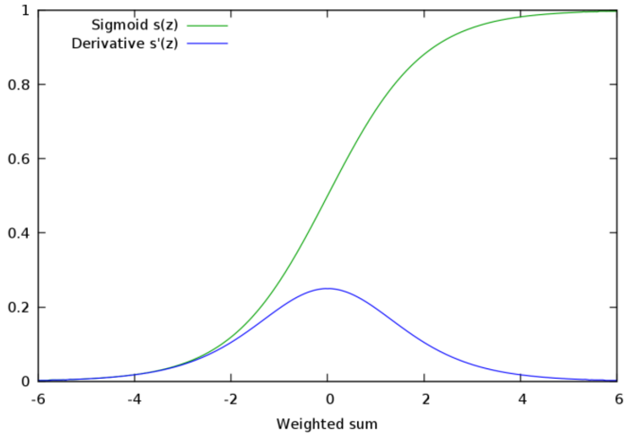
打开手机中国

正在打开手机中国

点击说话

"Early" DNNs are difficult to train

- "Shallow" NN improves acoustic modeling in early 90's
  - But the benefits were not sufficient to challenge GMMs
- Lack of hardware and algorithms to train DNN
  - Scalability problem, i.e., training NN with many hidden layers on large amounts of data
  - Non-convex optimization with a lot of local optima
  - Vanishing/exploding gradient problem
    - Forward prop: repeated multiplication of s(z)
    - Back prop: repeated multiplication of s'(z)



Breakthroughs after 2006

- Computational power due to the use of GPU and large-scale CPU clusters
- Better learning algorithms and different nonlinearities
  - SGD allows the training to jump out of local optima due to the noisy gradients estimated from a small batch of samples.
  - SGD is effective for parallelizing over many machines with an asynchronous mode
  - Tricks: Dropout, Rectified Linear Units (ReLUs)
- Use deep belief net (DBN) for initialization – Layer-wise pre-training [Hinton+ 06]

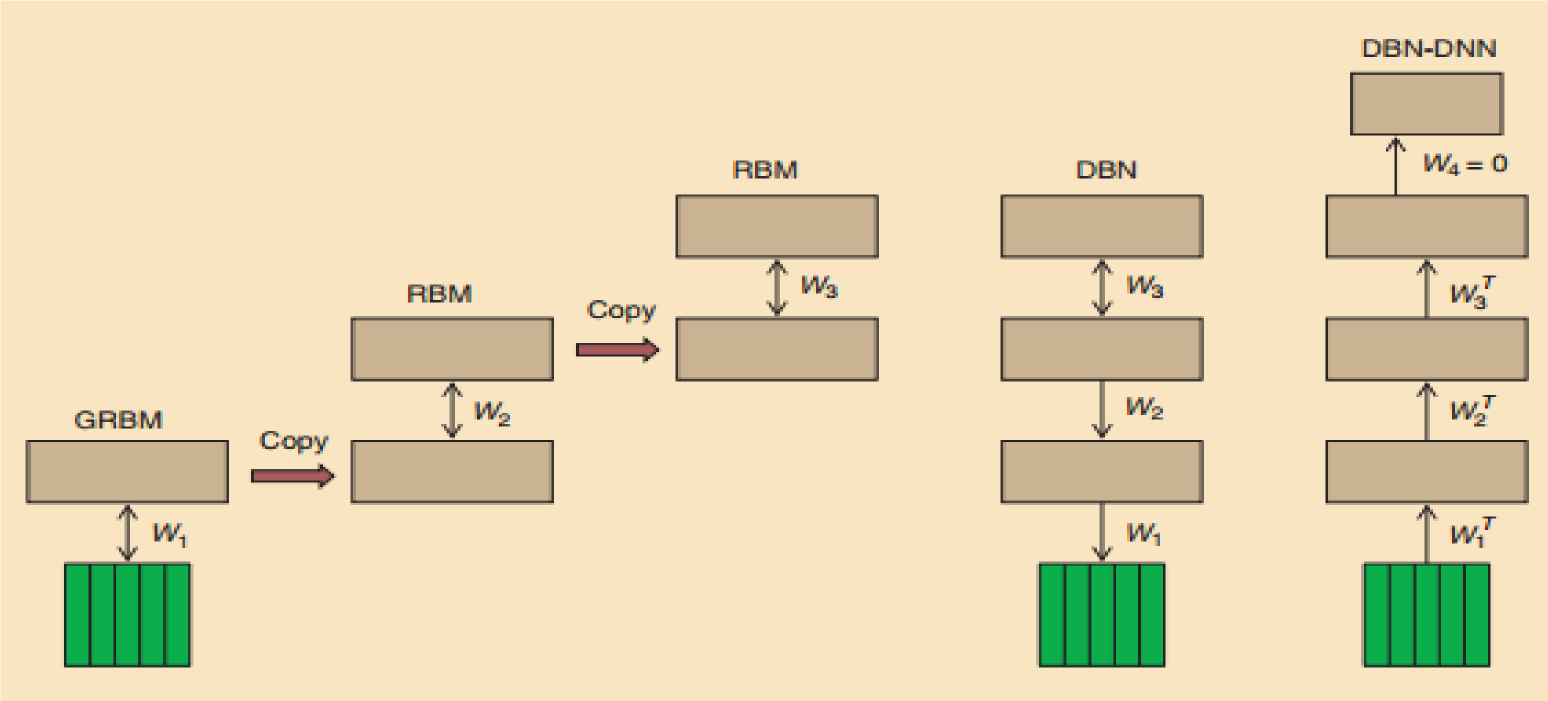
Geoff Hinton

Li Deng

Dong Yu

DNN: (Fully-Connected) Deep Neural Networks

Hinton, Deng, Yu, et al., DNN for AM in speech recognition, *IEEE SPM*, 2012



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.

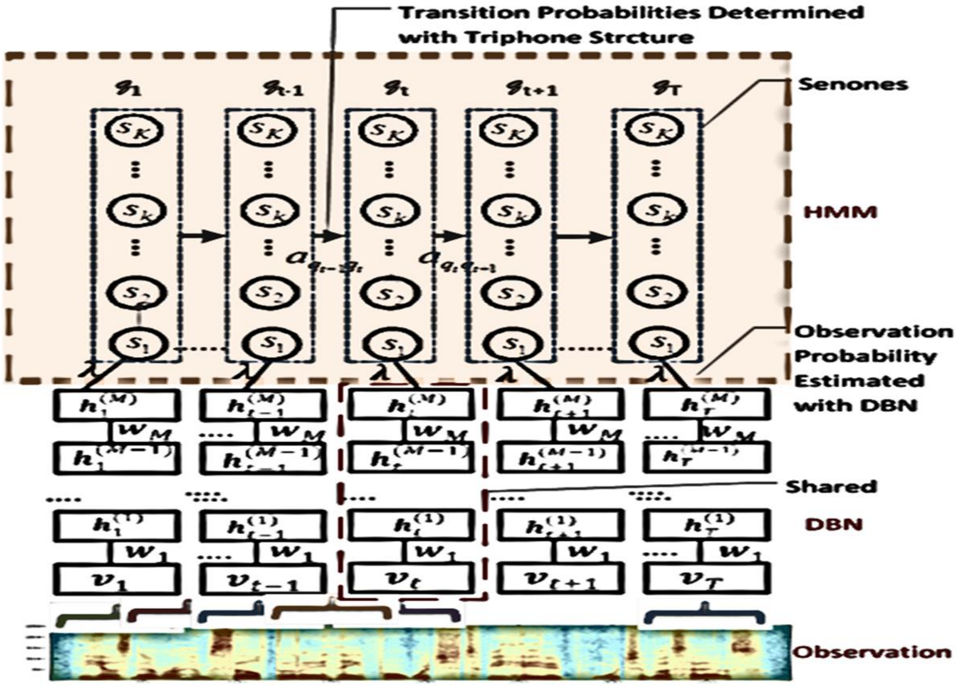
Then compose them into a single Deep Belief Network.

Then add outputs and train the DNN with backprop.

CD-DNN-HMM

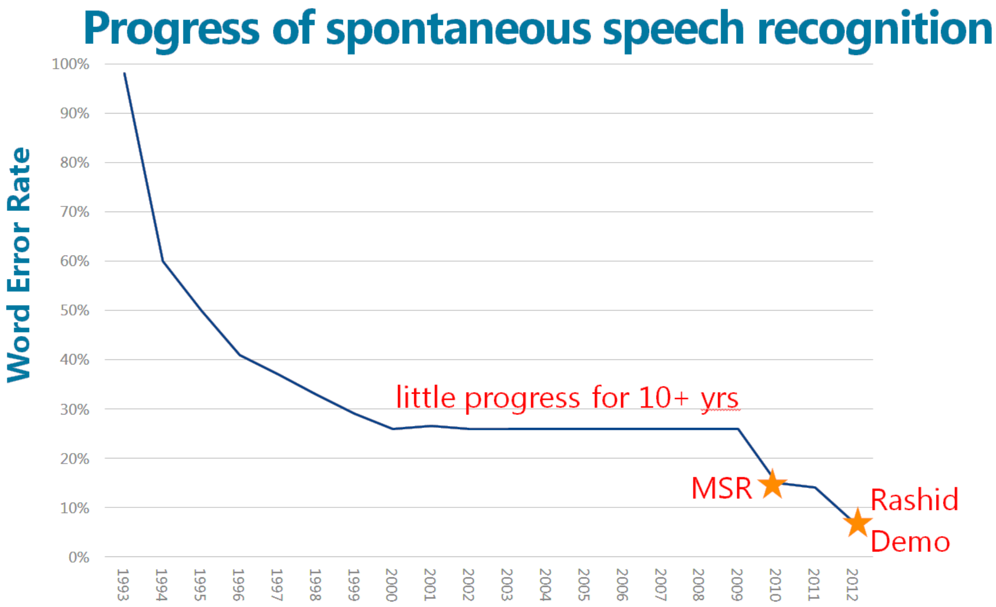
Dahl, Yu, Deng, and Acero, "Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition," *IEEE Trans. ASLP*, Jan. 2012

Seide, Li, and Yu, "Conversational Speech Transcription using Context-Dependent Deep Neural Networks," *INTERSPEECH* 2011.



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)!



The focus of this tutorial

- Is not on speech or image,
- But on text processing and understanding tasks
  - Statistical machine translation
  - Information retrieval
  - Image captioning
  - Question answering
  - Etc.

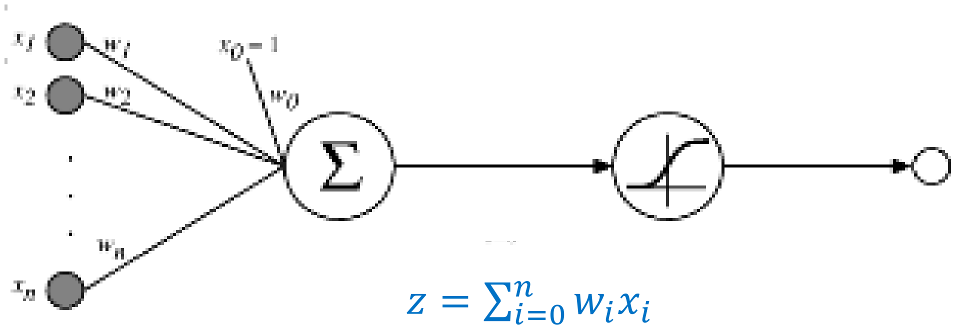
A query classification problem

- Given a search query $\boldsymbol{q}$, e.g., “denver sushi downtown”
- Identify its domain $\boldsymbol{c}$ e.g.,
  - Restaurant
  - Hotel
  - Nightlife
  - Flight
  - etc.
- So that a search engine can tailor the interface and result to provide a richer personalized user experience

A single neuron model

- For each domain $\boldsymbol{c}$, build a binary classifier
  - Input: represent a query q as a vector of features $x = [x_1, \dots x_n]^T$
  - Output: $y = P(1|q, c)$
  - q is labeled c is$P(1|q, c) > 0.5$
- Input feature vector, e.g., a bag of words vector
  - Regards words as atomic symbols: *denver, sushi, downtown*
  - Each word is represented as a one-hot vector: $[0, \dots, 0,1,0, \dots, 0]^T$
  - Bag of words vector = sum of one-hot vectors
  - We may use other features, such as n-grams, phrases, (hidden) topics

A single neuron model



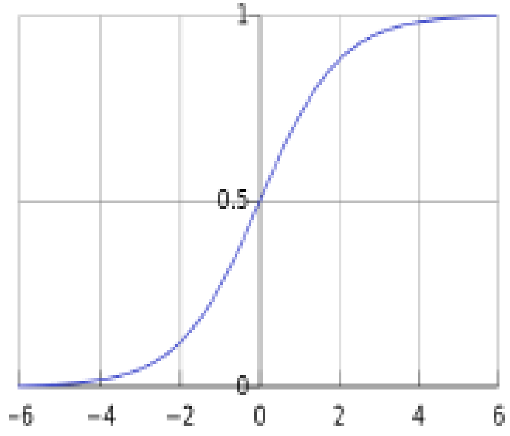
Input features x

$z = \sum_{i=0}^{n} w_i x_i$

Output: $P(1|q, c)$

$y = \sigma(z) = \frac{1}{1+\exp(-z)}$

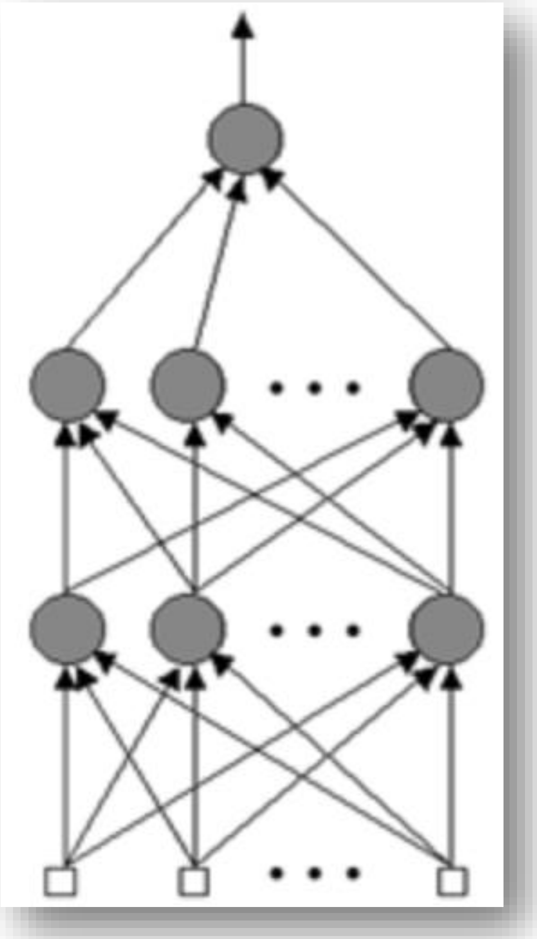
- w: weight vector to be learned
- z: weighted sum of input features
- σ: the logistic function
  - Turn a score to a probability
  - non-linear activation function, essential in DNN models



Model training: how to assign $\boldsymbol{w}$

- Training data: a set of $\left(x^{(m)}, y^{(m)}\right)_{m=\{1,2,\ldots,M\}}$ pairs
  - Input $x^{(m)} \in R^n$
  - Output $y^{(m)} = \{0,1\}$
- optimize parameters $\boldsymbol{w}$ on training data
  - minimize a loss function (mean square error loss)
    - $\min_{w} \sum_{m=1}^{M} L^m$
    - where $L^{(m)} = \frac{1}{2}\left(f_w\left(x^{(m)}\right) - y^{(m)}\right)^2$
  - Using Stochastic Gradient Descent (SGD)
    - Initialize w randomly
    - Update for each training sample until convergence: $w^{new} = w^{old} - \eta \frac{\partial L}{\partial w}$

Multi-layer (deep) neural networks



Output layer $y^o = \sigma(w^T y^2)$

Vector w

2st hidden layer $y^2 = \sigma(\mathbf{W}_2 y^1)$

Projection matrix $\mathbf{W}_2$

1st hidden layer $y^1 = \sigma(\mathbf{W}_1 x)$

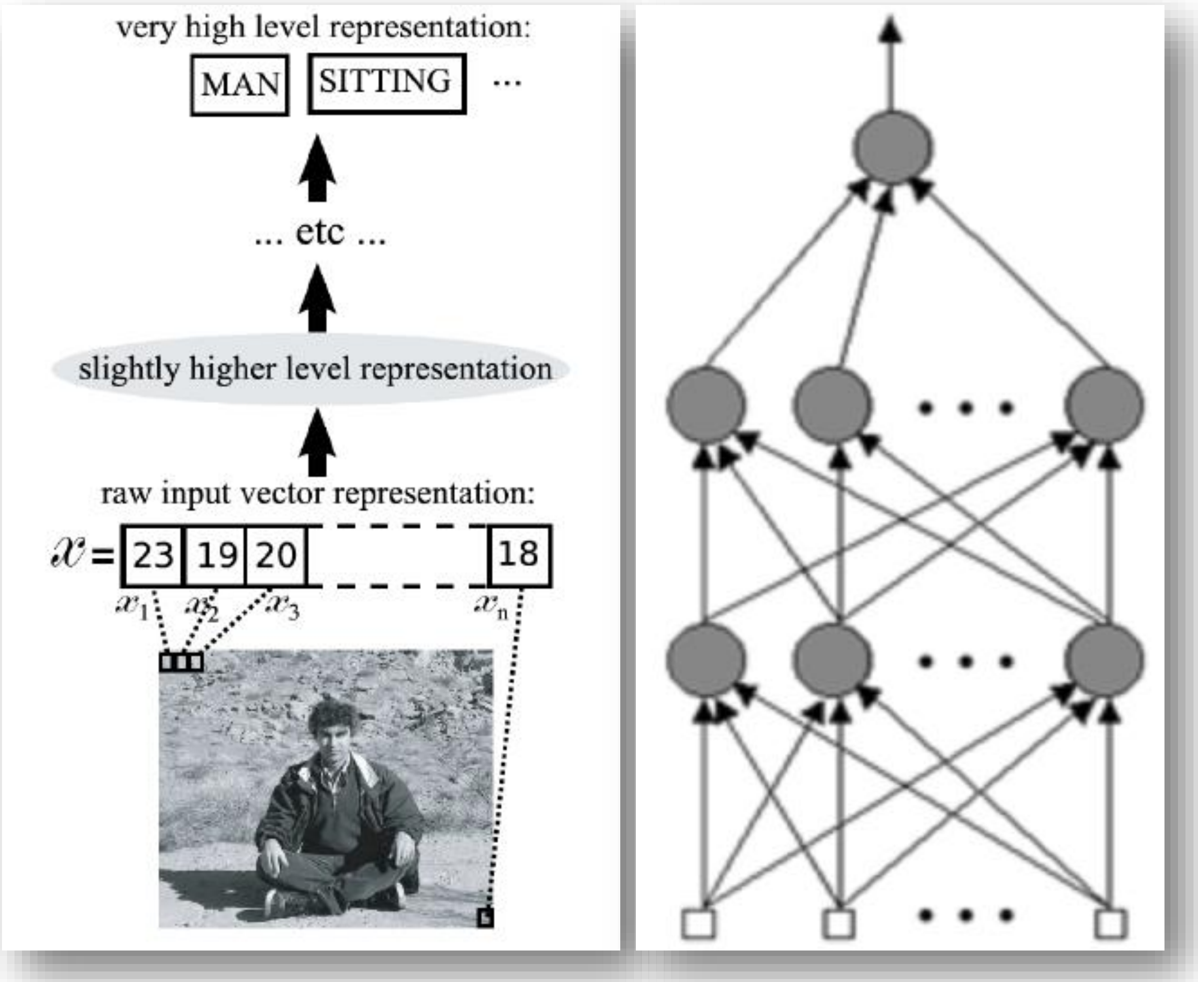
Projection matrix $\mathbf{W}_1$

Input features x

This is exactly the **single neuron model** with **hidden** features.

Feature generation: project raw input features (bag of words) to **hidden** features (topics).

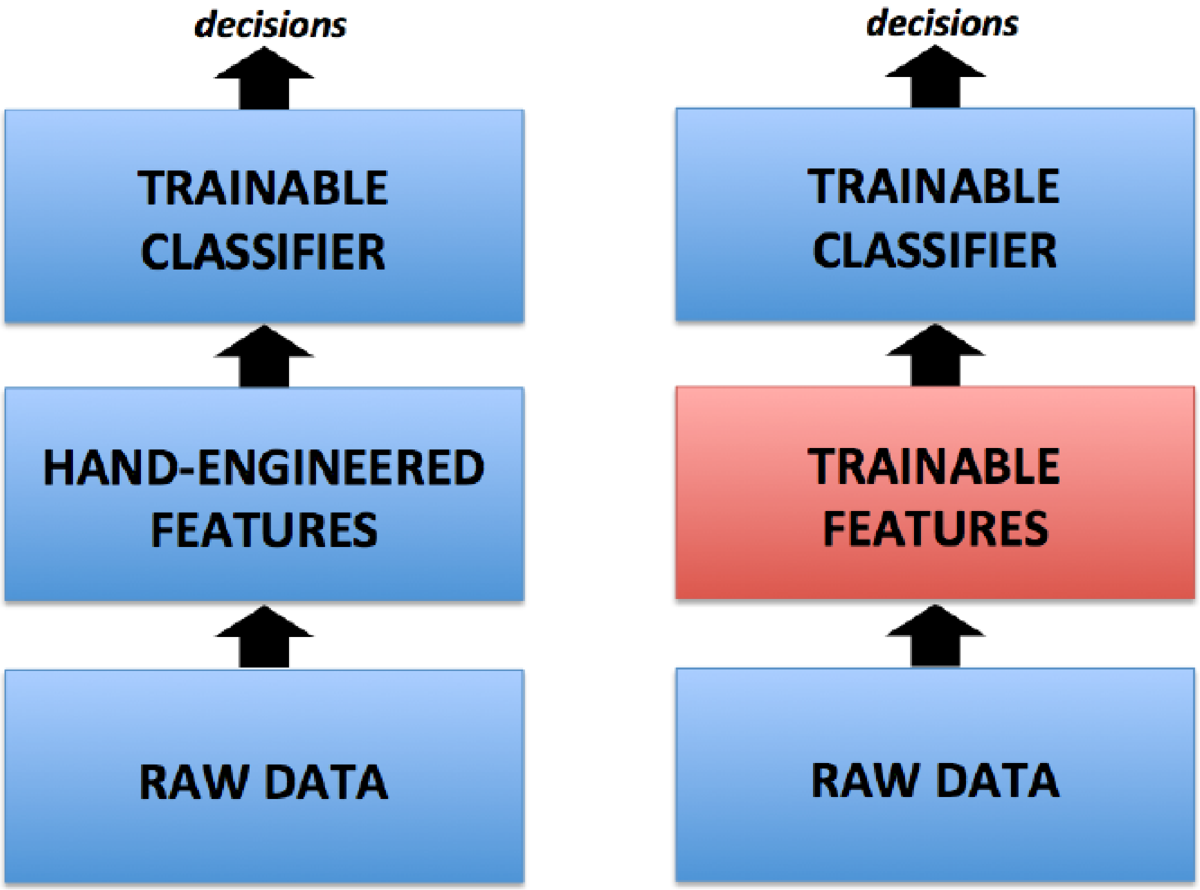
DNN for image processing



[Bengio 09]

This is exactly the **single neuron model** with **hidden** features.

Project raw input features to **hidden** features (high level representation).



Standard Machine Learning Process

decisions

TRAINABLE CLASSIFIER

HAND-ENGINEERED FEATURES

RAW DATA

Deep Learning

decisions

TRAINABLE CLASSIFIER

TRAINABLE FEATURES

RAW DATA

Adapted from [Duh 14]

Deep Semantic Similarity Model (DSSM)

[Huang+ 13; Gao+ 14a; Gao+ 14b; Shen+ 14, Yih+ 15]

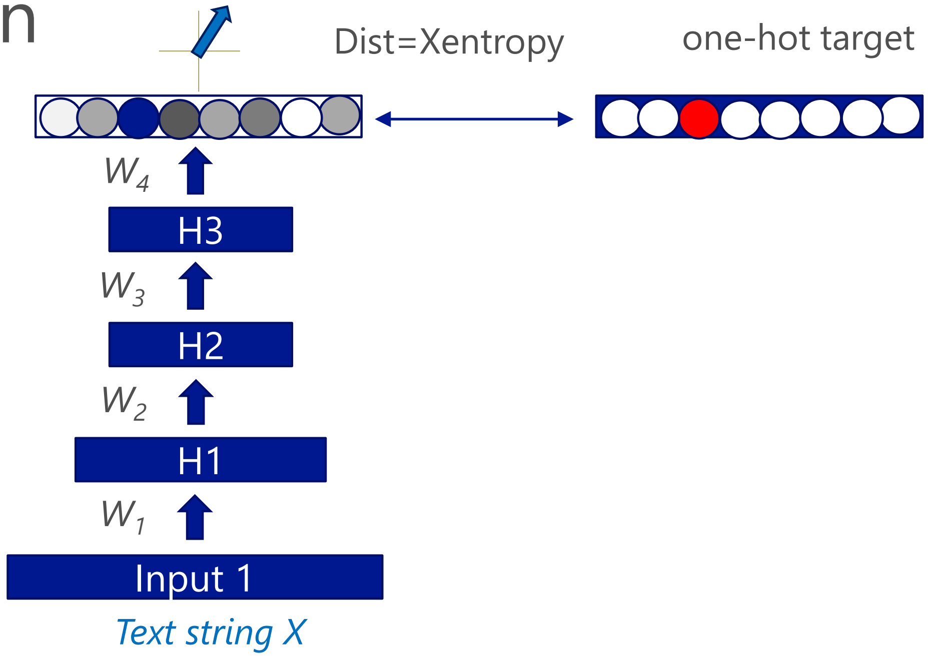
- Compute semantic similarity btw text strings X and Y
  - Map X and Y to feature vectors in a latent semantic space via deep neural net
  - Compute the cosine similarity between the feature vectors
  - Also called "Deep Structured Similarity Model" in [Huang+ 13]

- DSSM for NLP tasks

Tasks	**X**	**Y**
Machine translation	*Text in language A*	*Translation in language B*
Web search	*Search query*	*Web document*
Image captioning	*Image*	*Caption*
Question Answering	*Question*	*Answer*

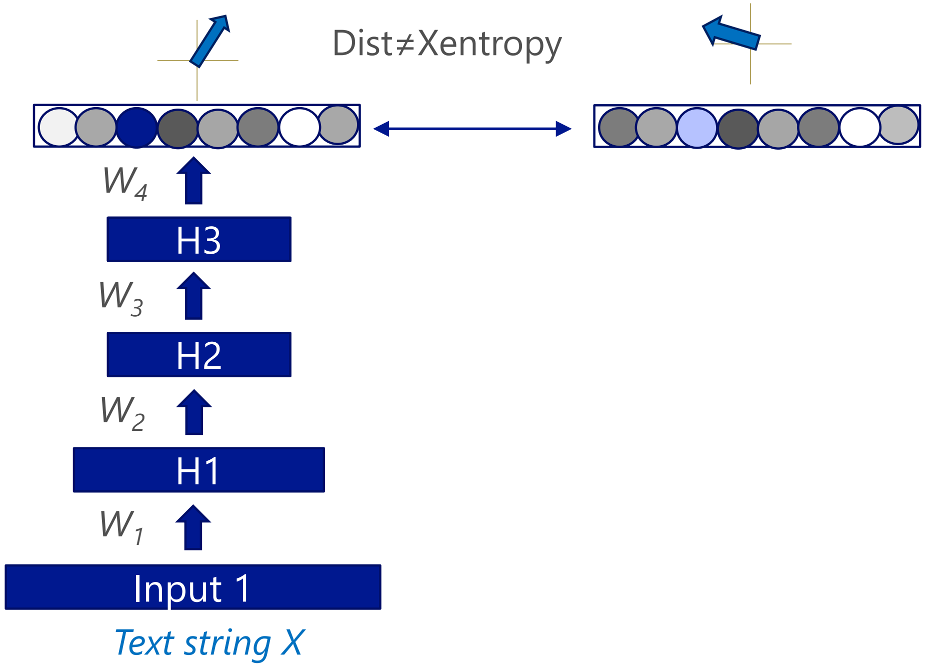
From common DNNs to DSSM

- Common DNN models
  - Mainly for classification
  - Target: one-hot vector
  - Example of DNN:



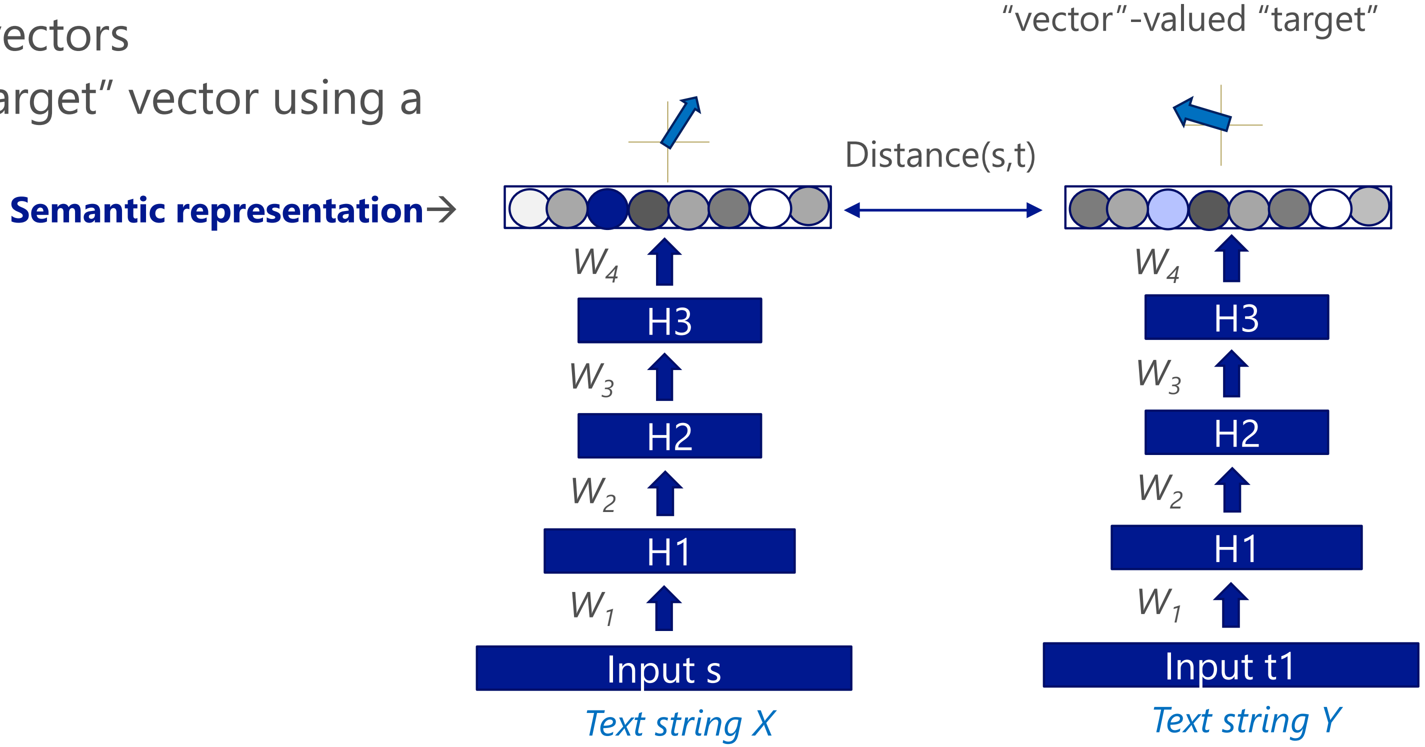
From common DNNs to DSSM

- To construct a DSSM
  - For ranking (not classification with DNN)
  - Step 1: target from “one-hot” to continuous-valued vectors



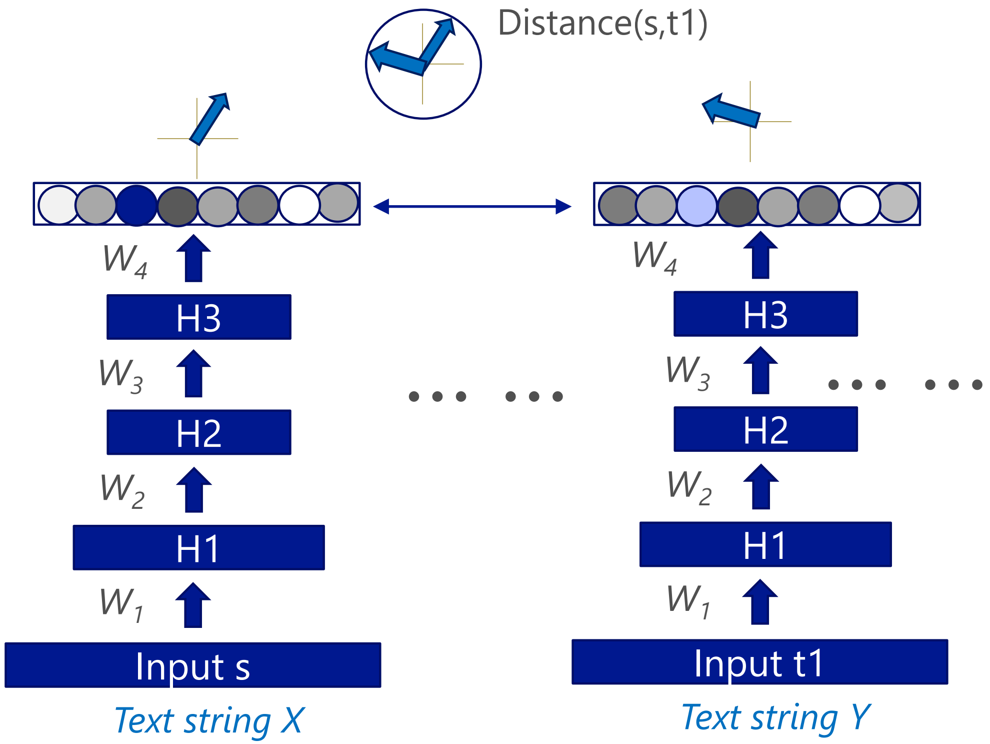
From common DNNs to DSSM

- To construct a DSSM
  - Step 1: target from “one-hot” to continuous-valued vectors
  - Step 2: derive the “target” vector using a DNN



From common DNNs to DSSM

- To construct a DSSM
  - Step 1: target from "one-hot" to continuous-valued vectors
  - Step 2: derive the "target" vector using a DNN
  - Step 3: normalize two "semantic" vectors & compute their similarity
- Use semantic similarity to rank translations/docs/entities
  - sim(X, Y1)
  - sim(X, Y2)
  - sim(X, Y3)
  - ……



Part II

Deep learning in statistical machine translation (SMT)

Tutorial Outline

- Part I: Background
- Part II: Deep learning in statistical machine translation (SMT)
  - Review of SMT and DNN in SMT
  - Deep semantic translation models
  - Recurrent neural language models
  - Neural network joint models
  - Neural machine translation
- Part III: Learning semantic representations
- Part IV: Natural language understanding
- Part V: Conclusion

Statistical machine translation (SMT)

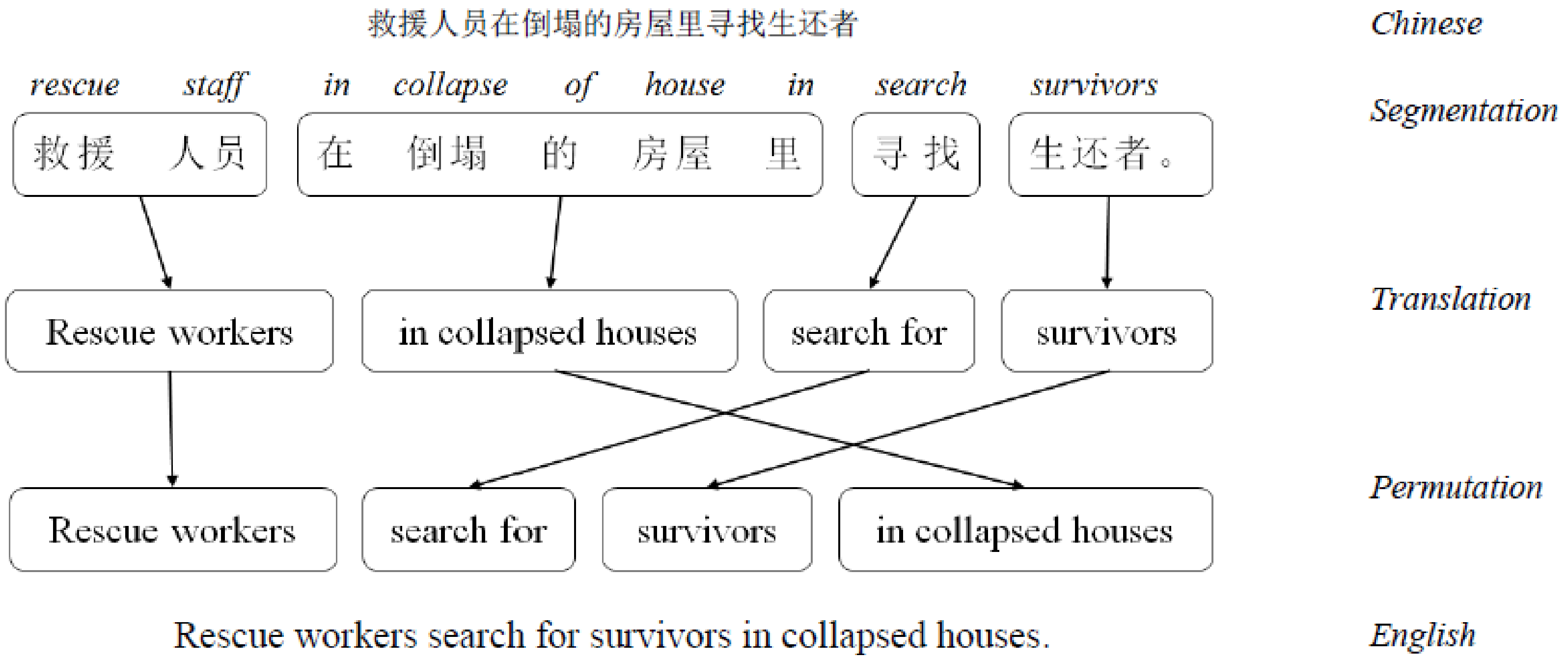
S: 救援 人员 在 倒塌的 房屋 里 寻找 生还者
T: Rescue workers search for survivors in collapsed houses

- Statistical decision: $T^* = \operatorname*{argmax}_T P(T|S)$
- Source-channel model: $T^* = \operatorname*{argmax}_T P(S|T)P(T)$
- Translation models: $P(S|T)$ and $P(T|S)$
- Language model: $P(T)$
- Log-linear model: $P(T|S) = \frac{1}{Z(S,T)} \exp \sum_i \lambda_i h_i(S,T)$
- Evaluation metric: BLEU score (higher is better)

Phrase-based SMT

救援人员在倒塌的房屋里寻找生还者 *Chinese*

Phrase-based SMT



救援人员在倒塌的房屋里寻找生还者 *Chinese*

rescue staff in collapse of house in search survivors

救援 人员 | 在 倒塌 的 房屋 里 | 寻找 | 生还者。 *Segmentation*

Rescue workers | in collapsed houses | search for | survivors *Translation*

Rescue workers | search for | survivors | in collapsed houses *Permutation*

Rescue workers search for survivors in collapsed houses. *English*

A taxonomy of neural nets in SMT [Duh 2014]

Core Engine: What is being modeled?

- Target word probability:
  - Language Model: [Schwenk et al., 2012, Vaswani et al., 2013, Niehues and Waibel, 2013, Auli and Gao, 2014]
  - LM w/ Source: [Kalchbrenner and Blunsom, 2013, Auli et al., 2013, Devlin et al., 2014, Cho et al., 2014, Bahdanau et al., 2014, Sundermeyer et al., 2014, Sutskever et al., 2014]
- Translation/Reordering probabilities under Phrase-based MT:
  - Translation: [Maskey and Zhou, 2012, Schwenk, 2012, Liu et al., 2013, Gao et al., 2014a, Lu et al., 2014, Tran et al., 2014, Wu et al., 2014a]
  - Reordering: [Li et al., 2014b]
- Tuple-based MT: [Son et al., 2012, Wu et al., 2014b, Hu et al., 2014]
- ITG Model: [Li et al., 2013, Zhang et al., 2014, Liu et al., 2014]

Related Components:

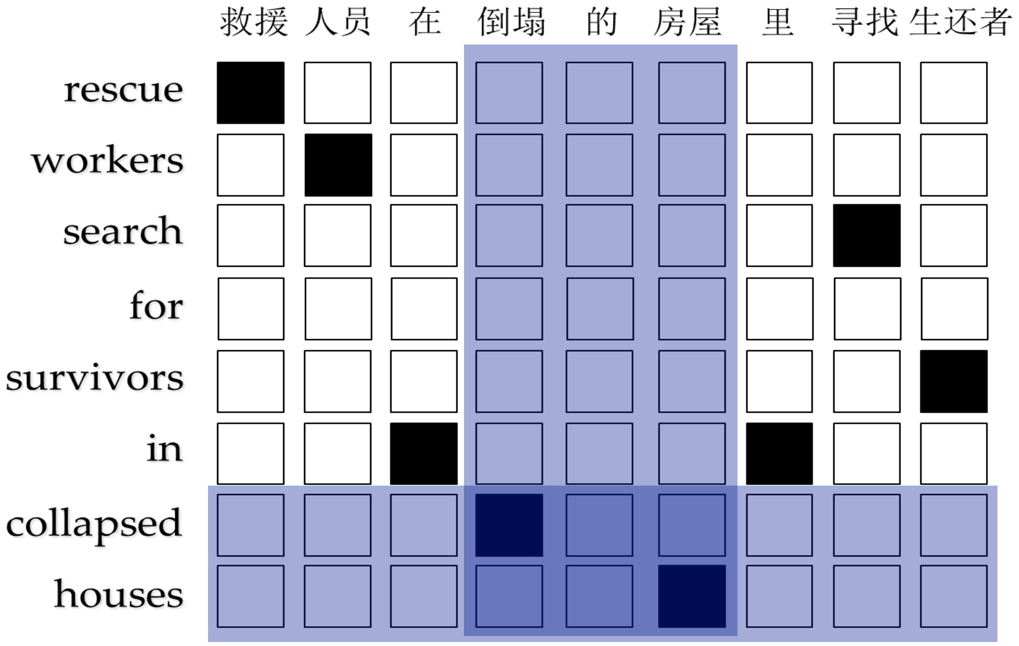
- Word Align: [Yang et al., 2013, Tamura et al., 2014, Songyot and Chiang, 2014]
- Adaptation / Topic Context: [Duh et al., 2013, Cui et al., 2014]
- Multilingual Embeddings:
  [Klementiev et al., 2012, Lauly et al., 2013, Zou et al., 2013, Kočiský et al., 2014, Faruqui and Dyer, 2014, Hermann and Blunsom, 2014, Chandar et al., 2014]

Examples of NN in phrase-based SMT

- Neural nets as components in log-linear model
  - Translation model $P(T|S)$ or $P(S|T)$: the use of DSSM [Gao+ 14]
  - Language model $P(T)$: the use of RNN [Auli+ 2013; Auli & Gao 14]
  - Joint model $P(t_i|S, t_1 \dots t_{i-1})$: FFLM + source words [Devlin+ 14]

- Neural machine translation
  - Build a single, large NN that reads a sentence and outputs a translation
  - RNN encoder-decoder [Cho+ 2014; Sutskever+ 14]
    - Long short-term memory (gated hidden units)
  - Jointly learning to align and translate [Bahdanau+ 15]

Phrase translation modeling



$(\boldsymbol{s}, \boldsymbol{t})$

(救援, rescue)
(人员, workers)
(在, in)
(倒塌, collapsed)
(房屋, house)
(里, in)
(寻找, search)
(生还者, survivors)
(救援 人员, rescue workers)
(在 倒塌, in collapsed)
(倒塌 的, collapsed)
(的 房屋, house)
(寻找, search for)
(寻找 生还者, search for survivors)
(生还者, for survivors)
(倒塌 的 房屋, collapsed house)

MLE: $P(\boldsymbol{t}|\boldsymbol{s}) = \frac{N(\boldsymbol{s},\boldsymbol{t})}{\sum_{\boldsymbol{t}'} N(\boldsymbol{s},\boldsymbol{t}')}$

Simple, but suffers the data sparseness problem

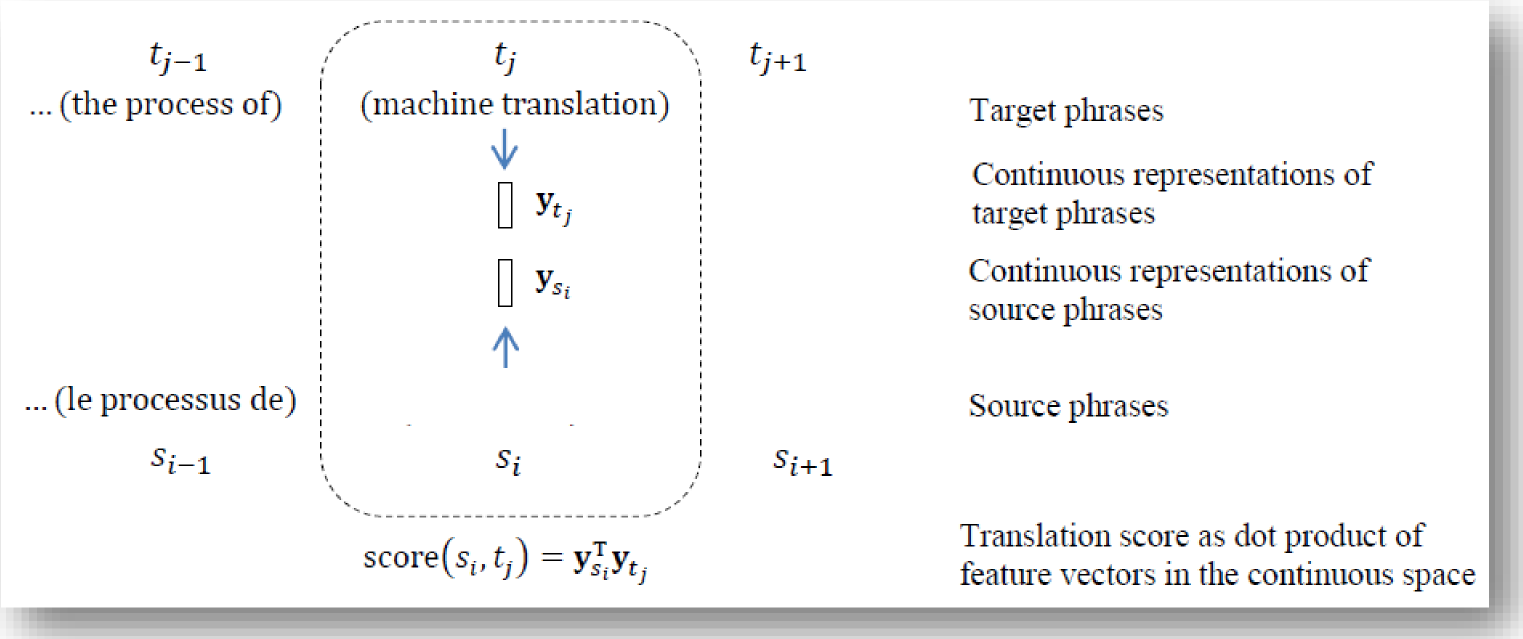
Deep Semantic Similarity Model (DSSM)

[Huang+ 13; Gao+ 14a; Gao+ 14b; Shen+ 14, Yih+ 15]

- Compute semantic similarity btw text strings X and Y
  - Map X and Y to feature vectors in a latent semantic space via deep neural net
  - Compute the cosine similarity between the feature vectors
  - Also called "Deep Structured Similarity Model" in [Huang+ 13]
- DSSM for NLP tasks

Tasks	**X**	**Y**
Machine translation	***Text in language A***	***Translation in language B***
Web search	*Search query*	*Web document*
Image captioning	*Image*	*Caption*
Question Answering	*Question*	*Answer*

DSSM for phrase translation modeling [Gao+ 14a]



- Two neural nets (one for source side, one for target side)
  - Input: bag-of-words representation of source/target phrase
  - Output: vector $\mathbf{y}_s$ for source phrase, $\mathbf{y}_t$ for target phrase
- Phrase translation score = dot product of these vectors
  - $\mathrm{score}(s, t) \equiv \mathrm{sim}_{\boldsymbol{\theta}}(\mathbf{x}_s, \mathbf{x}_t) = \mathbf{y}_s^{\mathrm{T}} \mathbf{y}_t$
- Alleviate data sparsity, enable complex scoring functions, etc.

Model training procedure

- Generate N-best lists using a baseline SMT system
  - Oracle BLEU in N-best is much better than 1-best
- Optimize neural net parameters $\boldsymbol{\theta}$ on the N-best lists of training data
  - Expected BLEU objective: $\mathrm{xBleu}(\boldsymbol{\theta}) = \sum_{T \in \mathrm{GEN}(S_i)} P(T|S_i)\mathrm{sBleu}(T_i, T)$
  - Update $\boldsymbol{\theta}$ with SGD: $\boldsymbol{\theta}^{\boldsymbol{new}} = \boldsymbol{\theta} - \eta \frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}},$
  - where $\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{(s,t)} \frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \mathrm{sim}_{\boldsymbol{\theta}}(\mathbf{x}_s, \mathbf{x}_t)} \frac{\partial \mathrm{sim}_{\boldsymbol{\theta}}(\mathbf{x}_s, \mathbf{x}_t)}{\partial \boldsymbol{\theta}}$
- Incorporate DSSM as a feature in log-linear model
  - Feature weight is optimized using MERT on development data.
  - No decoder modification
- Loop if desired

[Gao, He, Yih, Deng 2014]

N-gram language modeling

- Word n-gram model (e.g., n = 3)
  - A word depends only on n-1 preceding words
  - $P(w_1 w_2 \dots w_n) = P(w_1) P(w_2|w1) \prod_{i=2 \dots n} P(w_i|w_{i-2} w_{i-1})$
  - Cannot capture long-distance dependency

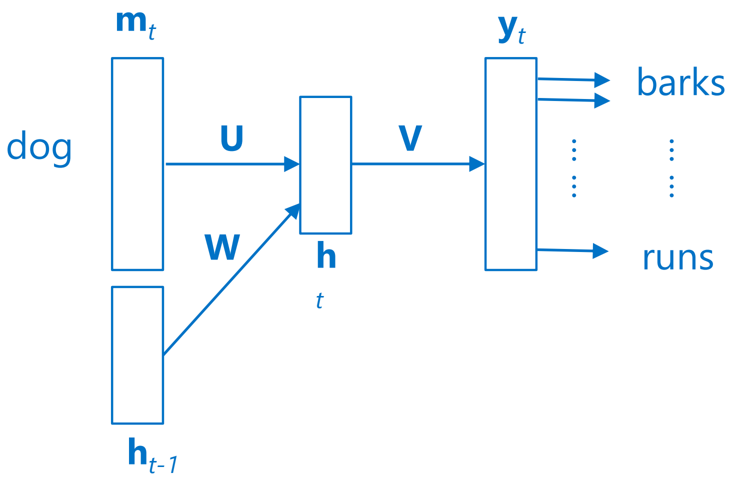
the dog of our neighbor barks

- Problem of using long history
  - Rare events: unreliable probability estimates

model	**# parameters**
unigram $P(w_1)$	20,000
bigram $P(w_2|w_1)$	400M
trigram $P(w_3|w_1 w_2)$	8 x 10^{12}
4-gram $P(w_4|w_1 w_2 w_3)$	1.6 x 10^{17}

[Manning & Schütze 99]

Recurrent neural net for language modeling



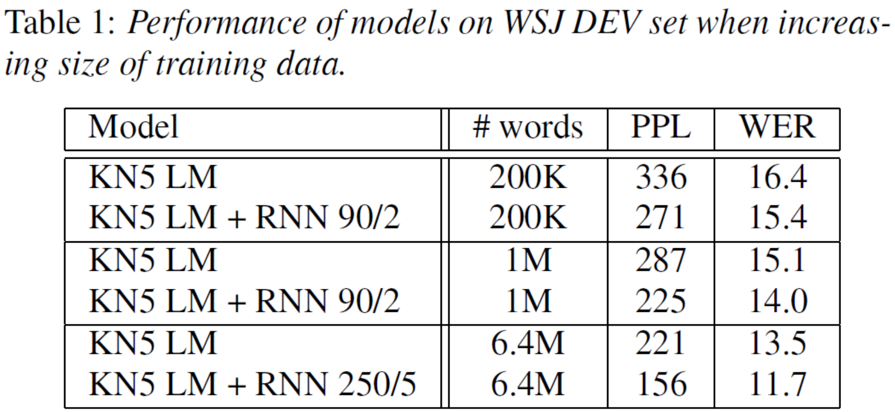


Table 1: *Performance of models on WSJ DEV set when increasing size of training data.*

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

[Mikolov+ 11]

m_t: input one-hot vector at time step t
h_t: encodes the history of all words up to time step t
y_t: distribution of output words at time step t

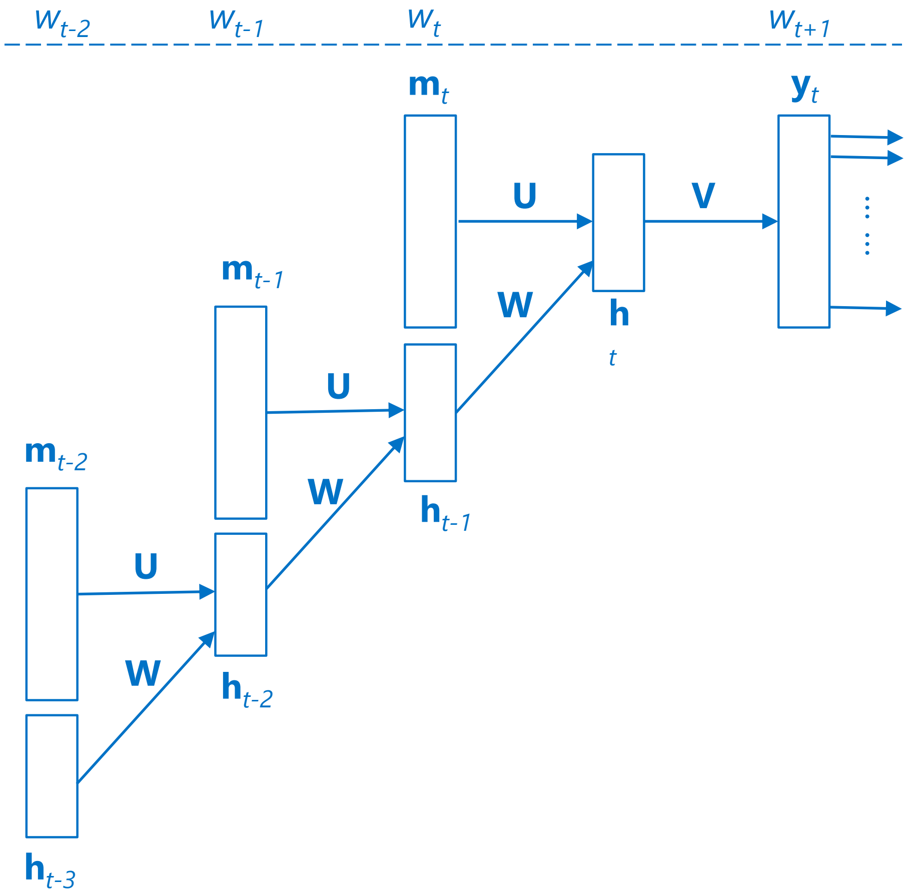
$$\mathbf{z}_t = \mathbf{U}\mathbf{m}_t + \mathbf{W}\mathbf{h}_{t-1}$$
$$\mathbf{h}_t = \sigma(\mathbf{z}_t)$$
$$\mathbf{y}_t = g(\mathbf{V}\mathbf{h}_t)$$

where

$$\sigma(z) = \frac{1}{1+\exp(-z)}, \quad g(z_m) = \frac{\exp(z_m)}{\sum_k \exp(z_k)}$$

$g(.)$ is called the *softmax* function

RNN unfolds into a DNN over time



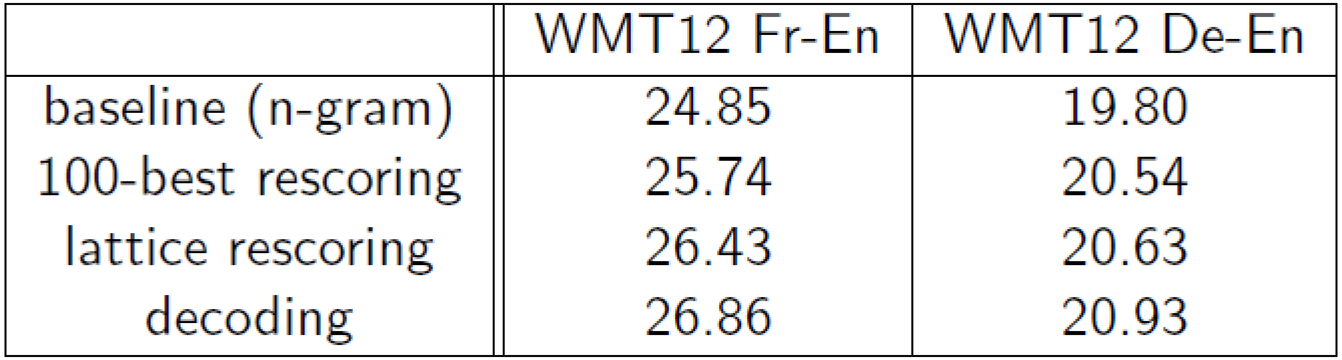
$$\mathbf{z}_t = \mathbf{U}\mathbf{m}_t + \mathbf{W}\mathbf{h}_{t-1}$$
$$\mathbf{h}_t = \sigma(\mathbf{z}_t)$$
$$\mathbf{y}_t = g(\mathbf{V}\mathbf{h}_t)$$

where

$$\sigma(z) = \frac{1}{1+\exp(-z)}, \quad g(z_m) = \frac{\exp(z_m)}{\sum_k \exp(z_k)}$$

RNN LM decoder integration [Auli & Gao 14]

- RNN LMs require history going back to start-of-sentence. Harder to do dynamic programming.
- To score new words, each decoder state needs to maintain *h*. For recombination, merge hypotheses by traditional n-gram context and the best *h*



	WMT12 Fr-En	WMT12 De-En
baseline (n-gram)	24.85	19.80
100-best rescoring	25.74	20.54
lattice rescoring	26.43	20.63
decoding	26.86	20.93

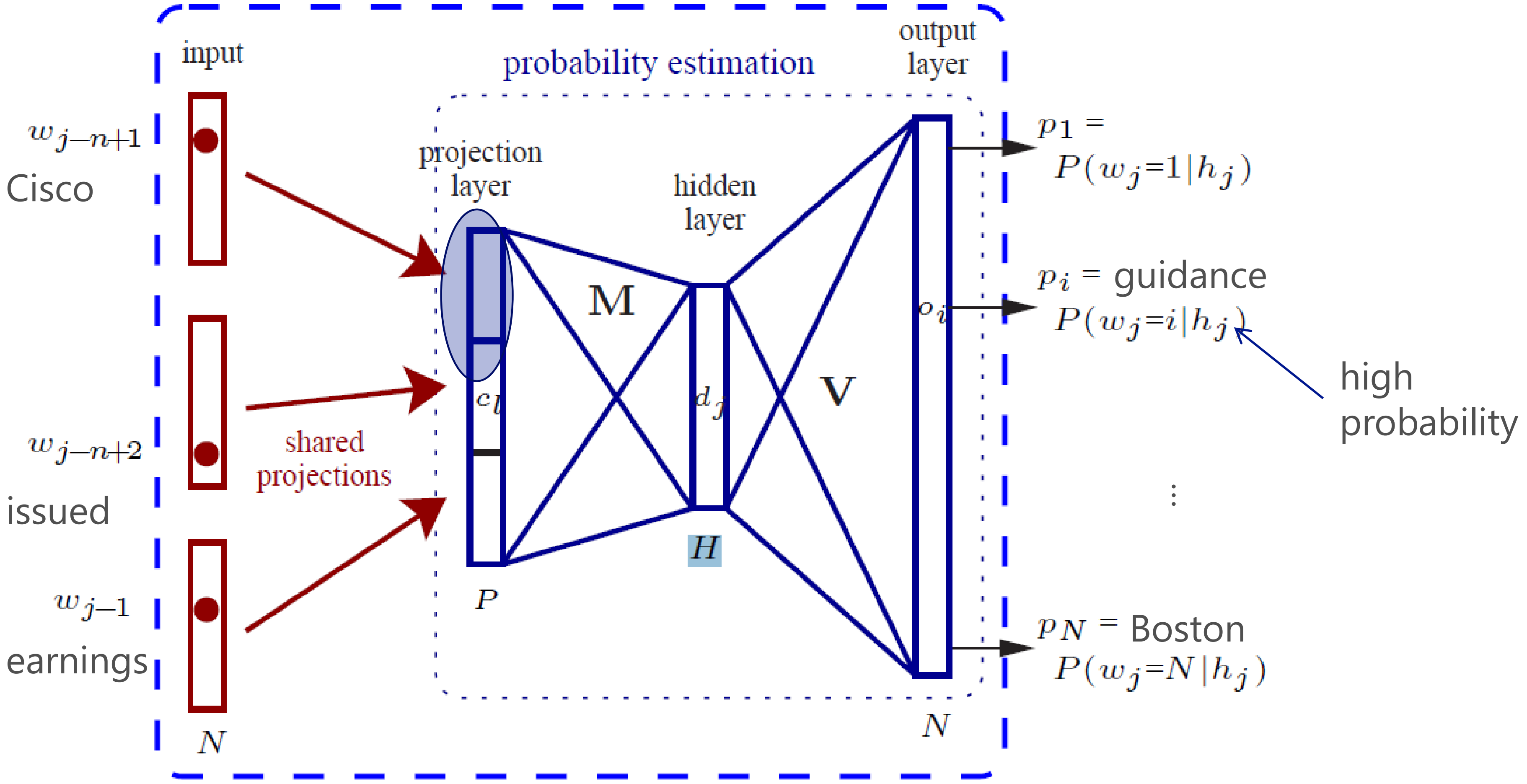
Joint model: language model with source

- $P(t_i|t_{i-2}t_{i-1}, S)$

- How to model $\boldsymbol{S}$?
  - Entire source sentence or aligned source words
  - S as a word sequence, bag of words, or vector representation
  - How to learn the vector representation of S?

- Neural network joint models based on
  - RNN language model [Auli+ 13]
  - Feedforward neural language model [Devlin+ 14]

Feed-forward neural language model [Bengio+ 03]



Joint model of [Devlin+ 14]

P(the | get, will, i, 就, 取, 钱, 给, 了)

- Extend feed-forward LM to include window around aligned source words.
  - **Heuristic: if align to multiple source words, choose middle; if unaligned, inherit alignment from closest target word**
- Train on bitext with alignment; optimize target likelihood.

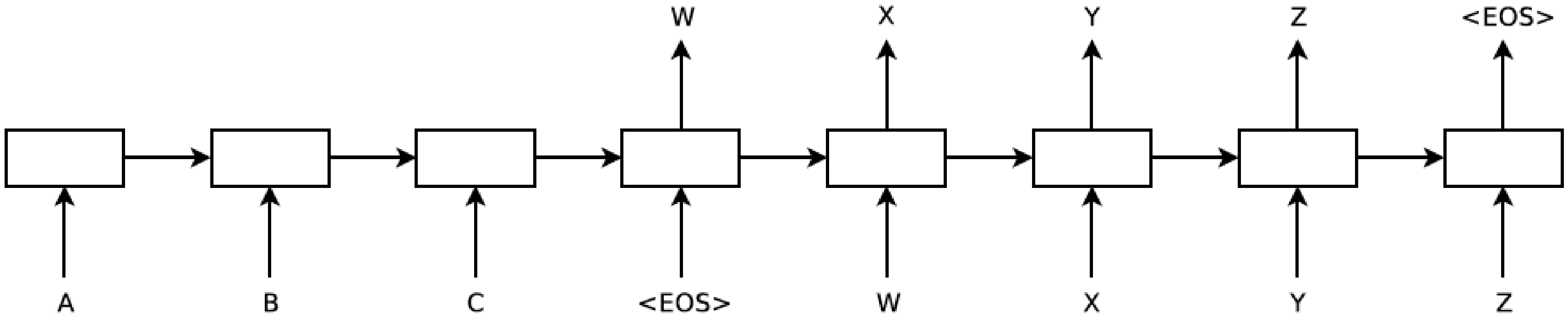
Neural machine translation

[Sutskever+ 14; Cho+ 14; Bahdanau+ 15]

- Build a single, large NN that reads a sentence and outputs a translation
  - Unlike phrase-based system that consists of many component models
- Encoder-decoder based approach
  - An encoder RNN reads and encodes a source sentence into a fixed-length vector
  - A decoder RNN outputs a variable-length translation from the encoded vector
  - Encoder-decoder RNNs are jointly learned on bitext, optimize target likelihood

Encoder-decoder model of [Sutskever+ 2014]

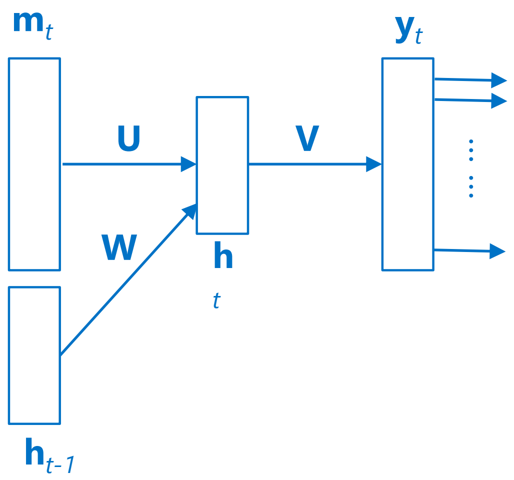
- "A B C" is source sentence; "W X Y Z" is target sentence

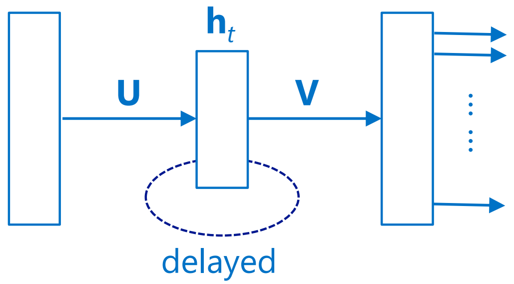


- Treat MT as general sequence-to-sequence transduction
  - **Read source; accumulate hidden state; generate target**
  - **<EOS> token stops the recurrent process**
  - **In practice, read source sentence in reverse leads to better MT results**
- Train on bitext; optimize target likelihood using SGD

Potentials and difficulties of RNN

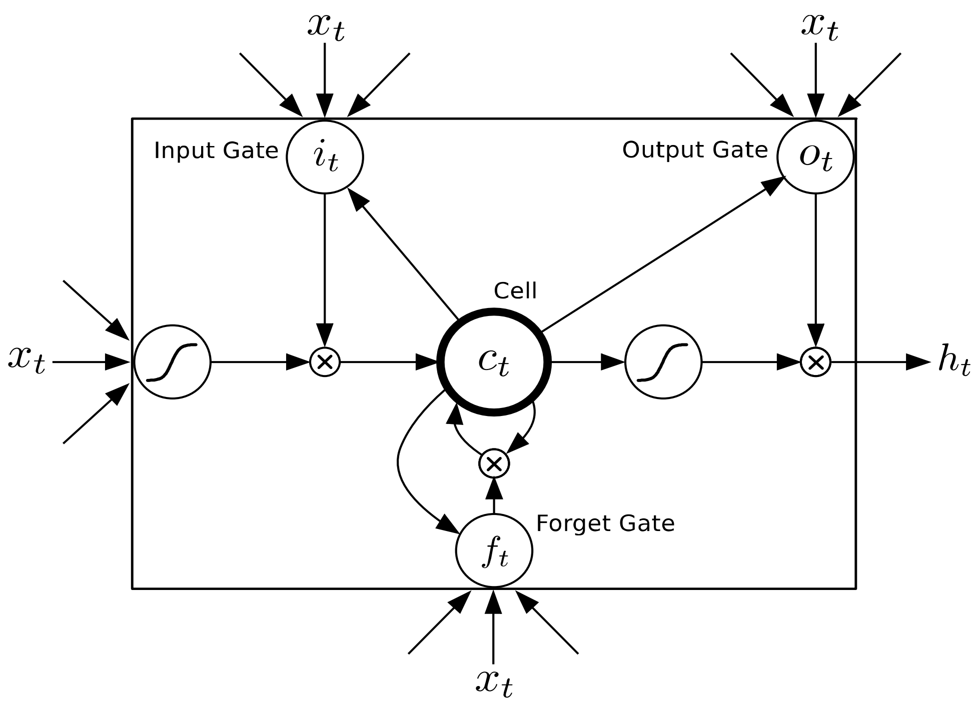
- In theory, RNN can “store” in $\boldsymbol{h}$ all information about past inputs
- But in practice, standard RNN cannot capture very long distance dependency
  - Vanishing/exploding gradient problem in backpropagation
  - Not robust to noise
- Solution: long short-term memory (LSTM)





A long short-term memory cell

[Hochreiter & Schmidhuber 97; Graves+ 13]



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

Information flow in an LSTM unit of the RNN, with both diagrammatic and mathematical descriptions. W’s are weight matrices, not shown but can easily be inferred in the diagram (Graves et al., 2013).

A 2-gate memory cell [Cho+ 14]

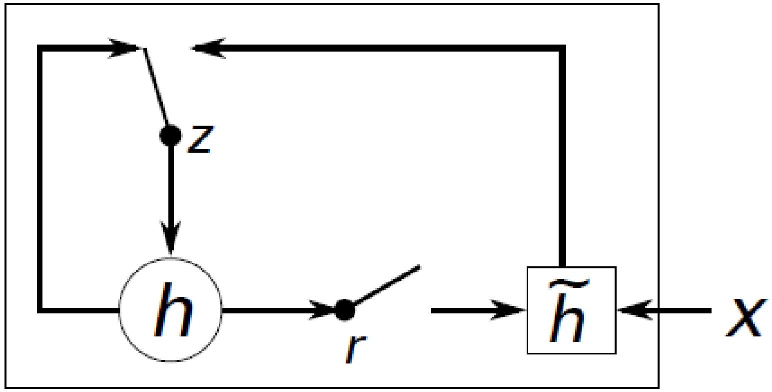


Figure 2: An illustration of the proposed hidden activation function. The update gate z selects whether the hidden state is to be updated with a new hidden state $\tilde{h}$. The reset gate r decides whether the previous hidden state is ignored. See

$$r_j = \sigma\left(\left[\mathbf{W}_r\mathbf{x}\right]_j + \left[\mathbf{U}_r\mathbf{h}_{\langle t-1\rangle}\right]_j\right)$$

$$z_j = \sigma\left(\left[\mathbf{W}_z\mathbf{x}\right]_j + \left[\mathbf{U}_z\mathbf{h}_{\langle t-1\rangle}\right]_j\right)$$

$$\tilde{h}_j^{\langle t\rangle} = \phi\left(\left[\mathbf{W}\mathbf{x}\right]_j + \left[\mathbf{U}\left(\mathbf{r}\odot\mathbf{h}_{\langle t-1\rangle}\right)\right]_j\right)$$

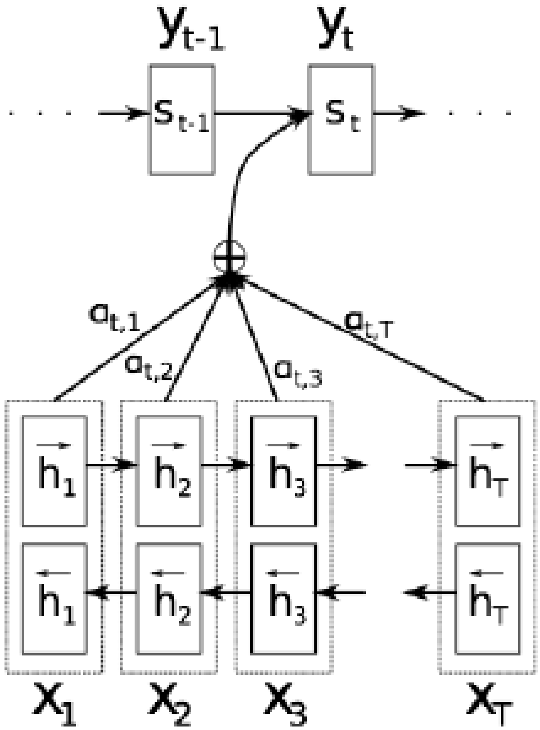
$$h_j^{\langle t\rangle} = z_j h_j^{\langle t-1\rangle} + (1-z_j)\tilde{h}_j^{\langle t\rangle}$$

Joint learning to align and translate

- Issue with encoder-decoder model for SMT
  - Compressing a source sentence into a fixed-length vector makes it difficult for RNN to cope with long sentences.
- Attention model of [Bahdanan+ 15]
  - Encodes the input sentence into a sequence of vectors and choose a subset of these vectors adaptively while decoding
  - An idea similar to that of [Devlin+ 14]

Attention model of [Bahdanan+ 15]

- Encoder:
  - bidirectional RNN to encode each word and its context
- Decoder:
  - Searches for a set of source words that are most relevant to the target word to be predicted.
  - Predicts a target word based on the context vectors associated with these source words and all the previous generated target words.
- Close to state-of-the-art performance
  - **Better at translating long sentences**



Interim summary

- A brief history of DNN

- DNNs in statistical machine translation
  - Feed Forward Neural Networks
  - Recurrent Neural Networks (RNN)
  - Long Short-Term Memory (LSTM)
  - Deep Semantic Similarity Model (DSSM)

Tutorial Outline

- Part I: Background
- Part II: Deep learning in statistical machine translation (SMT)
- Part III: Learning semantic representations
  - Sentence to vector
  - The deep semantic similarity model (DSSM)
  - Convolutional & Recurrent DSSM
  - Applications to IR and contextual entity ranking
  - Multimodal semantic learning for image captioning
- Part IV: Natural language understanding
- Part V: Conclusion

Part III

Deep Learning for Semantic Representations

Deep Learning for Semantic Representations

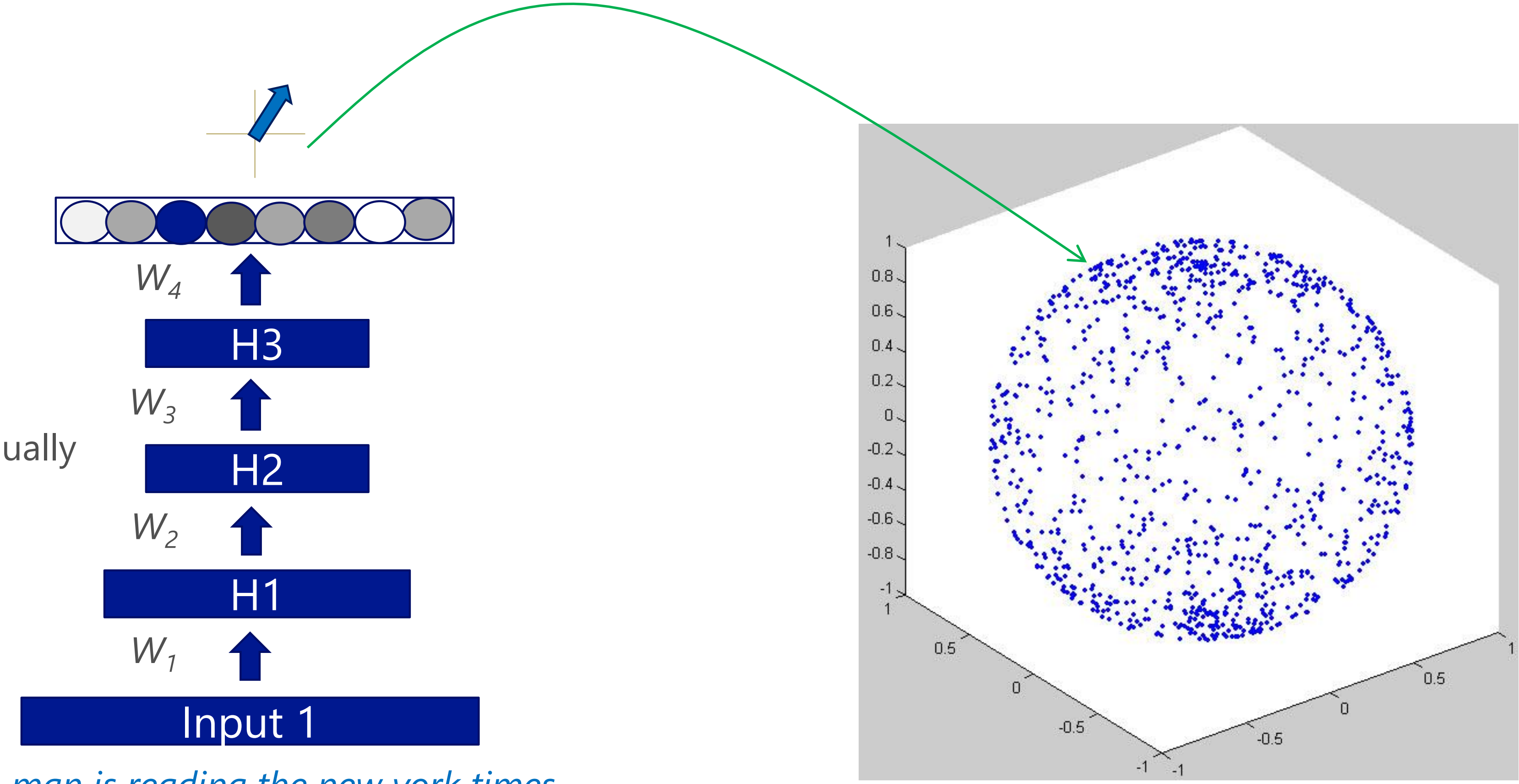
- Sentence to vector
- The deep semantic similarity model (DSSM)
- Convolutional & Recurrent DSSM
- Applications to IR and contextual entity ranking
- Multimodal semantic learning for image captioning

Learning semantic representation
e.g., from a raw sentence to an abstract semantic vector (Sent2Vec)

Abstract representation
in the semantic space

each non-linear layer gradually
extracts deeper invariance

Raw text, e.g., a
sequence of words



a man is reading the new york times

Sent2Vec is crucial in many NLP tasks

Tasks	***Source***	***Target***
Web search	*search query*	*web documents*
Ad selection	*search query*	*ad keywords*
Contextual entity ranking	*mention (highlighted)*	*entities*
Online recommendation	*doc in reading*	*interesting things / other docs*
Machine translation	*phrases in language S*	*phrases in language T*
Knowledge-base construction	*entity*	*entity*
Question answering	*pattern | mention*	*relation | entity*
Personalized recommendation	*user*	*app, movie, etc.*
Image search	*query*	*image*
Image captioning	*image*	*text*
…		

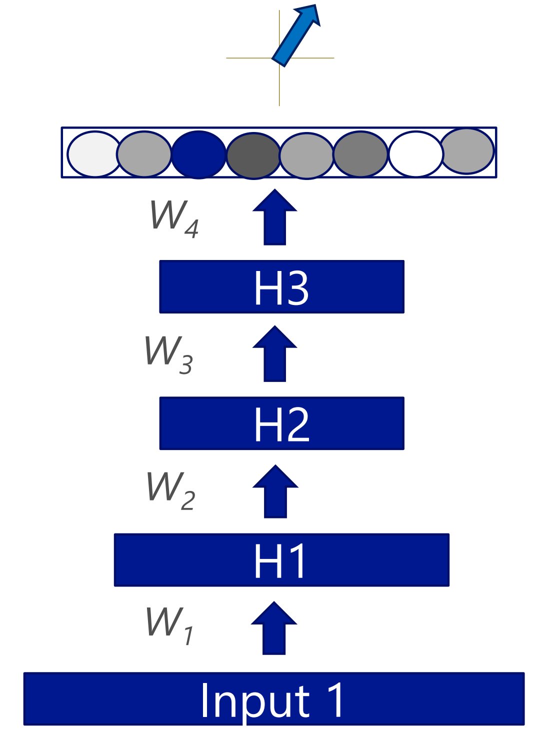
Sent2Vec is crucial in many NLP tasks

Tasks	***Source***	***Target***
Web search	***search query***	***web documents***
Ad selection	*search query*	*ad keywords*
Contextual entity ranking	***mention (highlighted)***	***entities***
Online recommendation	*doc in reading*	*interesting things / other docs*
Machine translation	*phrases in language S*	*phrases in language T*
Knowledge-base construction	*entity*	*entity*
Question answering	*pattern | mention*	*relation | entity*
Personalized recommendation	*user*	*app, movie, etc.*
Image search	*query*	*image*
Image captioning	***image***	***text caption***
…		

Deep Learning for Semantic Representations

- Sentence to vector
- The deep semantic similarity model (DSSM)
- Convolutional & Recurrent DSSM
- Applications to IR and contextual entity ranking
- Multimodal semantic learning for image captioning

The supervision problem:



a man is reading the new york times

However

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

Fortunately

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

Deep Structured Semantic Model

Deep Structured Semantic Model/Deep Semantic Similarity Model (**DSSM**)

  Sentence to vector!

The DSSM is built upon sub-word units for scalability and generalizability

  e.g., letter-trigrams, phones, roots/morphs, instead of *words*

The DSSM is trained by optimizing an similarity-driven objective

  projecting semantically similar sentences to vectors close to each other

  projecting semantically different sentences to vectors far apart

The DSSM is learned from various signals, with or without human labeling effort

  semantically-similar text pairs

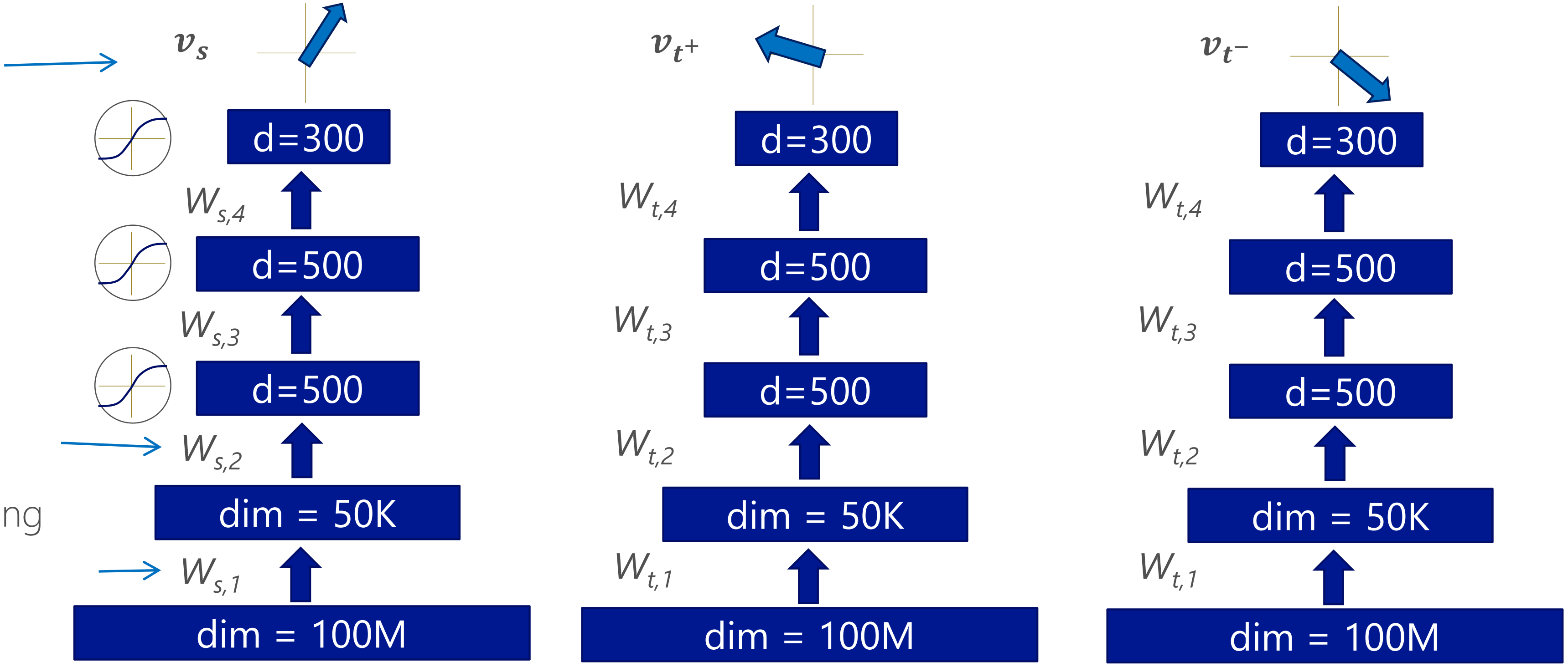
  e.g., user behavior log data, contextual text

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

DSSM: a similarity-driven Sent2Vec model

Initialization:

Neural networks are initialized with random weights



Semantic vector → v_s | v_{t^+} | v_{t^-}

d=300 | d=300 | d=300

$W_{s,4}$ | $W_{t,4}$ | $W_{t,4}$

d=500 | d=500 | d=500

$W_{s,3}$ | $W_{t,3}$ | $W_{t,3}$

d=500 | d=500 | d=500

Letter-trigram embedding matrix → $W_{s,2}$ | $W_{t,2}$ | $W_{t,2}$

dim = 50K | dim = 50K | dim = 50K

Letter-trigram encoding matrix (fixed) → $W_{s,1}$ | $W_{t,1}$ | $W_{t,1}$

Bag-of-words vector: dim = 100M | dim = 100M | dim = 100M

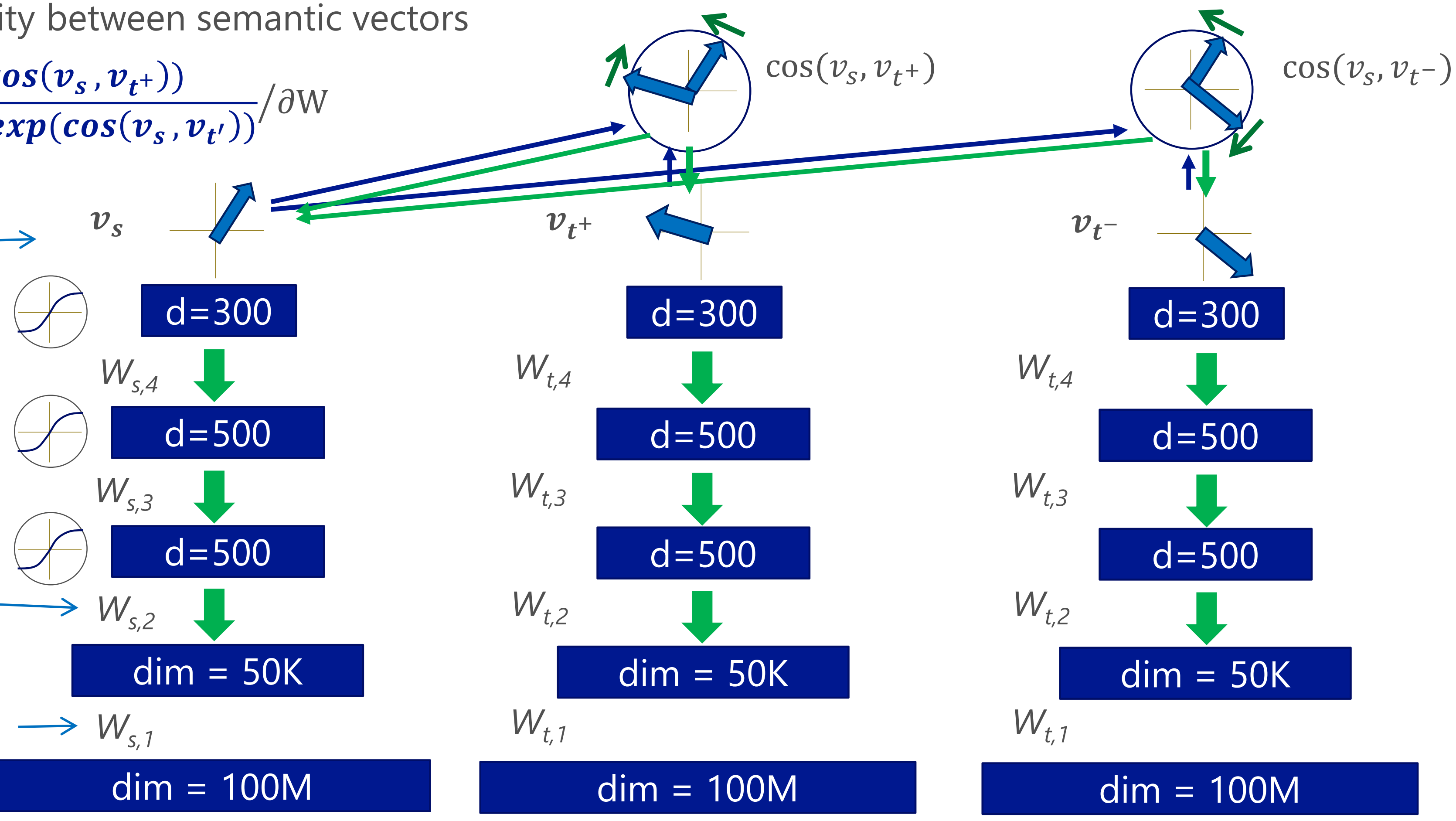
Input word/phrase: *s*: “**racing car**” | *t^+*: “**formula one**” | *t^-*: “**racing to me**”

DSSM: a similarity-driven Sent2Vec model

Training:

Compute Cosine similarity between semantic vectors

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



Semantic vector

Letter-trigram embedding matrix

Letter-trigram encoding matrix (fixed)

Bag-of-words vector

Input word/phrase

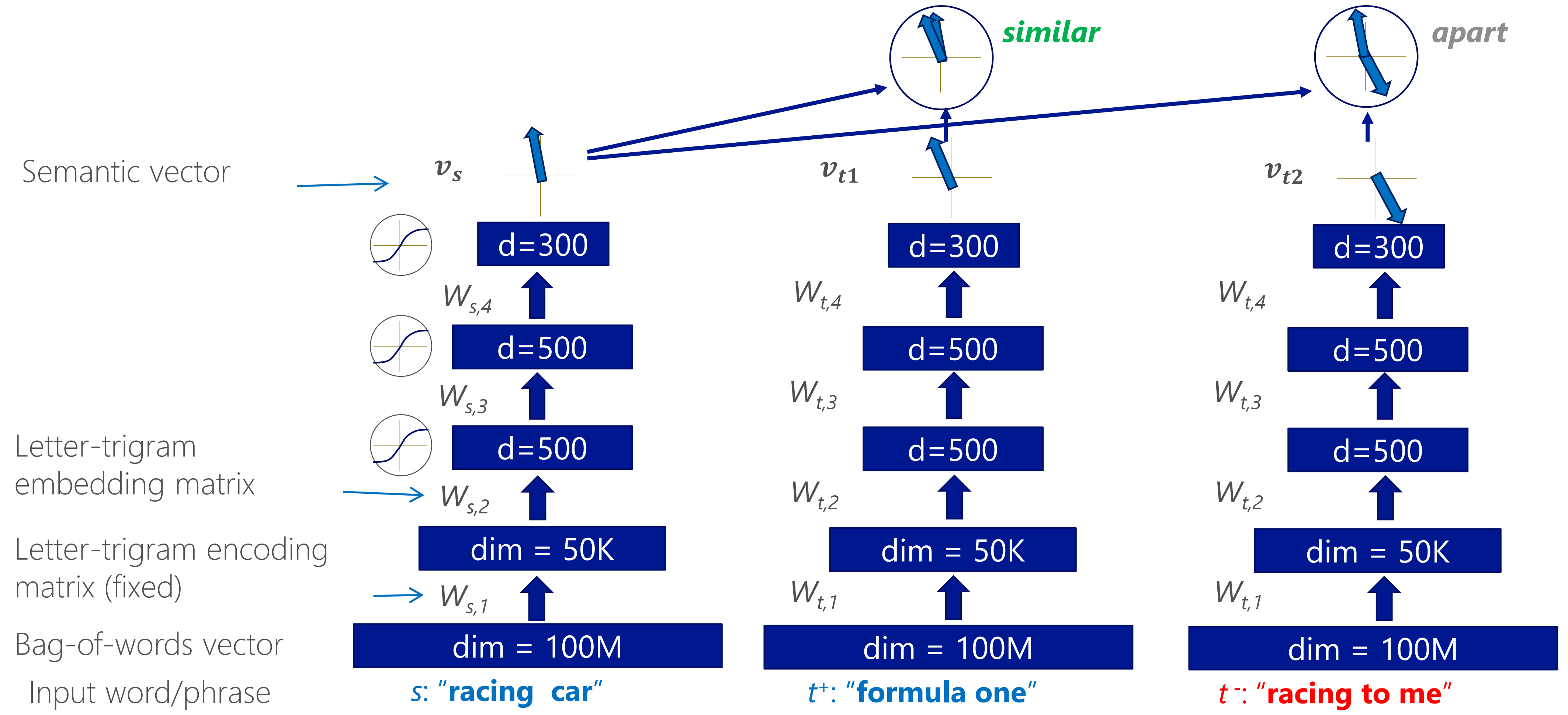
s: "**racing car**"

t⁺: "**formula one**"

t⁻: "**racing to me**"

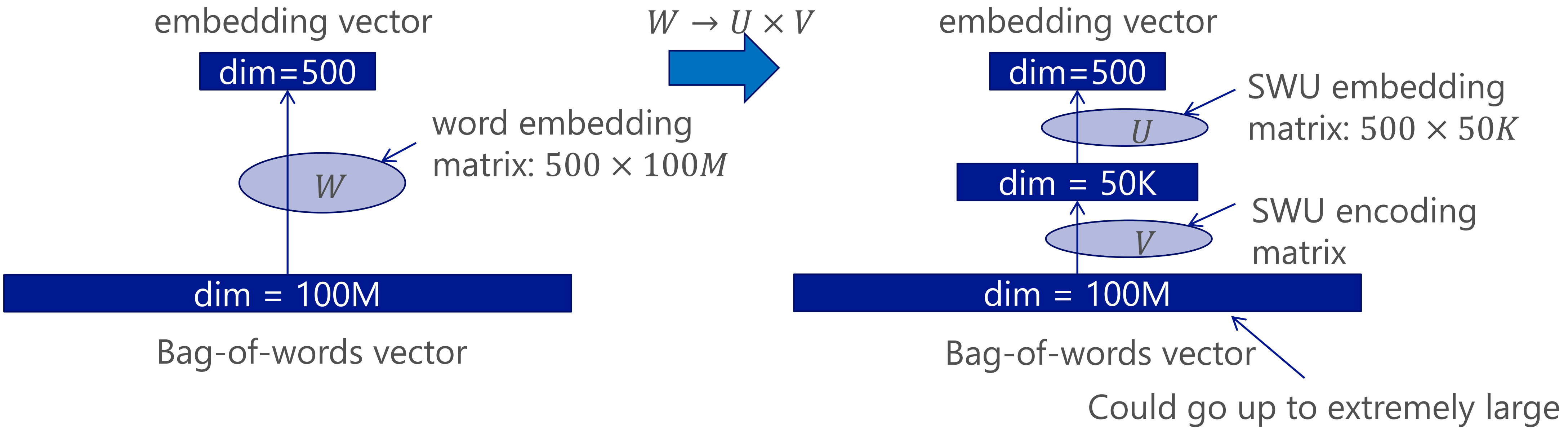
DSSM: a similarity-driven Sent2Vec model

Runtime:



DSSM: built on top of sub-word units

Decompose *any* word into sub-word units (SWU)



Preferable for large scale NL tasks

- Arbitrary size of vocabulary (*scalability*)
- Misspellings, word fragments, new words, etc. (*generalizability*)

Options:

- Letters, context-dept letters, positioned-phones, context-dept phones, positioned-roots/morphs, context-dept morphs

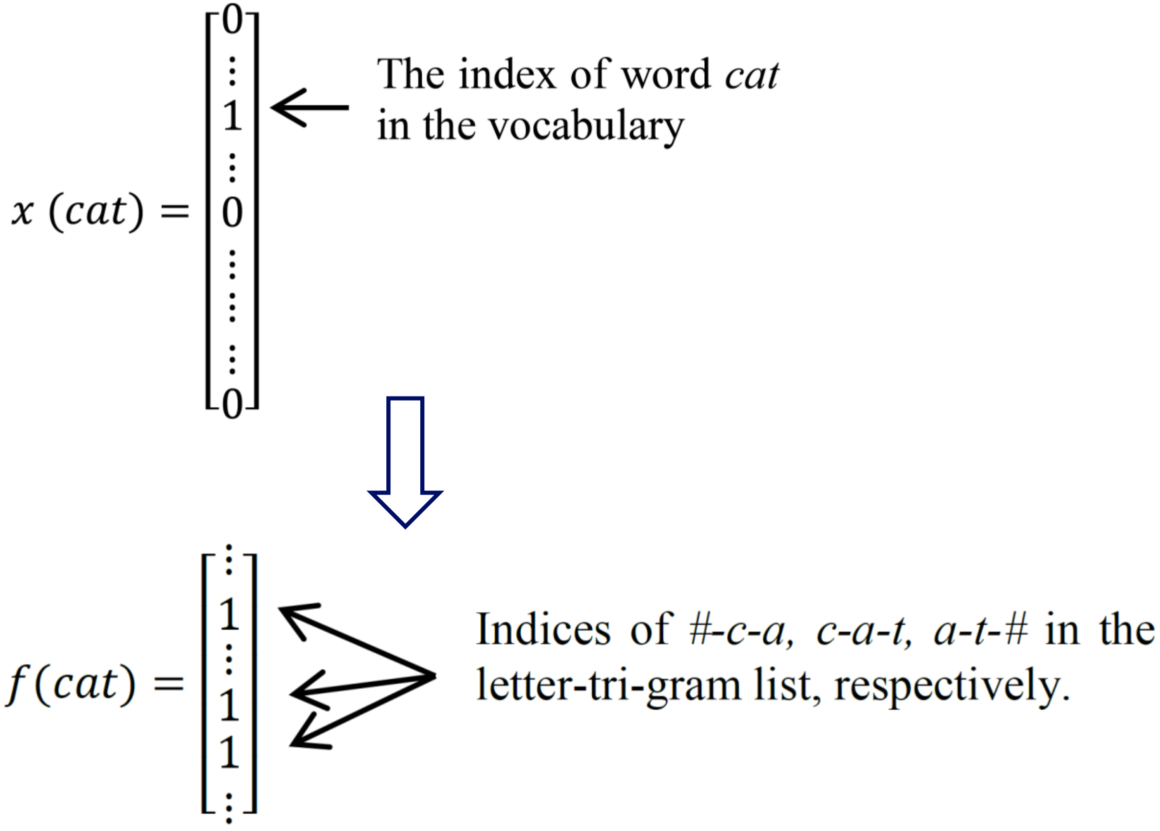
Or

Random projection (random basis unit)

Multi-hashing approach to word input representation

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.



$$x(cat) = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}$$

The index of word *cat* in the vocabulary

$$f(cat) = \begin{bmatrix} \vdots \\ 1 \\ \vdots \\ 1 \\ 1 \\ \vdots \end{bmatrix}$$

Indices of *#-c-a*, *c-a-t*, *a-t-#* in the letter-tri-gram list, respectively.

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

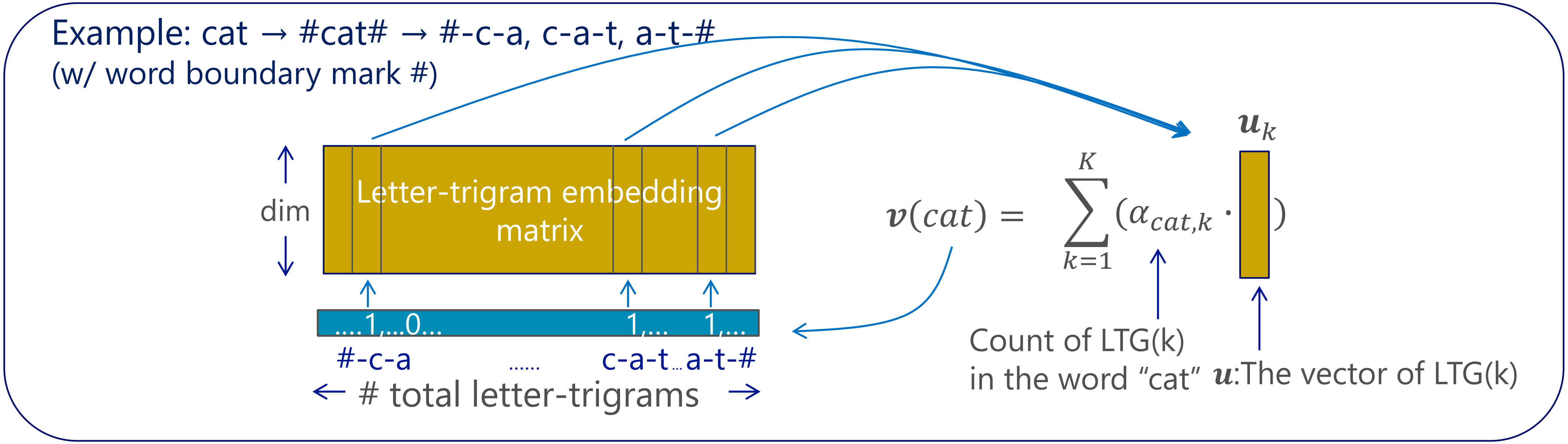
Vocabulary size	Unique letter-tg observed in voc	Number of Collisions
40K	10306	2 (0.005%)
500K	30621	22 (0.004%)

Learning sub-word unit embedding 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 #)

$$\boldsymbol{v}(cat) = \sum_{k=1}^{K} (\alpha_{cat,k} \cdot \boldsymbol{u}_k)$$

Count of LTG(k) in the word “cat”

$\boldsymbol{u}$:The vector of LTG(k)

Training objectives

Objective: cosine similarity based loss

Using web search as an example:

- a query $\boldsymbol{q}$ and a list of docs $\boldsymbol{D} = \{d^+, d_1^-, \ldots d_K^-\}$
  - d^+ positive doc; $d_1^-, \ldots d_K^-$ are negative docs to q (e.g., sampled from not clicked docs)
- Objective: the posterior probability of the clicked doc given the query

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

e.g., $v_{\boldsymbol{\theta}}(q) = \sigma(W_{s,4} \times \sigma(W_{s,3} \times \sigma(W_{s,2} \times ltg(q)))$

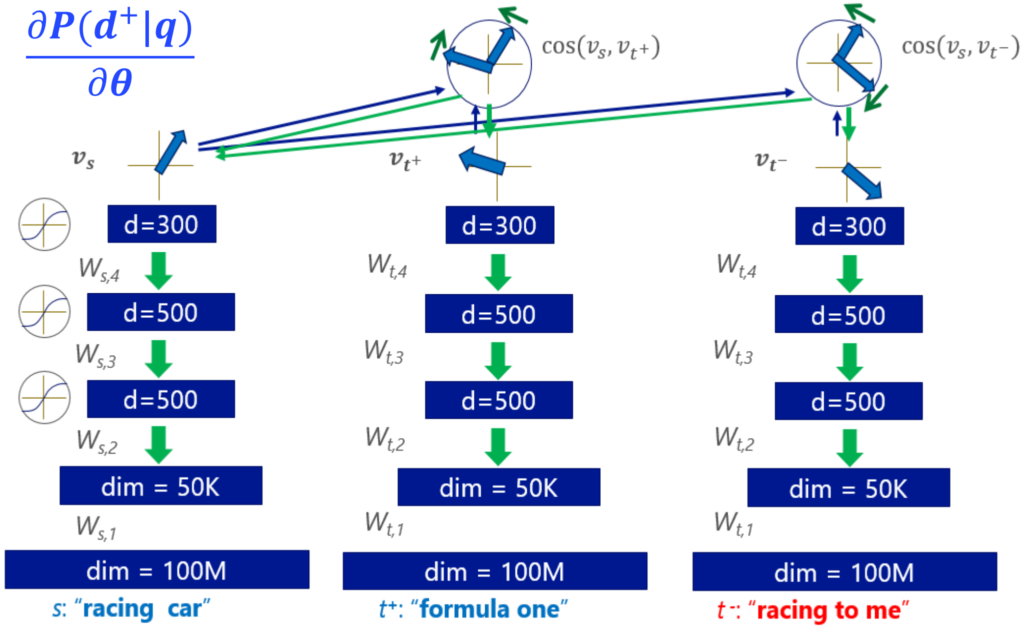
$v_{\boldsymbol{\theta}}(d) = \sigma(W_{t,4} \times \sigma(W_{t,3} \times \sigma(W_{t,2} \times ltg(d)))$

where $\theta = \{W_{s,2\sim4}, W_{t,2\sim4}\}, \sigma(\,)$ is a tanh function.

Optimization

- Optimize $\boldsymbol{\theta}$ to maximize $P(d^+|q)$.
- $\boldsymbol{\theta}$ is randomly initialized
- SGD training on GPUs

  e.g. NVidia K40



Please refer to the full version of the paper for detailed derivation.

[Huang, He, Gao, Deng, Acero, Heck, 2013]

Mine semantically-similar text pairs from Search Logs


how to deal with stuffy nose?

stuffy nose treatment

cold home remedies

Best Home Remedies for Cold and Flu

Wind Heat External Pathogens

By: Catherine Browne, L.Ac., MH, Dipl. Ac.

In Chinese medicine, colds and flu's are delineated into several different energetic classifications. Here we will outline the different types of cold and flu viruses that you will likely encounter, and

QUERY (Q)	Clicked Doc Title (T)
how to deal with stuffy nose	best home remedies for cold and flu
stuffy nose treatment	best home remedies for cold and flu
cold home remedies	best home remedies for cold and flu
… …	… …
go israel	forums goisrael community
skate at wholesale at pr	wholesale skates southeastern skate supply
breastfeeding nursing blister baby	clogged milk ducts babycenter
thank you teacher song	lyrics for teaching educational children s music
immigration canada lacolle	cbsa office detailed information

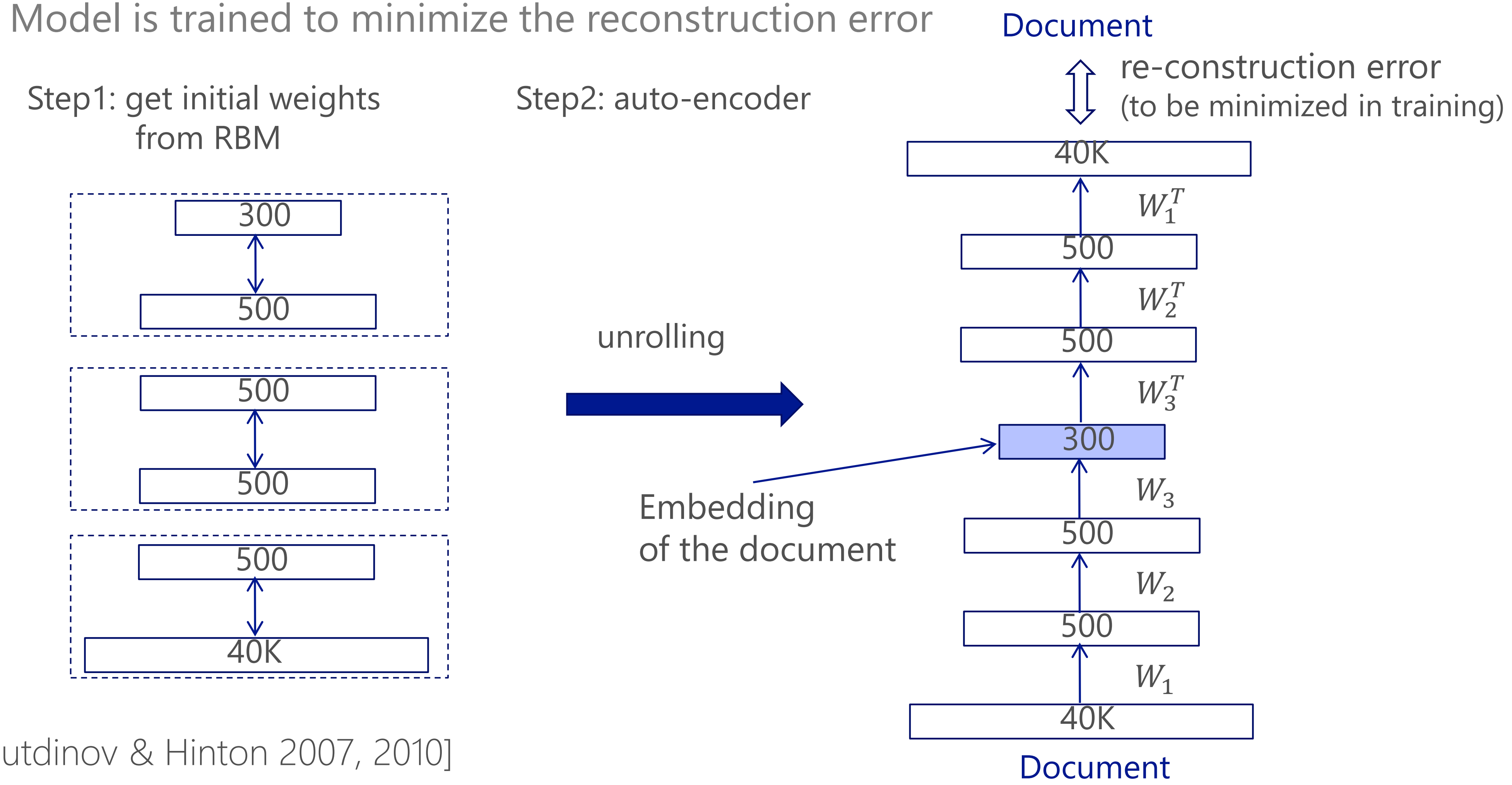
[Gao, He, Nie, CIKM2010]

Semantic Hashing

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



Step1: get initial weights from RBM

300
500

500
500

500
40K

Step2: auto-encoder

unrolling

Embedding of the document

Document

re-construction error (to be minimized in training)

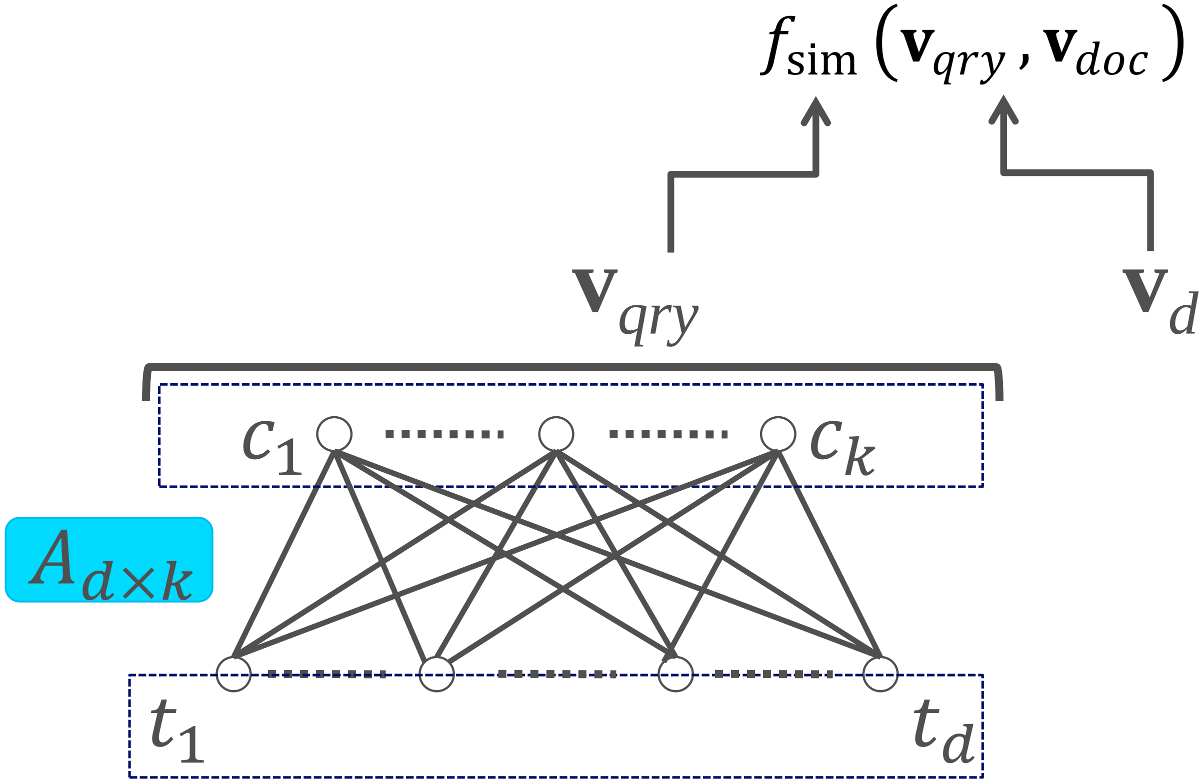
40K
W_1^T
500
W_2^T
500
W_3^T
300
W_3
500
W_2
500
W_1
40K

Document

[Salakhutdinov & Hinton 2007, 2010]

S2Net

- Model form is the same as LSA/PCA
- Learning the projection matrix discriminatively



$$V_{qry} = A^T F_{qry}$$

[Yih, Toutanova, Platt, Meek, 2011]

DSSM: Web Search

- Training data:
  - 100M query/clicked-doc-title pairs from search log
- Test set:
  - 16,510 English queries sampled from 1-yr. log
  - 5-level relevance label for each query-doc pair
  - Evaluated by NDCG
- Baselines
  - Lexicon matching models: BM25
  - Topic model: PLSA

Results on a document retrieval task

Docs are ranked by the cosine similarity between query vector and doc vector

	NDCG@1
BM25	30.8
LSA (Deerwester et al., 1990)	29.8
PLSA (Hofmann 1999)	29.5
Auto-Encoder (Hinton et al., 2010)	31.0
DPM (w/ S2Net (Yih et al., 2011))	32.9
Word Translation Model (Gao et al, 2010)	33.2
Bilingual Topic Model (Gao et al., 2011)	33.7
DSSM	**36.2**

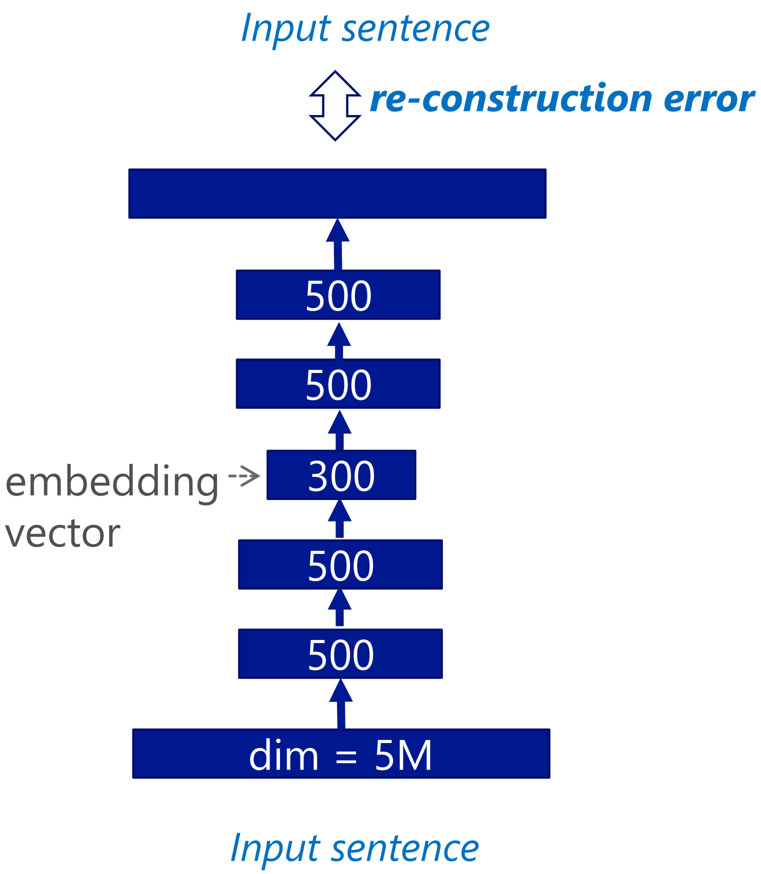
The DSSM improves 5~7 pt NDCG over shallow models

The higher the NDCG score the better, 1% NDCG difference is statistically significant.

- The DSSM learns superior semantic embedding
- Letter-trigram + the DSSM gives superior results

Reflection: from Auto-encoder to DSSM

Auto-encoder

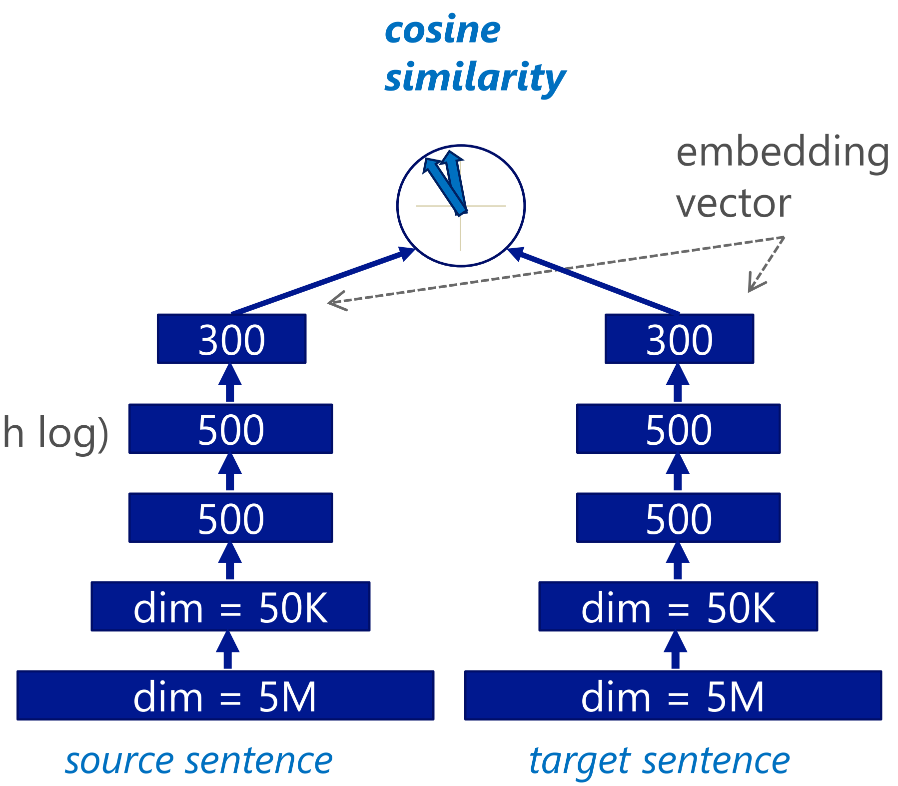


Training loss func.:
AE: reconstruction error
DSSM: distance between
embedding vectors

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



Deep Learning for Semantic Representations

- Sentence to vector
- The deep semantic similarity model (DSSM)
- Convolutional & Recurrent DSSM
- Applications to IR and contextual entity ranking
- Multimodal semantic learning for image captioning

Convolutional DSSM

Model local context at the convolutional layer
Model global context at the pooling layer

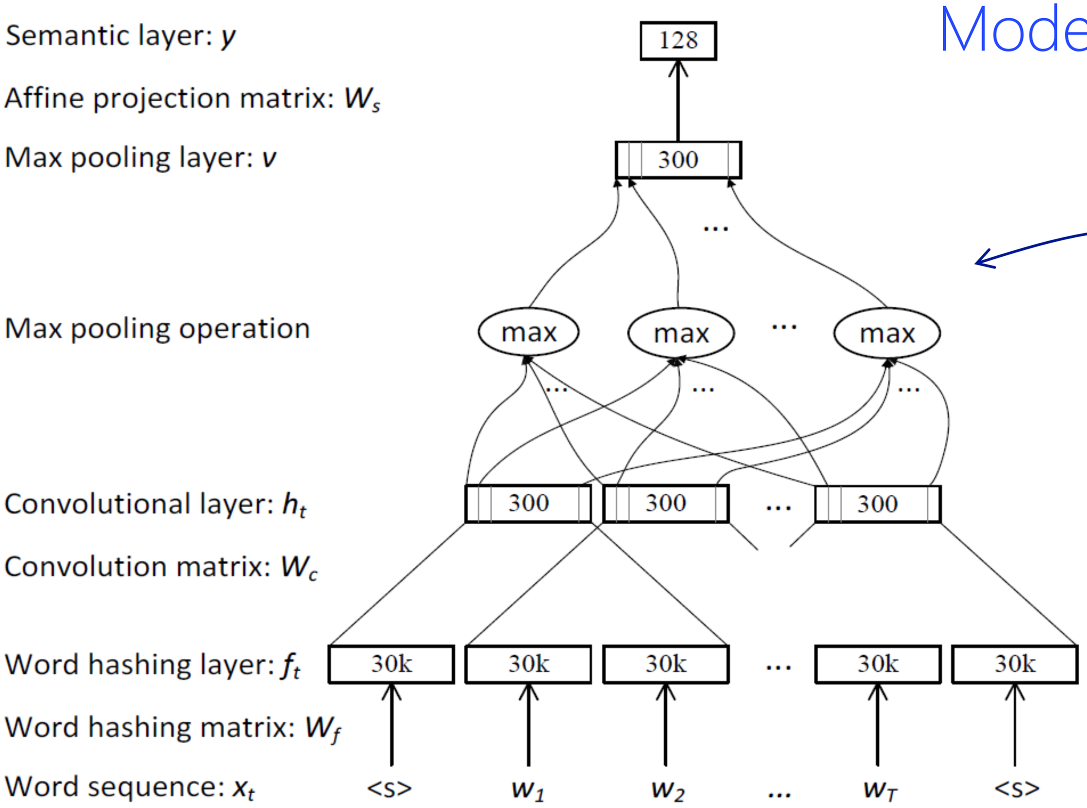
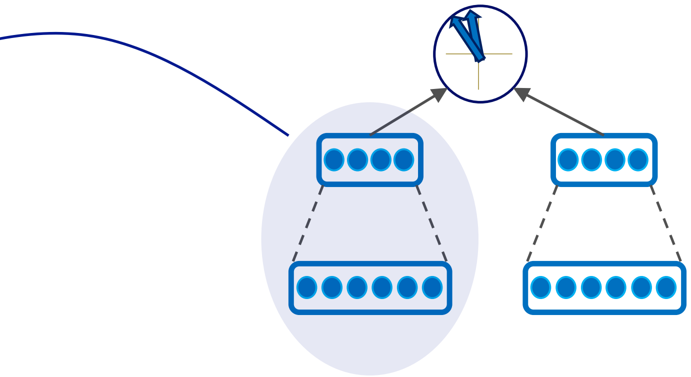


Figure 1: Illustration of the C-DSSM. A convolutional layer with the window size of three is illustrated.

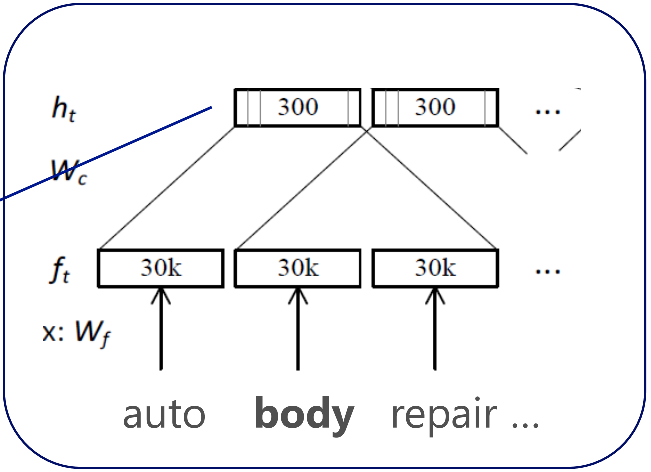


[Shen, He, Gao, Deng, Mesnil, WWW2014 & CIKM2014;
Gao, Pantel, Gamon, He, Deng, Shen, EMNLP2014]

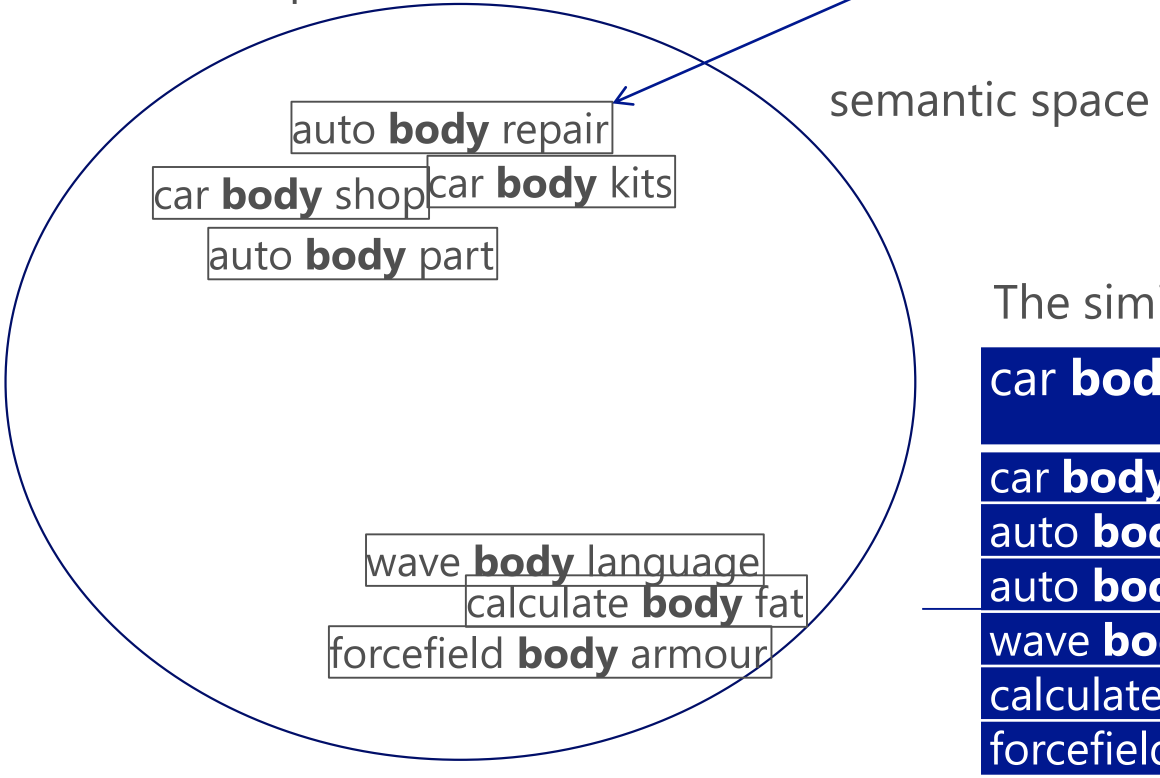
- What does the model learn at the convolutional layer?

Capture the local context dependent word sense

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



$$h_t = W_c \times [f_{t-1}, f_t, f_{t+1}]$$



The similarity between different "***body***" within contexts

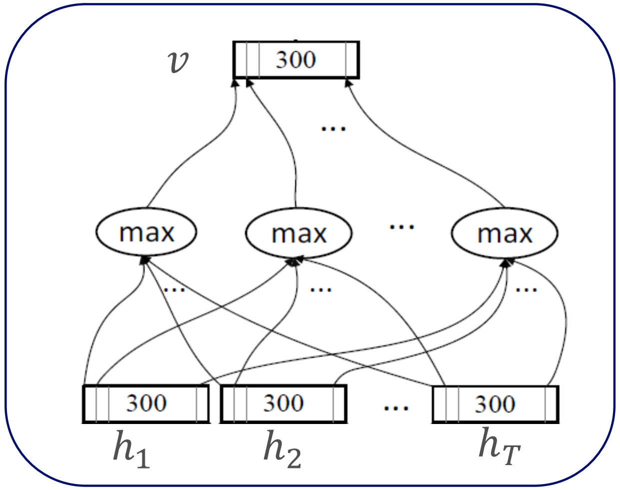
car **body** shop	**cosine similarity**
car **body** kits	0.698
auto **body** repair	0.578
auto **body** parts	0.555
wave **body** language	0.301
calculate **body** fat	0.220
forcefield **body** armour	0.165

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



$$v(i) = \max_{t=1,\dots,T}\{h_t(i)\}$$

where $i = 1, \dots, 300$

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

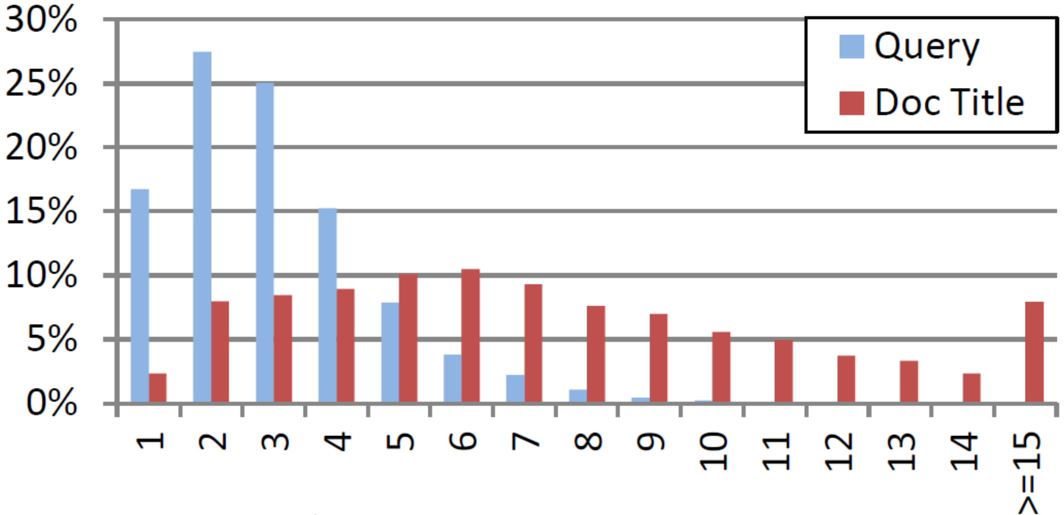
DSSM for Information Retrieval

- 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

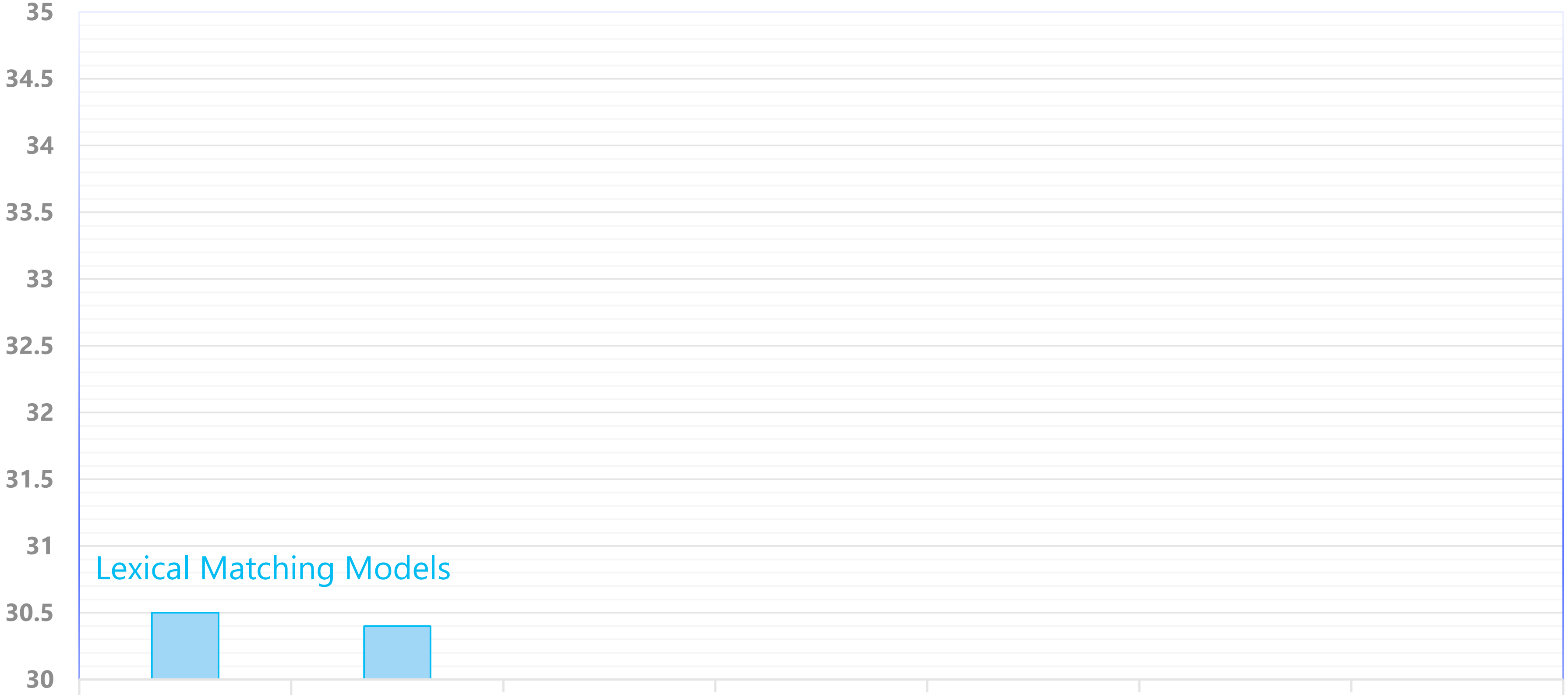
- GPU (NVidia GPU K40)



Dist. of query and doc title length

Main Experiment Results

ULM : Zhai and Lafferty 2001

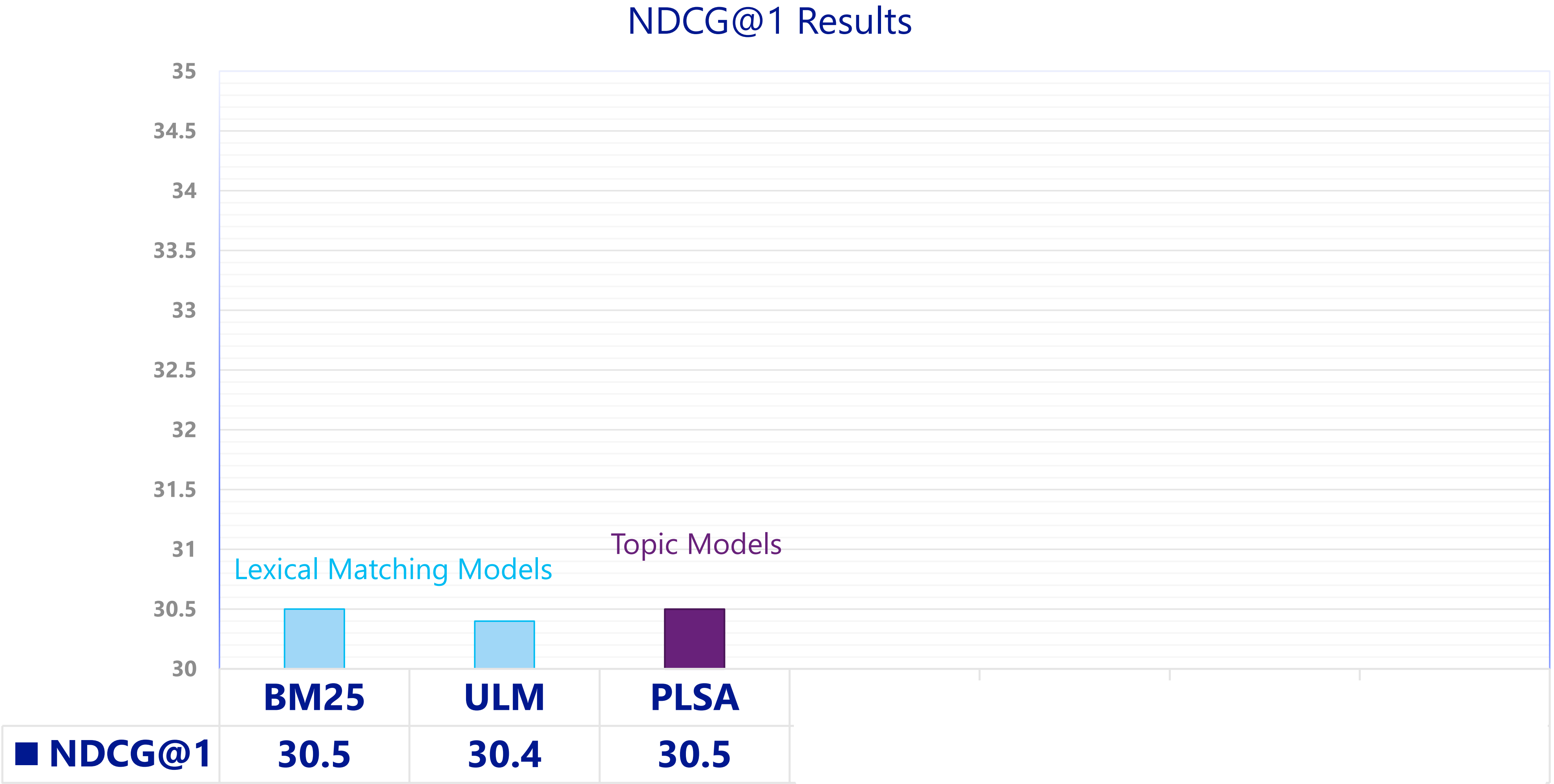


NDCG@1 Results

	BM25	ULM
■ NDCG@1	30.5	30.4

Main Experiment Results

PLSA: Hofmann 1999

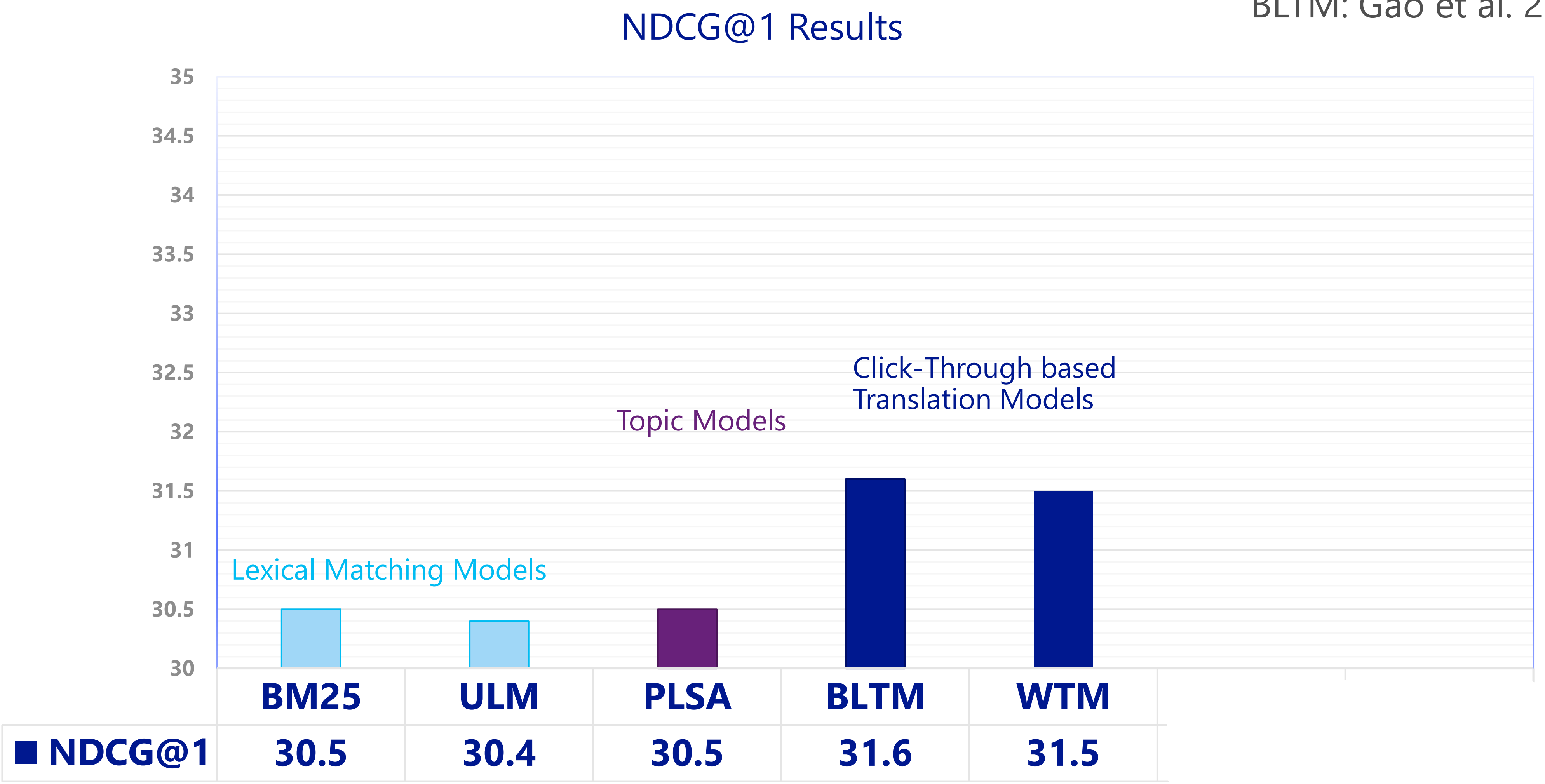


	BM25	ULM	PLSA
■ NDCG@1	30.5	30.4	30.5

Main Experiment Results

WTM: Gao et al. 2010

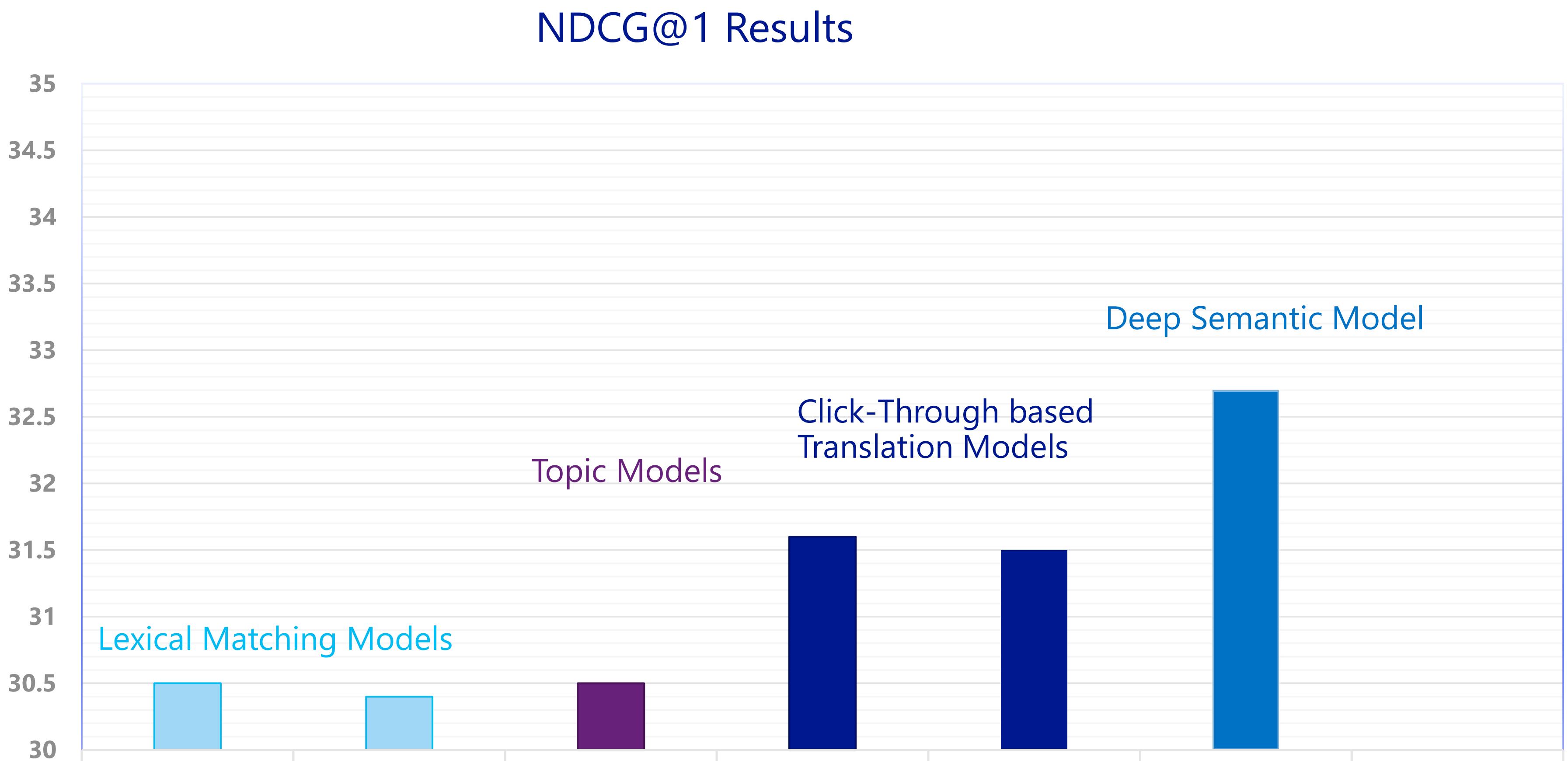
BLTM: Gao et al. 2011



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

Main Experiment Results

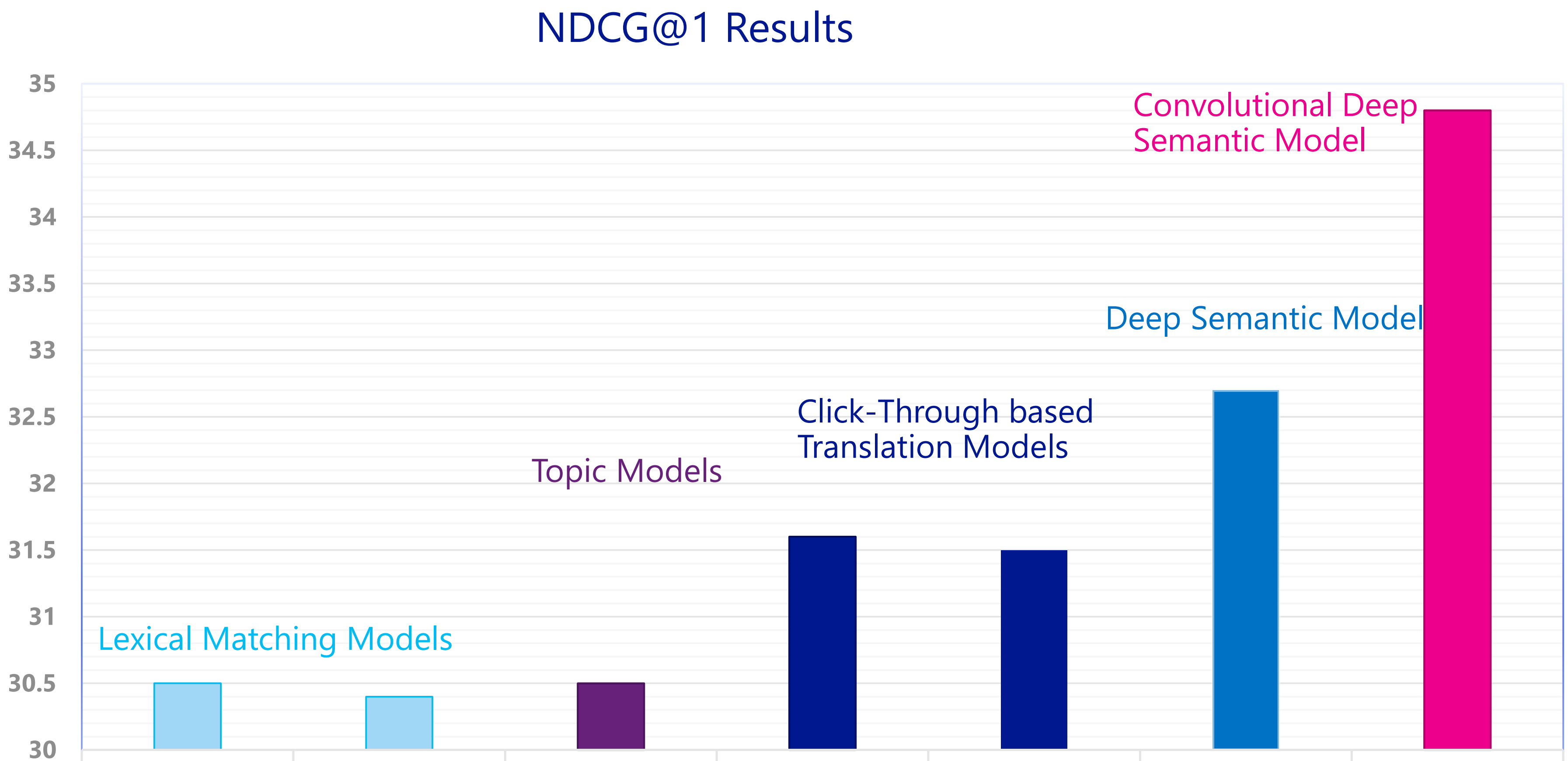
DSSM: Huang et al. 2013



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

Main Experiment Results

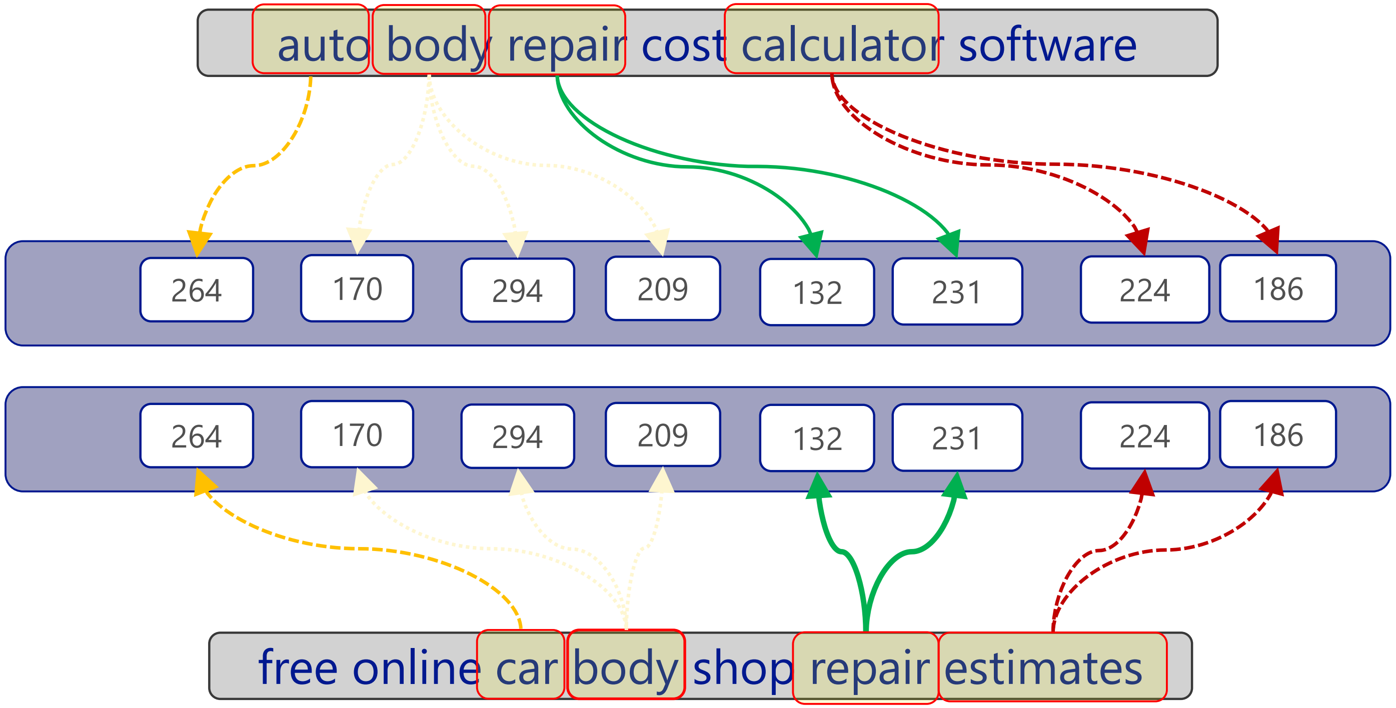
CDSSM: Shen et al. 2014



	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

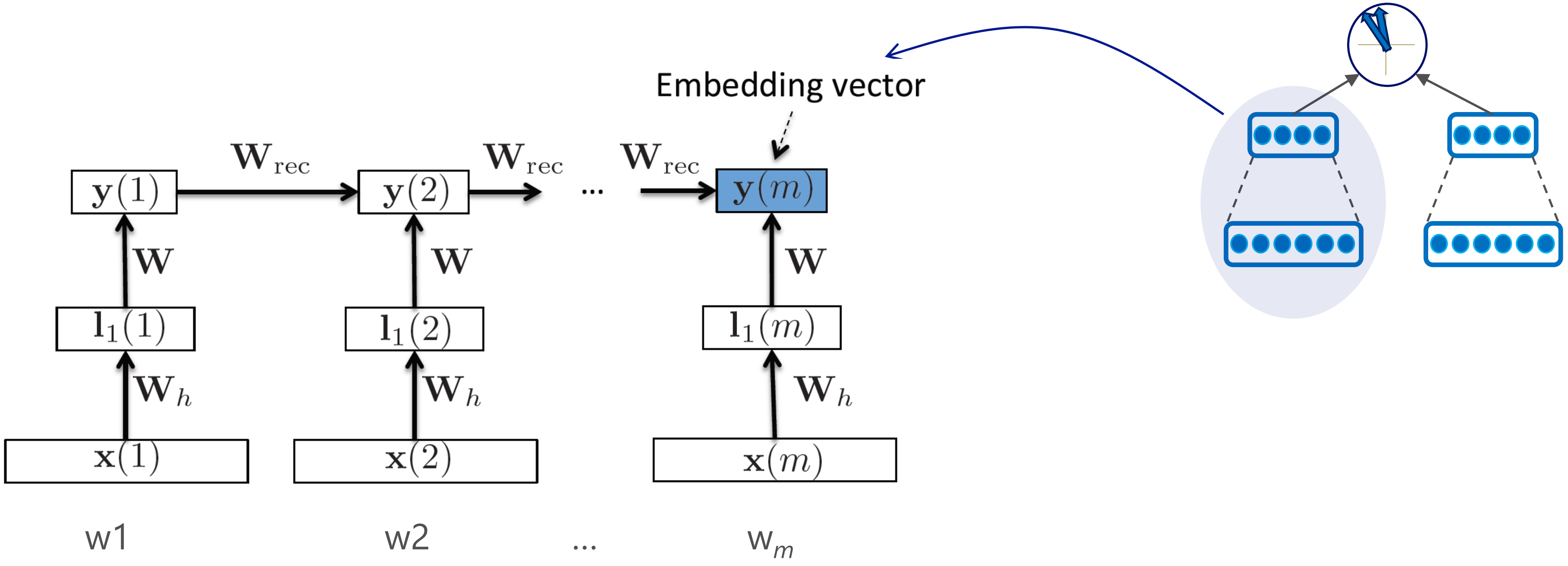
- Semantic matching of query and document



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

Recurrent DSSM

- Encode the word one by one in the recurrent hidden layer
- The hidden layer at the last word codes the semantics of the full sentence
- Model is trained by a cosine similarity driven objective

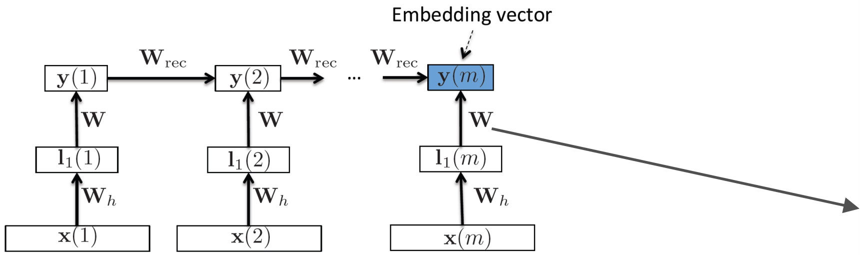


[Palangi, Deng, Shen, Gao, He, Chen, Song, Ward, 2015]

Using LSTM cells

LSTM (long short term memory) uses special cells in RNN

[Hochreiter and J. Schmidhuber, 1997]



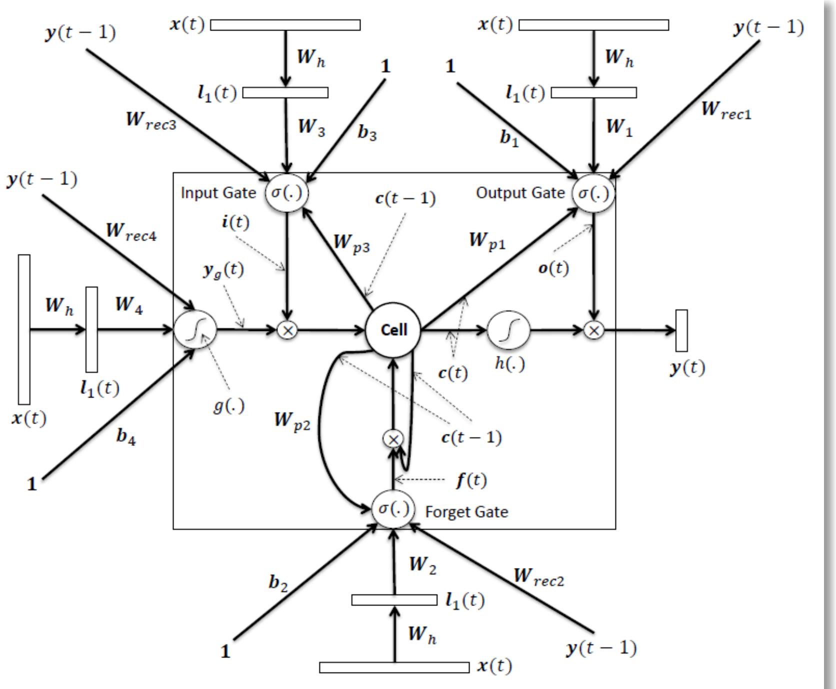


Figure 2. The basic LSTM architecture used for sentence embedding

$$
\begin{aligned}
\mathbf{y}_g(t) &= g(\mathbf{W}_4\mathbf{l}_1(t) + \mathbf{W}_{\text{rec}4}\mathbf{y}(t-1) + \mathbf{b}_4) \\
\mathbf{i}(t) &= \sigma(\mathbf{W}_3\mathbf{l}_1(t) + \mathbf{W}_{\text{rec}3}\mathbf{y}(t-1) + \mathbf{W}_{p3}\mathbf{c}(t-1) + \mathbf{b}_3) \\
\mathbf{f}(t) &= \sigma(\mathbf{W}_2\mathbf{l}_1(t) + \mathbf{W}_{\text{rec}2}\mathbf{y}(t-1) + \mathbf{W}_{p2}\mathbf{c}(t-1) + \mathbf{b}_2) \\
\mathbf{c}(t) &= \mathbf{f}(t) \circ \mathbf{c}(t-1) + \mathbf{i}(t) \circ \mathbf{y}_g(t) \\
\mathbf{o}(t) &= \sigma(\mathbf{W}_1\mathbf{l}_1(t) + \mathbf{W}_{\text{rec}1}\mathbf{y}(t-1) + \mathbf{W}_{p1}\mathbf{c}(t) + \mathbf{b}_1) \\
\mathbf{y}(t) &= \mathbf{o}(t) \circ h(\mathbf{c}(t)) \qquad (2)
\end{aligned}
$$

where $\circ$ denotes Hadamard (element-wise) product.

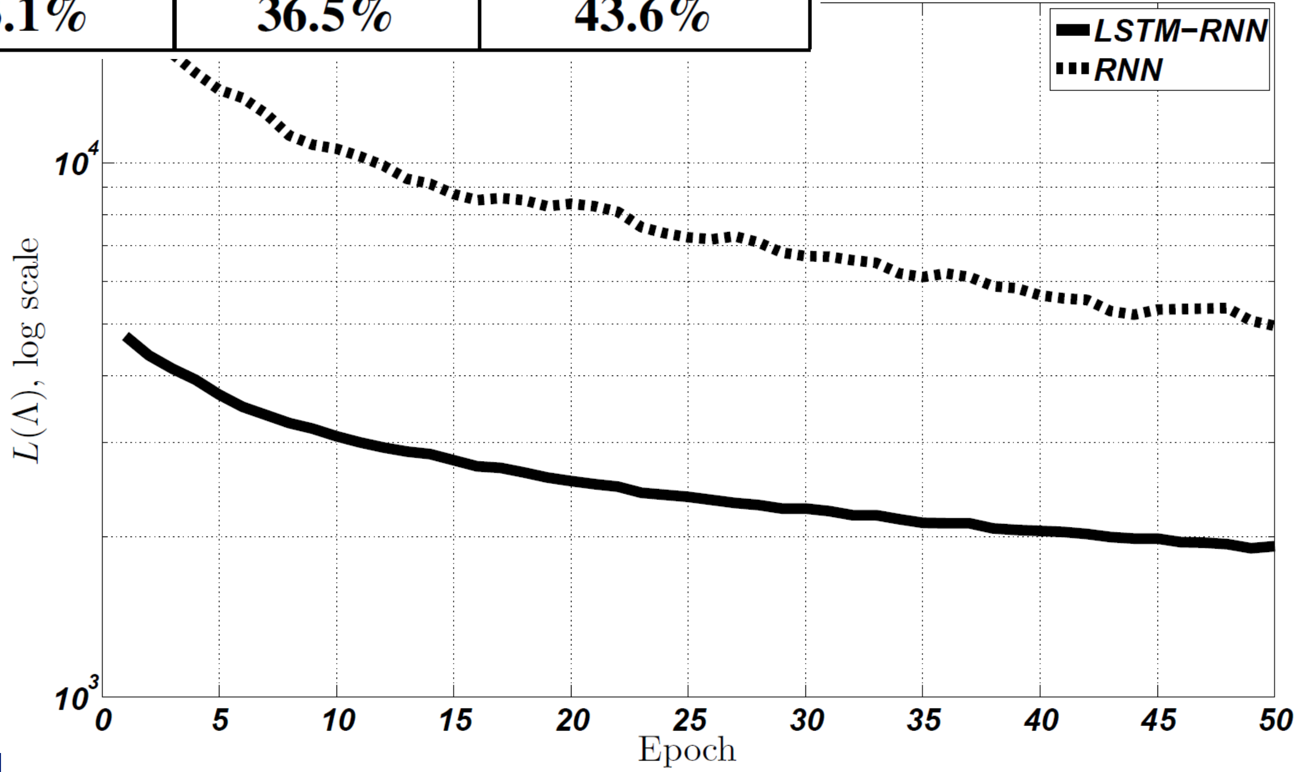
[Palangi, Deng, Shen, Gao, He, Chen, Song, Ward, Deep Sentence Embedding Using the LSTM network: Analysis and Application to IR, 2015]

Results

Model	NDCG@1	NDCG@3	NDCG@10
BM25	30.5%	32.8%	38.8%
PLSA (T=500)	30.8%	33.7%	40.2%
DSSM (nhid = 288/96), 2 Layers	31.0%	34.4%	41.7%
CLSM (nhid = 288/96), 2 Layers	31.8%	35.1%	42.6%
RNN (nhid = 288), 1 Layer	31.7%	35.0%	42.3%
LSTM-RNN (ncell = 96), 1 Layer	**33.1%**	**36.5%**	**43.6%**

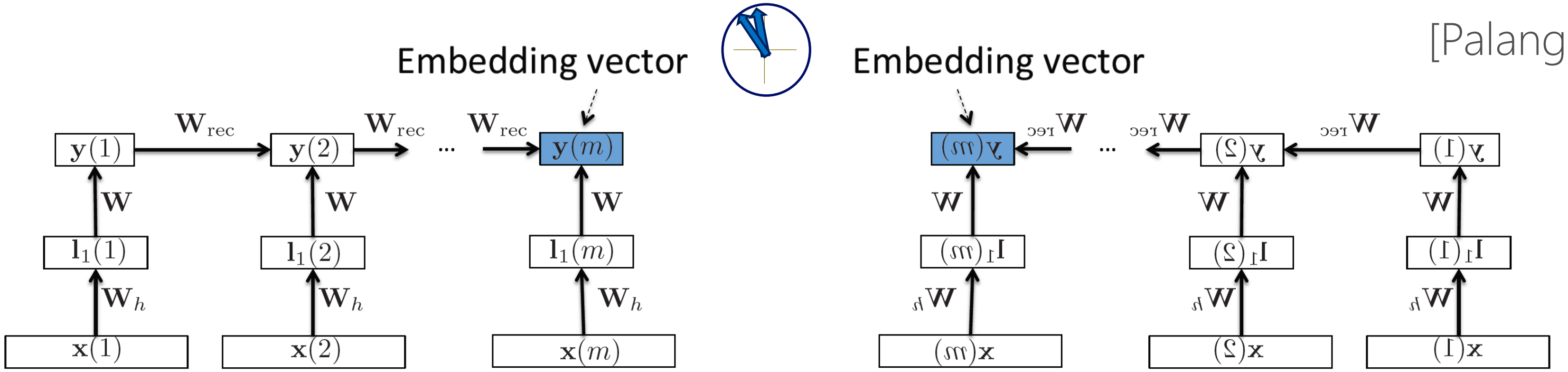
LSTM learns much faster than regular RNN

LSTM effectively represents the semantic information of a sentence using a vector



Related work

[Palangi et al, 2015]

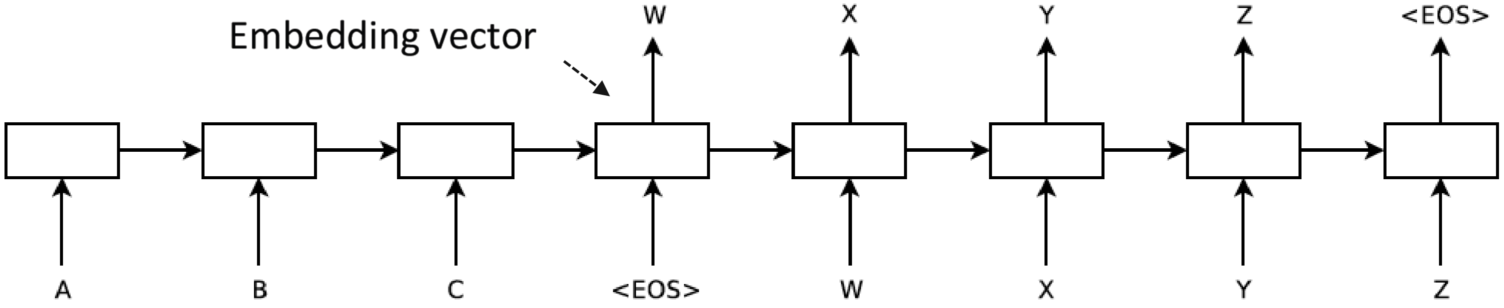


Source sentence

Target sentence

Optimize *sentence-level* semantic similarity

vs.



Source sentence

Target sentence

Optimize *word-level* perplexity

[Sutskever, Vinyals, Le, 2014. Sequence to Sequence Learning with Neural Networks]

Some other related work

Deep CNN for text input
Mainly classification tasks in the paper

[Kalchbrenner, Grefenstette, Blunsom, A Convolutional Neural Network for Modelling Sentences, ACL2014]

Paragraph Vector
Learn a vector for a paragraph

Quoc Le, Tomas Mikolov, Distributed Representations of Sentences and Documents, in ICML 2014

Recursive NN (ReNN)
Tree structure, e.g., for parsing

[Socher, Lin, Ng, Manning, "Parsing natural scenes and natural language with recursive neural networks", 2011]

Tensor product representation (TPR)
Tree representation

[Smolensky and Legendre: The Harmonic Mind, From Neural Computation to Optimality-Theoretic Grammar, MIT Press, 2006]

Tree-structured LSTM Network
Tree structure LSTM

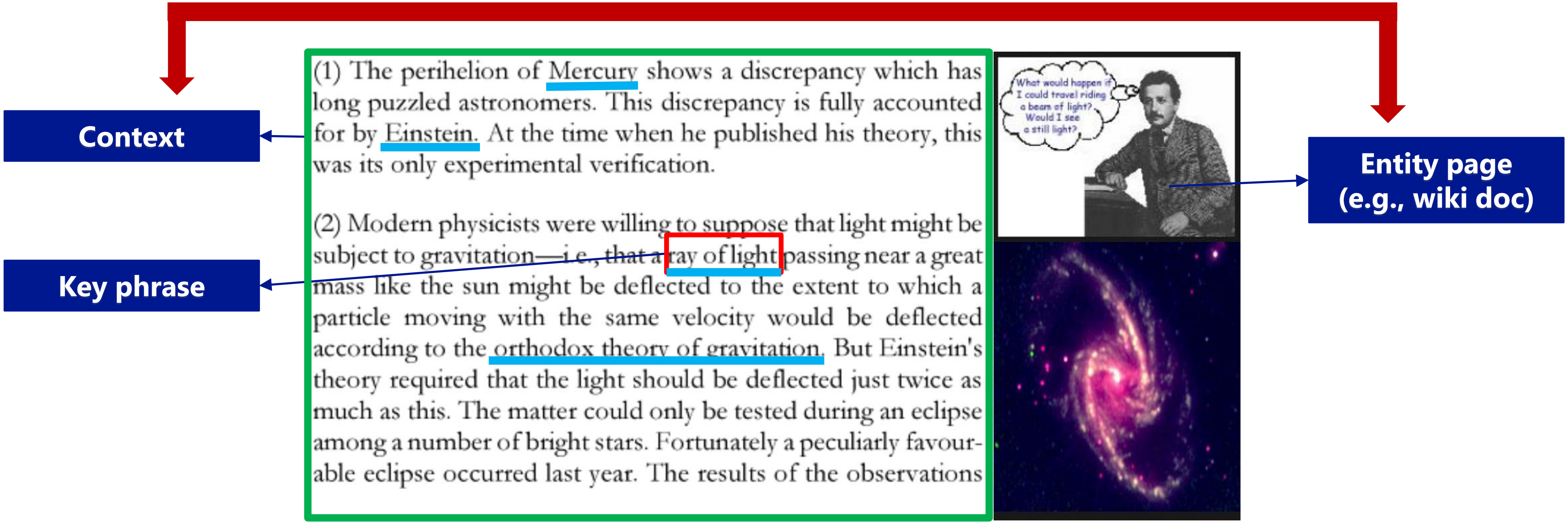
[Tai, Socher, Manning. 2015. Improved Semantic Representations From Tree-Structured LSTM Networks.]

Deep Learning for Semantic Representations

- Sentence to vector
- The deep semantic similarity model (DSSM)
- Convolutional & Recurrent DSSM
- Applications to IR and contextual entity ranking
- Multimodal semantic learning for image captioning

Contextual Entity Ranking

Given a user-highlighted text span representing an entity of interest, search for supplementary document for the entity



Context

Key phrase

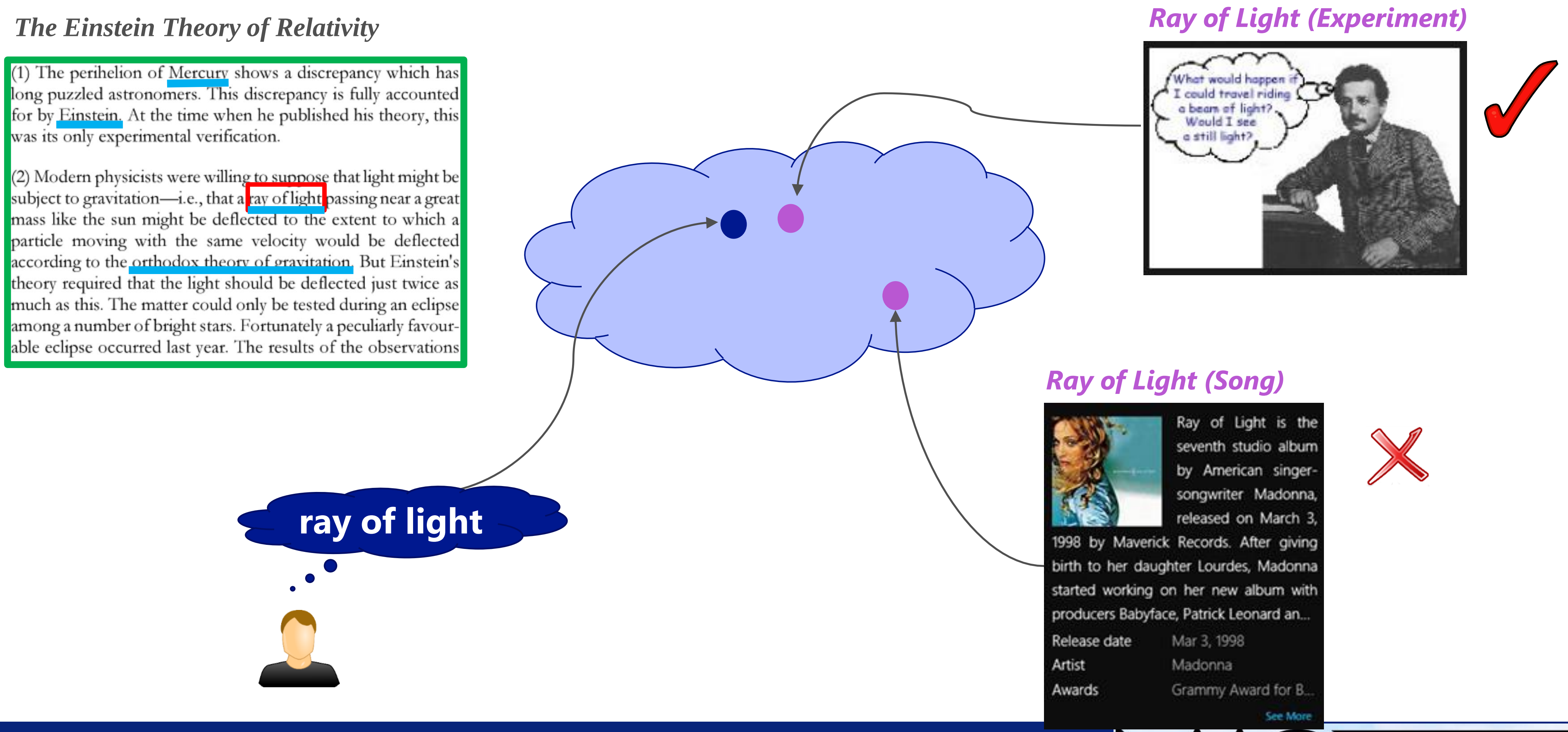
(1) The perihelion of Mercury shows a discrepancy which has long puzzled astronomers. This discrepancy is fully accounted for by Einstein. At the time when he published his theory, this was its only experimental verification.

(2) Modern physicists were willing to suppose that light might be subject to gravitation—i.e., that a ray of light passing near a great mass like the sun might be deflected to the extent to which a particle moving with the same velocity would be deflected according to the orthodox theory of gravitation. But Einstein's theory required that the light should be deflected just twice as much as this. The matter could only be tested during an eclipse among a number of bright stars. Fortunately a peculiarly favourable eclipse occurred last year. The results of the observations

Entity page (e.g., wiki doc)

Gao, Pantel, Gamon, He, Deng, Shen, “Modeling interestingness with deep neural networks.” EMNLP2014

Learning DSSM for contextual entity ranking



The Einstein Theory of Relativity

(1) The perihelion of Mercury shows a discrepancy which has long puzzled astronomers. This discrepancy is fully accounted for by Einstein. At the time when he published his theory, this was its only experimental verification.

(2) Modern physicists were willing to suppose that light might be subject to gravitation—i.e., that a ray of light passing near a great mass like the sun might be deflected to the extent to which a particle moving with the same velocity would be deflected according to the orthodox theory of gravitation. But Einstein's theory required that the light should be deflected just twice as much as this. The matter could only be tested during an eclipse among a number of bright stars. Fortunately a peculiarly favourable eclipse occurred last year. The results of the observations

Ray of Light (Experiment)

Ray of Light (Song)

Ray of Light is the seventh studio album by American singer-songwriter Madonna, released on March 3, 1998 by Maverick Records. After giving birth to her daughter Lourdes, Madonna started working on her new album with producers Babyface, Patrick Leonard an...

Release date Mar 3, 1998
Artist Madonna
Awards Grammy Award for B...

See More

Extract Labeled Pairs from Web Browsing Logs
Contextual Entity Search

- When a hyperlink H points to a Wikipedia P'

http://runningmoron.blogspot.in/

…

I spent a lot of time finding music that was motivating and that I'd also want to listen to through my phone. I could find none. None! I wound up downloading three Metallica songs, a Judas Priest song and one from Bush.

…

http://en.wikipedia.org/wiki/Bush_(band)

Bush (band)

From Wikipedia, the free encyclopedia

For the Canadian band, see Bush (Canadian band).

Bush are a British rock band formed in London in 1992.

The grunge band found its immediate success with the release of their debut album *Sixteen Stone* in 1994, which is certified 6× multi-platinum by the RIAA.[3] Bush went on to become one of the most commercially successful rock bands of the 1990s, selling over 10 million records in the United States. Despite their success in the United States, the band was less well known in their home country and enjoyed only marginal success

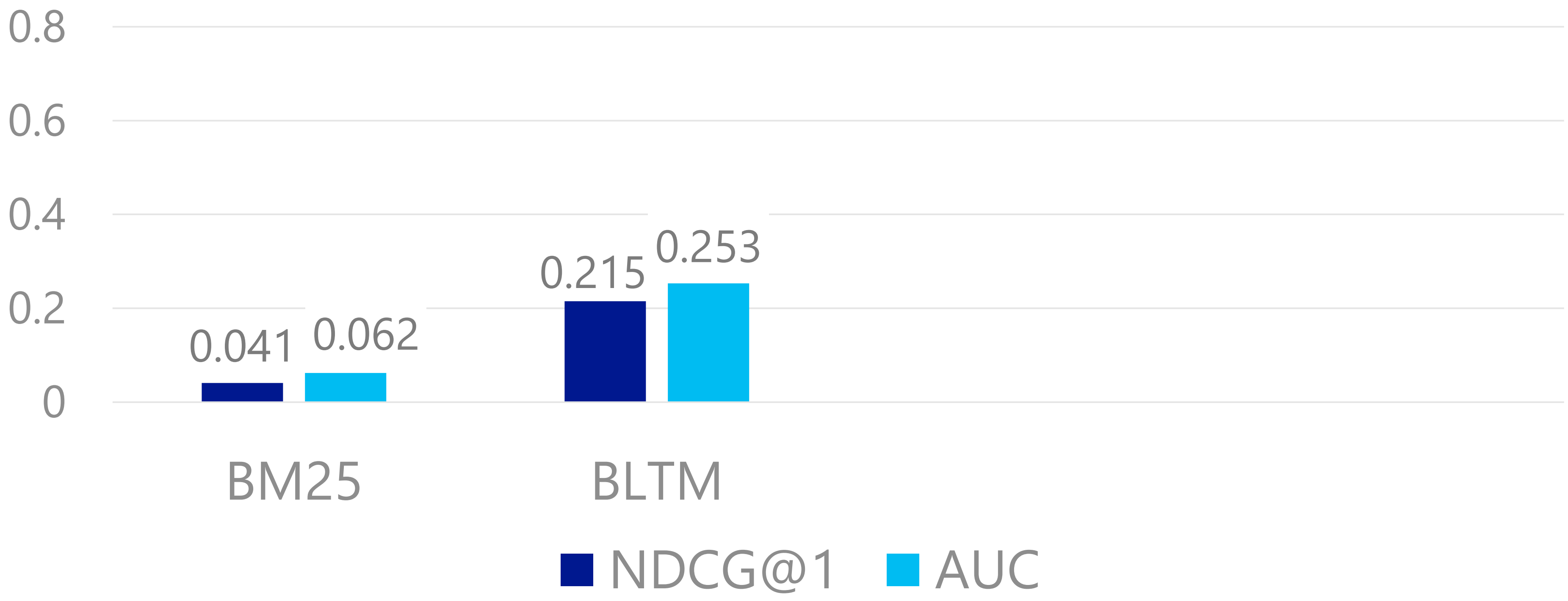
Bush performing in Texas 2011.

- (anchor text of H & surrounding words, text in P')

Contextual Entity Search: Settings

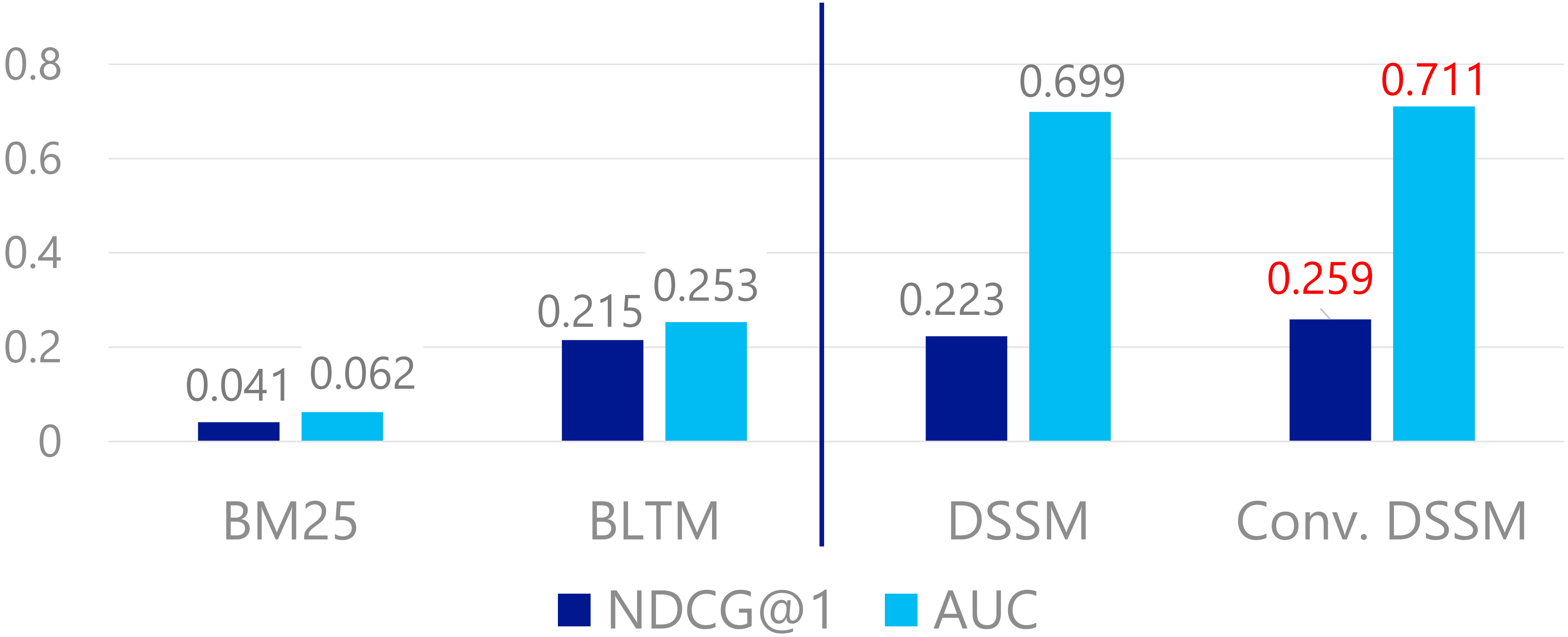
- Training/validation data: 18M of user clicks in wiki pages
- Evaluation data
  - Sample 10k Web documents as the source documents
  - Use named entities in the doc as query; retain up to 100 returned documents as target documents
  - Manually label whether each target document is a good page describing the entity
  - 870k labeled pairs in total
- Evaluation metric: NDCG and AUC

Contextual Entity Search Results: Baselines



- BM25: The classical document model in IR [Robertson+ 1994]
- BLTM: Bilingual Topic Model [Gao+ 2011]

Contextual Entity Search Results: DSSM



- DSSM: bag-of-words input
- Conv. DSSM: convolutional DSSM

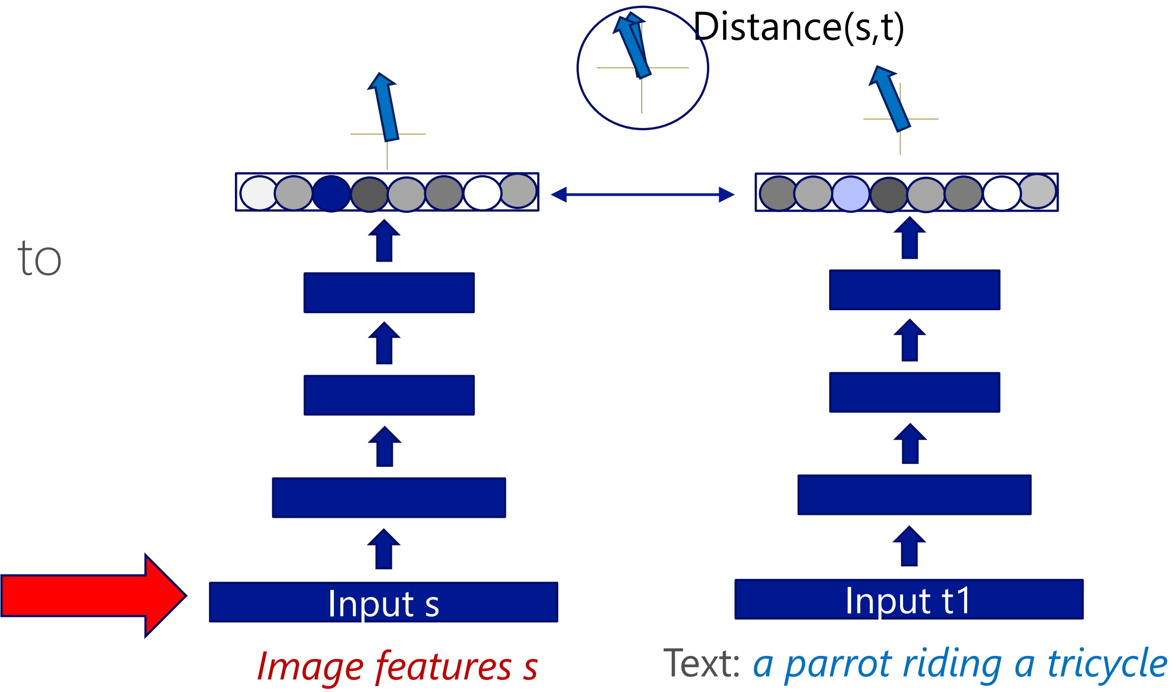
Deep Learning for Semantic Representations

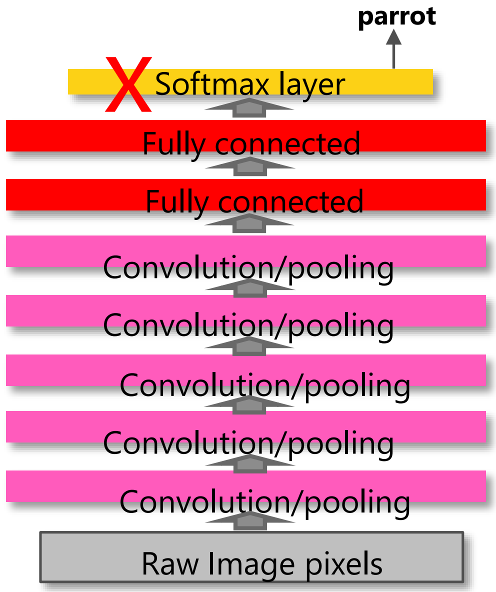
- Sentence to vector
- The deep semantic similarity model (DSSM)
- Convolutional & Recurrent DSSM
- Applications to IR and contextual entity ranking
- Multimodal semantic learning and image captioning

Deep Multimodal Similarity Model (DMSM)

Multimodal DSSM for image-text joint learning

- Recall DSSM for text inputs: *s, t*
- Now: replace text s by image s
- Pick complete captions affinitize to complete images





Q = image, D = caption, R = relevance

Relevance: $R(Q, D) = \text{cosine}(y_Q, y_D) = \frac{{y_Q}^T y_D}{\|y_Q\|\|y_D\|}$

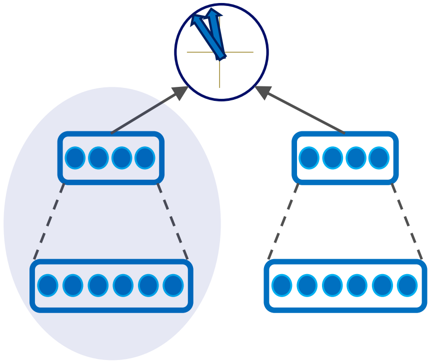
Caption probability: $P(D|Q) = \frac{\exp(\gamma R(Q, D))}{\Sigma_{D' \in \mathbf{D}} \exp(\gamma R(Q, D'))}$

Candidate captions ↗ ↖ Smoothing factor

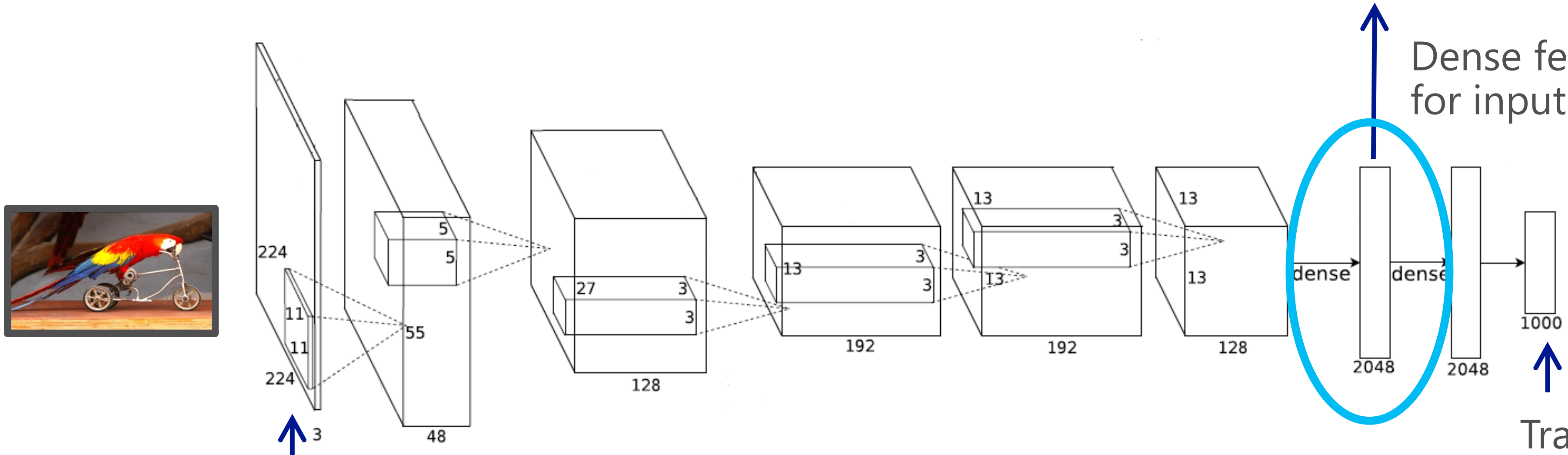
Objective: $L(\Lambda) = -\log \prod_{(Q,D^+)} P(D^+|Q)$

↖ Correct caption

The convolutional network at the image side



Feed the pre-trained image feature vector into the image side of the DMSM



Dense feature vector for input image

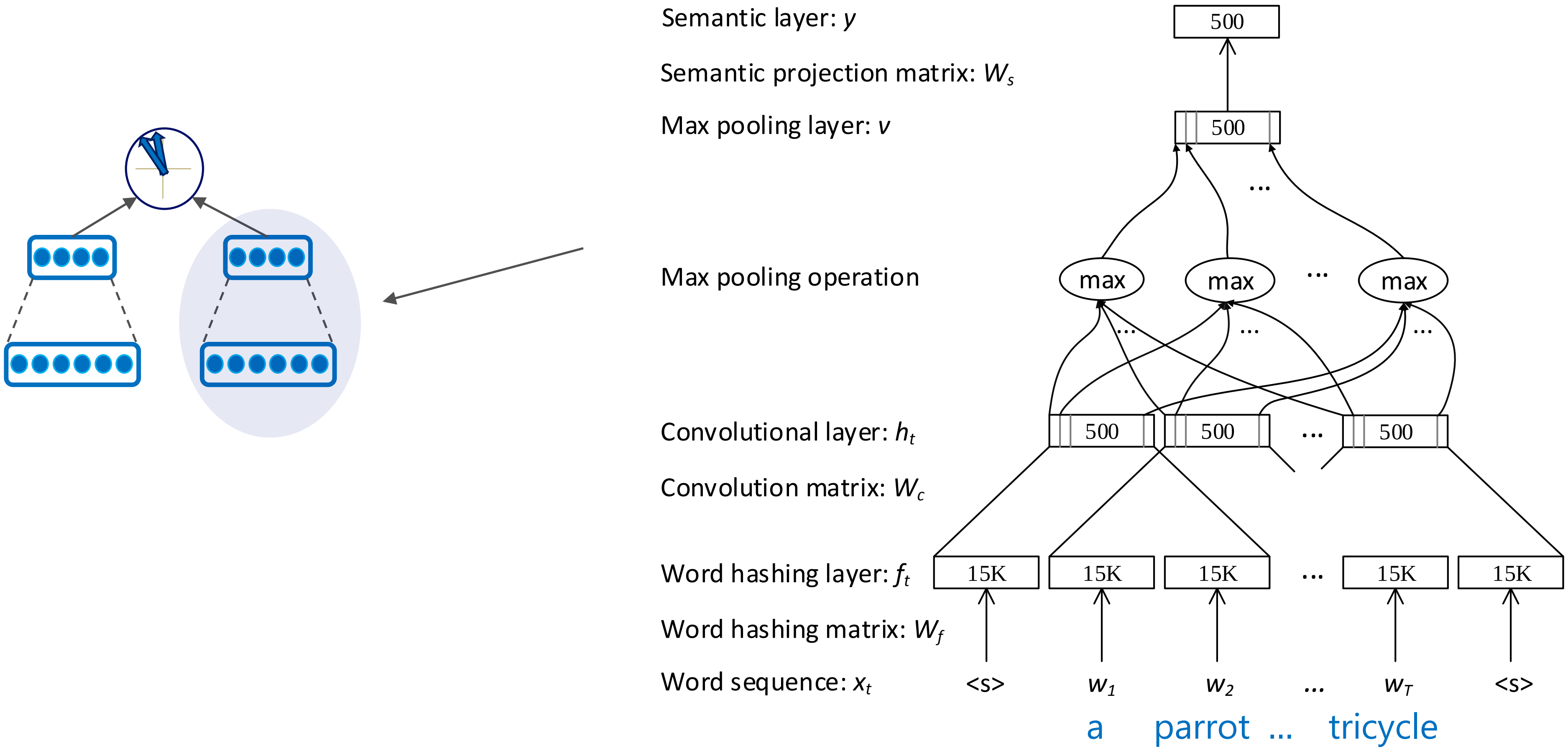
Raw pixels from input box

Trained to predict object in image

Pretrained from ImageNet [Krizhevsky et al., 2012]

The convolutional network at the caption side

Models fine-grained structural language information in the caption



Using convolutional neural network for the text caption side

The task: Image -> Language

- Why important?

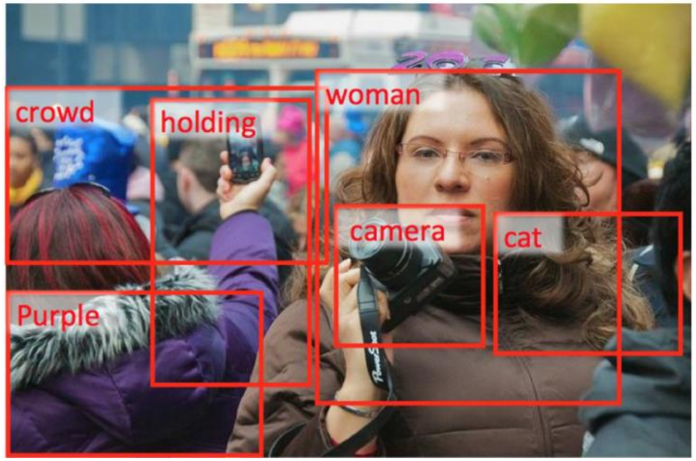
  For building intelligent machines that understand the semantics in complex scenes
  And language is like a regulator for *understanding as human do*.

- Why difficult?

  Need to detect multiple objects in arbitrary regions, and need to capture the complex semantics among these objects.

- What different (e.g., vs. ImageNet / object categorization)?

  Capturing the salient, coherent semantic information embedded in a picture.



➡ A woman holding a camera in a crowd.

The MSR system

Understand the image stage by stage:

Image word detection

Deep-learned features, applied to likely items in the image, trained to produce words in captions

Language generation

Maxent language model, trained on caption, conditional on words detected from the image

Global semantic re-ranking

Hypothetical captions re-ranked by deep-learned multi-modal similarity model looking at the entire image

Fang, Gupta, Iandola, Srivastava, Deng, Dollar, Gao, He, Mitchell, Platt, Zitnick, Zweig, "From Captions to Visual Concepts and Back," CVPR, June 2015

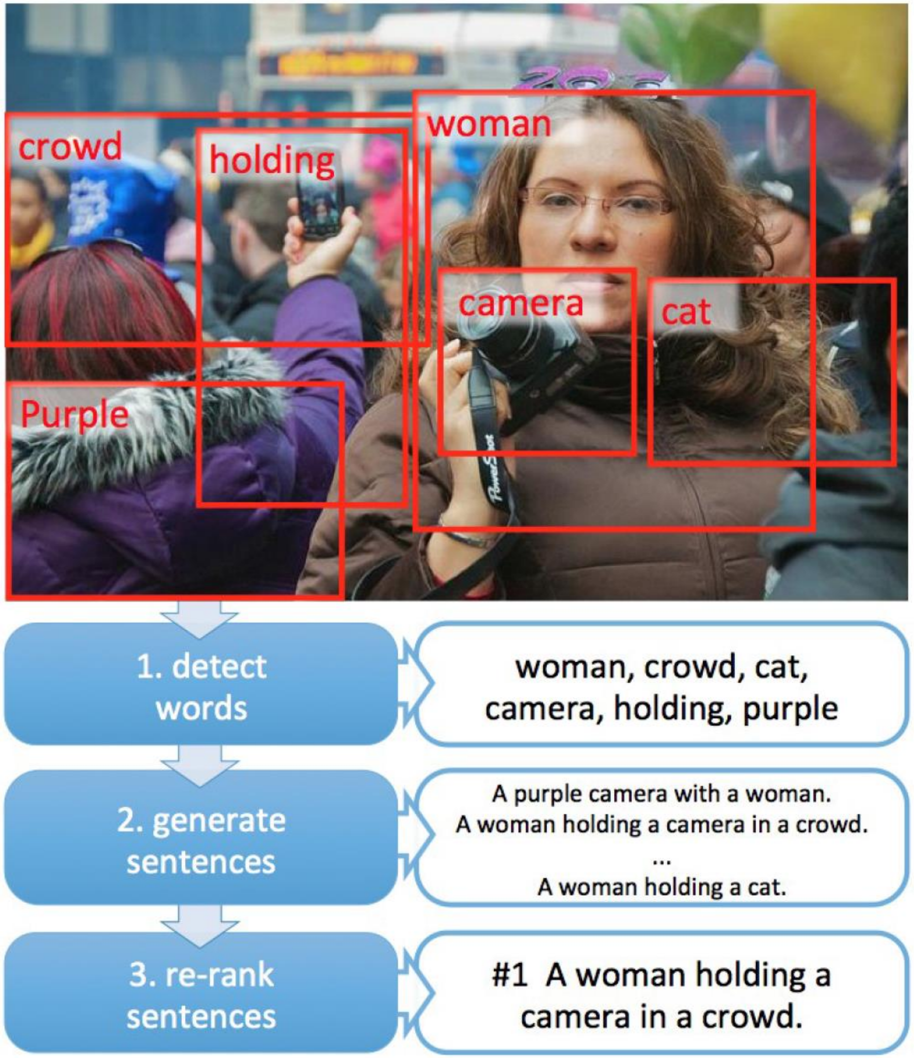
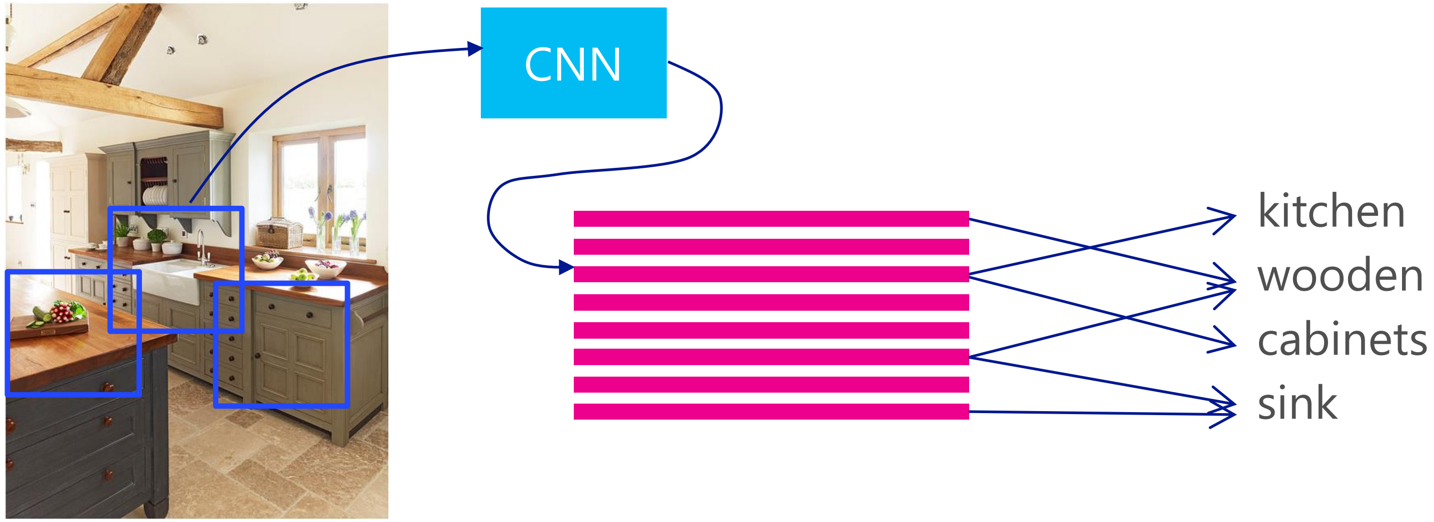


Figure 1. An illustrative example of our pipeline.

Train to predict words in captions



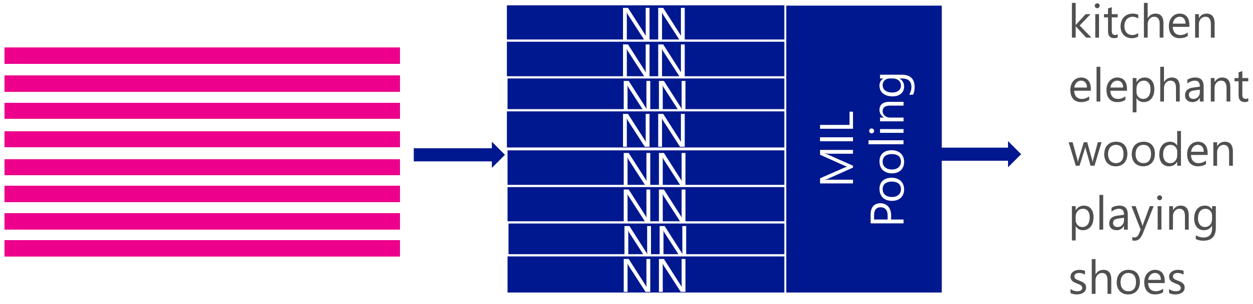
CNN

kitchen
wooden
cabinets
sink

Which words should be detected? Let a neural network figure it out

The prob that the j-th box of the i-th image corresponds to word *w* is

$$p_{ij}^{w} = \frac{1}{1 + \exp\left(-(\mathbf{v}_{\mathbf{w}}^{\mathbf{t}}\phi(b_{ij}) + u_w)\right)}$$



NN
NN
NN
NN
NN
NN
NN
NN

MIL Pooling

kitchen
elephant
wooden
playing
shoes

Vocabulary = the 1000 most common words in the training captions (92% of data)

Map features to likely image words

- Train with Multiple Instance Learning (MIL)
  - Use noisy-OR version (Zhang et al., 2005)

- For each word *w*, MIL uses positive and negative bags of bounding boxes
  - For each image i:
    - We have the "bag of boxes", b_i
    - b_i is **positive** if *w* in i's description
    - b_i is **negative** if *w* not in i's description
  - Probability that image i manifests word *w*, p_i^w:

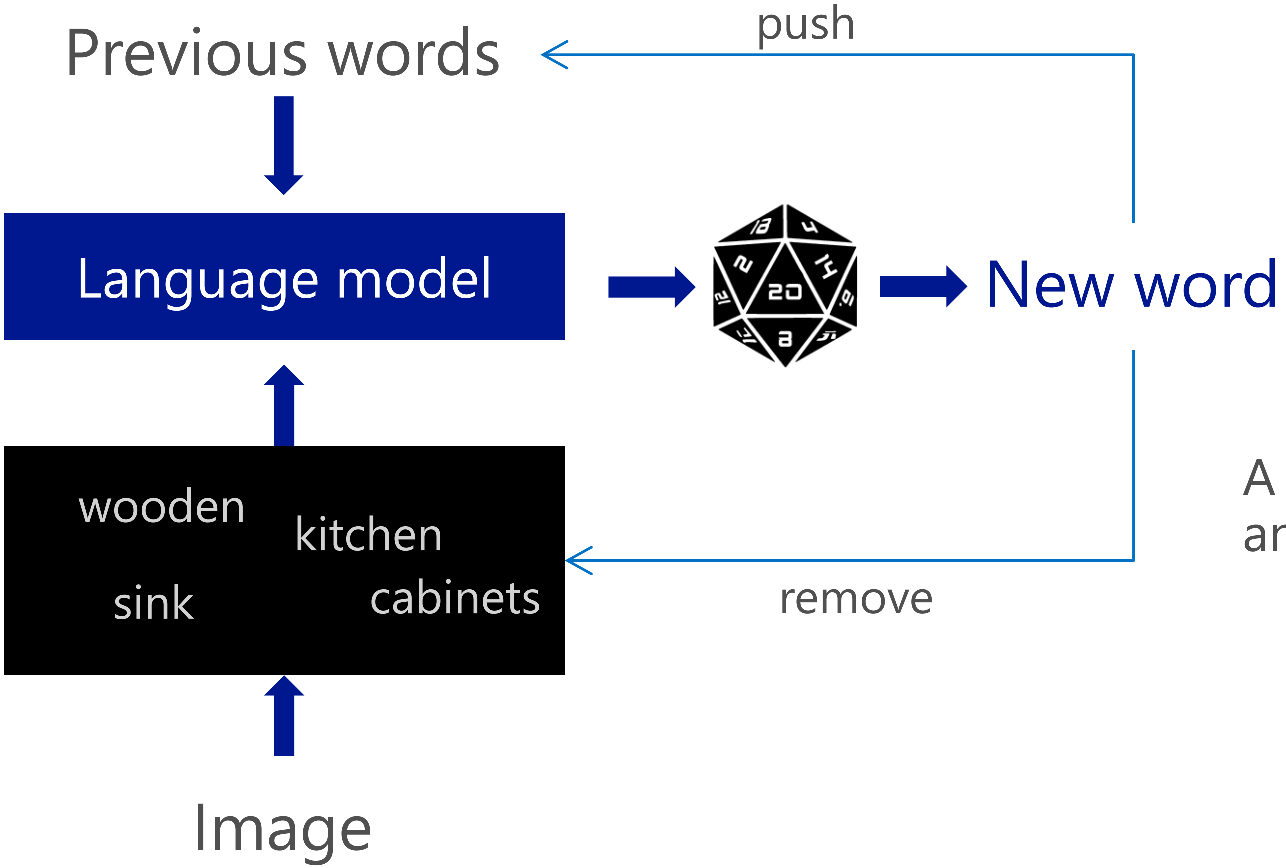
$$p_i^w = 1 - \prod_{j \in b_i} \left(1 - p_{ij}^w\right)$$

Each bounding box in image → $j \in b_i$

Calculated from CNN (last slide) → p_{ij}^w

Language models with a blackboard

A LM generates caption candidates given detected words



Previous words

push

Language model

New word

wooden kitchen sink cabinets

remove

Image

A kitchen with wooden cabinets and a sink

Maximum Entropy Language Model

- Berger et al., 1996

Word probability:

$$\Pr(w_l = \bar{w}_l | \bar{w}_{l-1}, \cdots, \bar{w}_1, \text{<s>}, \tilde{\mathcal{V}}_{l-1}) = \frac{\exp\left[\sum_{k=1}^{K} \lambda_k f_k(\bar{w}_l, \bar{w}_{l-1}, \cdots, \bar{w}_1, \text{<s>}, \tilde{\mathcal{V}}_{l-1})\right]}{\sum_{v \in \mathcal{V} \cup \text{</s>}} \exp\left[\sum_{k=1}^{K} \lambda_k f_k(v, \bar{w}_{l-1}, \cdots, \bar{w}_1, \text{<s>}, \tilde{\mathcal{V}}_{l-1})\right]}$$

Feature	Type	Definition	Description
Attribute	0/1	$\bar{w}_l \in \tilde{\mathcal{V}}_{l-1}$	Predicted word is in the attribute set, i.e. has been visually detected and not yet used.
N-gram +	0/1	$\bar{w}_{l-N+1}, \cdots, \bar{w}_l = \kappa$ and $\bar{w}_l \in \tilde{\mathcal{V}}_{l-1}$	N-gram ending in predicted word is κ and the predicted word is in the attribute set.
N-gram -	0/1	$\bar{w}_{l-N+1}, \cdots, \bar{w}_l = \kappa$ and $\bar{w}_l \notin \tilde{\mathcal{V}}_{l-1}$	N-gram ending in predicted word is κ and the predicted word is not in the attribute set.
End	0/1	$\bar{w}_l = \kappa$ and $\tilde{\mathcal{V}}_{l-1} = \emptyset$	The predicted word is κ and all attributes have been mentioned.
Score	$\mathbb{R}$	score($\bar{w}_l$) when $\bar{w}_l \in \tilde{\mathcal{V}}_{l-1}$	The log-probability of the predicted word when it is in the attribute set.

Objective:

All sentences

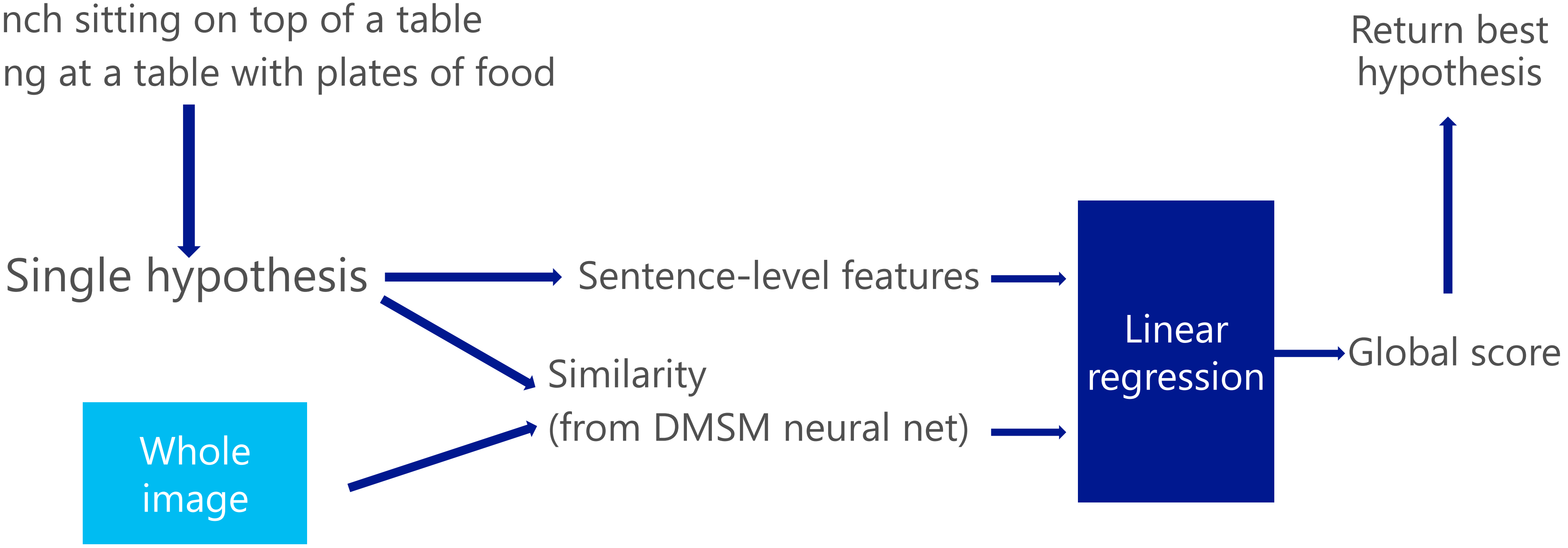
Sentence length

$$L(\Lambda) = \sum_{s=1}^{S} \sum_{l=1}^{\#(s)} \log \Pr(\bar{w}_l^{(s)} | \bar{w}_{l-1}^{(s)}, \cdots, \bar{w}_1^{(s)}, \text{<s>}, \tilde{\mathcal{V}}_{l-1}^{(s)})$$

Rerank hypotheses globally using DMSM

Top 500 hypotheses from the language model

A man sitting on a bench
A man sitting on a table
A white bench sitting on top of a table
A man sitting at a table with plates of food



Return best hypothesis

Single hypothesis

Sentence-level features

Similarity
(from DMSM neural net)

Whole image

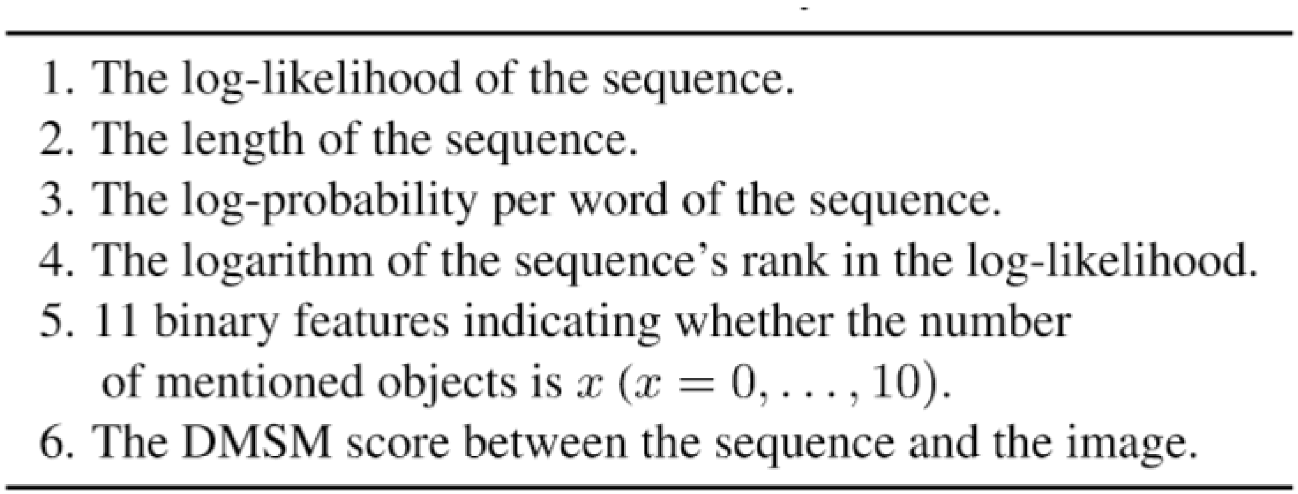
Linear regression

Global score

Image features from AlexNet (Krizhevsky et al., 2012) or VGG (Simonyan and Zisserman, 2014). They are fine-tuned with in-domain image data for DMSM

Linear regression based ranker

- Minimum error rate training (MERT) uses linear combination of features
- Trained on M-best lists using BLEU



1. The log-likelihood of the sequence.
2. The length of the sequence.
3. The log-probability per word of the sequence.
4. The logarithm of the sequence's rank in the log-likelihood.
5. 11 binary features indicating whether the number of mentioned objects is x ($x = 0, \ldots, 10$).
6. The DMSM score between the sequence and the image.

The MS COCO Benchmark

Microsoft COCO
Common Objects in Context

http://mscoco.org/

What is Microsoft COCO?

Microsoft COCO is a new image recognition, segmentation, and captioning dataset. Microsoft COCO has several features:

- ✓ **Object segmentation**
- ✓ **Recognition in Context**
- ✓ **Multiple objects per image**
- ✓ **More than 300,000 images**
- ✓ **More than 2 Million instances**
- ✓ **80 object categories**
- ✓ **5 captions per image**

Collaborators

Tsung-Yi Lin Cornell Tech

Michael Maire TTI Chicago

Serge Belongie Cornell Tech

Lubomir Bourdev Facebook AI

Ross Girshick Microsoft Research

James Hays Brown University

Pietro Perona Caltech

Deva Ramanan UC Irvine

Larry Zitnick Microsoft Research

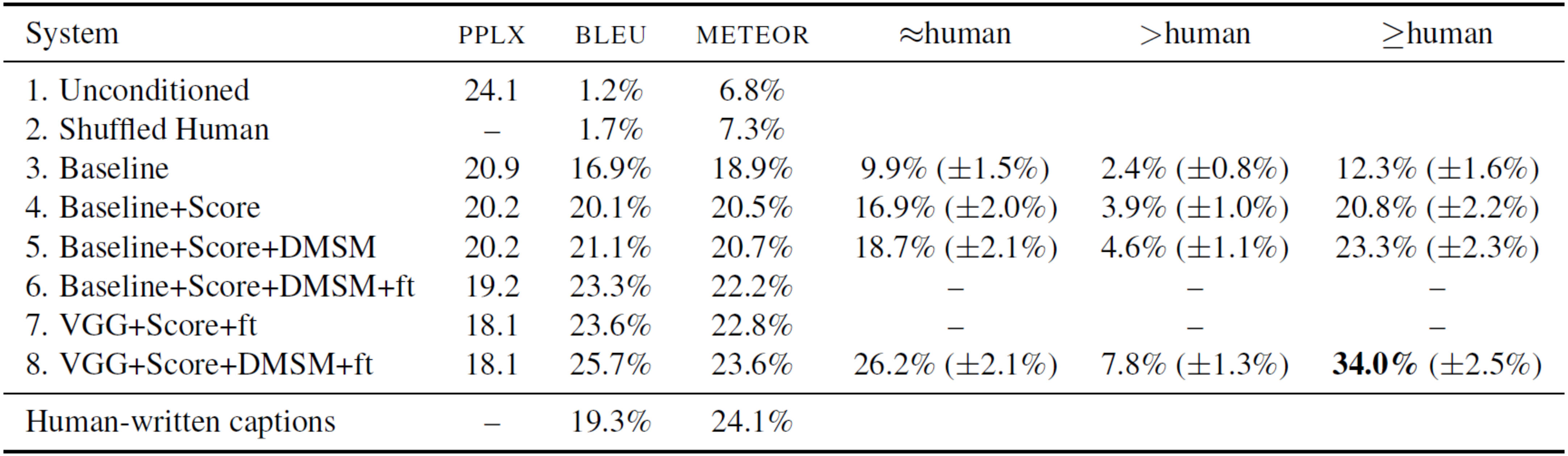
Piotr Dollár Facebook AI

CORNELL NYCTECH
Caltech
facebook
Brown University
UCIrvine University of California, Irvine
Microsoft Research

The man at bat readies to swing at the pitch while the umpire looks on.

A large bus sitting next to a very tall building.

Results



System	PPLX	BLEU	METEOR	≈human	>human	≥human
1. Unconditioned	24.1	1.2%	6.8%			
2. Shuffled Human	–	1.7%	7.3%			
3. Baseline	20.9	16.9%	18.9%	9.9% (±1.5%)	2.4% (±0.8%)	12.3% (±1.6%)
4. Baseline+Score	20.2	20.1%	20.5%	16.9% (±2.0%)	3.9% (±1.0%)	20.8% (±2.2%)
5. Baseline+Score+DMSM	20.2	21.1%	20.7%	18.7% (±2.1%)	4.6% (±1.1%)	23.3% (±2.3%)
6. Baseline+Score+DMSM+ft	19.2	23.3%	22.2%	–	–	–
7. VGG+Score+ft	18.1	23.6%	22.8%	–	–	–
8. VGG+Score+DMSM+ft	18.1	25.7%	23.6%	26.2% (±2.1%)	7.8% (±1.3%)	**34.0%** (±2.5%)
Human-written captions	–	19.3%	24.1%			

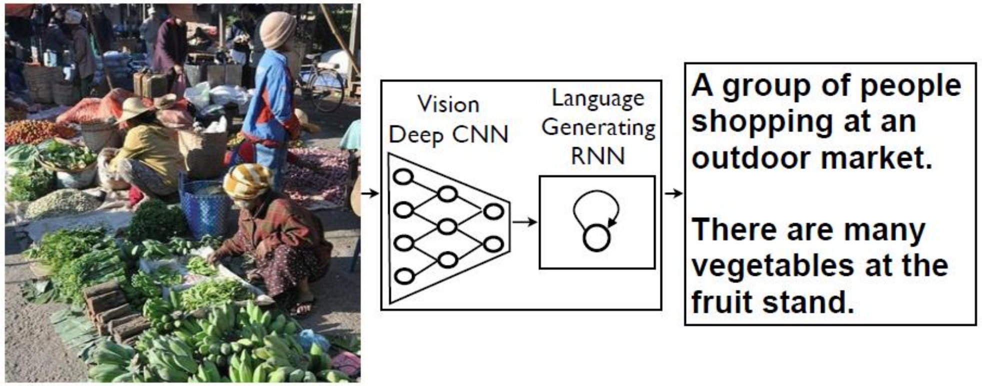
* we use 4 references when measuring BLEU and METEOR, while the official COCO eval server uses 5 references.

DMSM gives additional 2.1 pt BLEU (8 vs. 7) over a strong system.

Compared to human, our system is better or equal 34% of the time.

Related work

Use CNN to generate a whole-image feature vector,
then feed it into a LSTM language model to generate the caption.



A group of people shopping at an outdoor market.

There are many vegetables at the fruit stand.

Figure 1. NIC, our model, is based end-to-end on a neural network consisting of a vision CNN followed by a language generating RNN. It generates complete sentences in natural language from an input image, as shown on the example above.

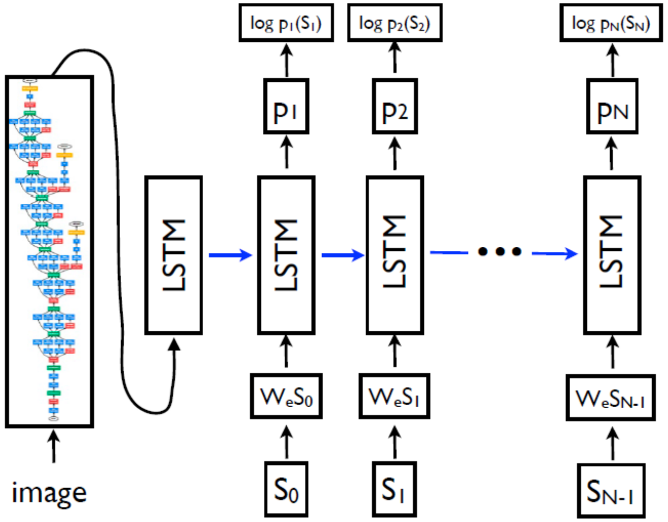


Figure 3. LSTM model combined with a CNN image embedder (as defined in [30]) and word embeddings. The unrolled connections between the LSTM memories are in blue and they correspond to the recurrent connections in Figure 2. All LSTMs share the same parameters.

Vinyals, Toshev, Bengio, Erhan, "**Show and Tell: A Neural Image Caption Generator**", CVPR 2015

Some other related work

Andrej and Fei-Fei, "**Deep Visual-Semantic Alignments for Generating Image Descriptions**". CVPR 2015

Use CNN to generate an image feature vector, then input it, at the 1st step, into a multimodal RNN language model to generate the caption.

Kiros, Salakhutdinov, Zemel, "**Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models**". TACL 2015

Use LSTM for image-language encoding and decoding

Mao, Xu, Yang, Wang, Huang, Yuille. "**Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN)**," ICLR 2015

Use CNN to generate a whole-image feature vector, then input it, at every step, into a multimodal RNN language model to generate the caption.

Xu, Ba, Kiros, Cho, Courville, Salakhutdinov, Zemel, Bengio, 2015. **Show, Attend and Tell: Neural Image Caption Generation with Visual Attention**.

Use CNN to generate a whole-image feature vector, then input it, at every step, into a multimodal RNN language model to generate the caption.

Hill and Korhonen, **2014 Learning Abstract Concept Embeddings from Multi-Modal Data: Since You Probably Can't See What I Mean**

m-DSSM helps pick the global semantically matching caption for a given image

Baseline: a clock tower in front of a building

w/ m-DSSM: a clock tower in the middle of the street

Baseline: a large jetliner sitting on top of a stop sign at an intersection on a city street

w/ m-DSSM: a stop light on a city street

Baseline: a red brick building

w/ m-DSSM: a living room filled with furniture and a flat screen tv sitting on top of a brick building

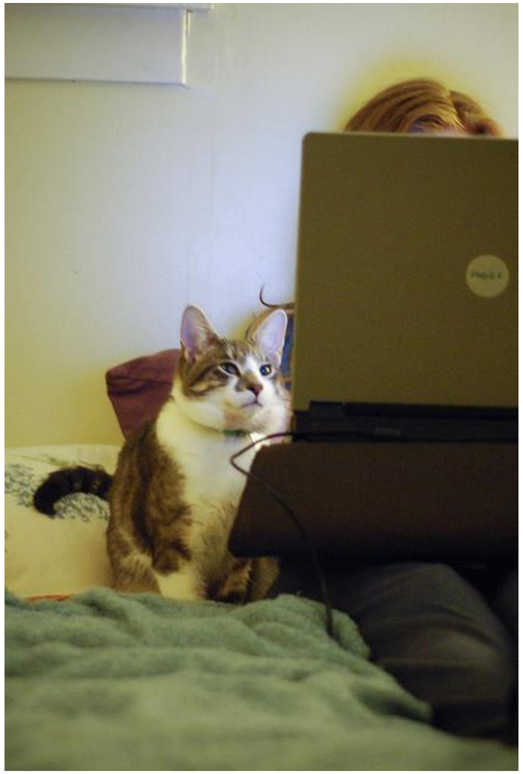
Baseline: a large jetliner sitting on top of a table

w/ m-DSSM: a display in a grocery store filled with lots of food on a table

m-DSSM helps pick the global semantically matching caption for a given image

Baseline: a young man riding a skateboard down a street holding a tennis racquet on a tennis court

w/ m-DSSM: a man riding a skateboard down a street



Baseline: a cat sitting on a table

w/ m-DSSM: a cat sitting on top of a bed

Baseline: a group of people standing in a kitchen

w/ m-DSSM: a group of people posing for a picture

Baseline: two elephants standing next to a baby elephant walking behind a fence

w/ m-DSSM: a baby elephant standing next to a fence

Interpretability

Our system not only generates the caption, but can also interpret it.

Interpretability

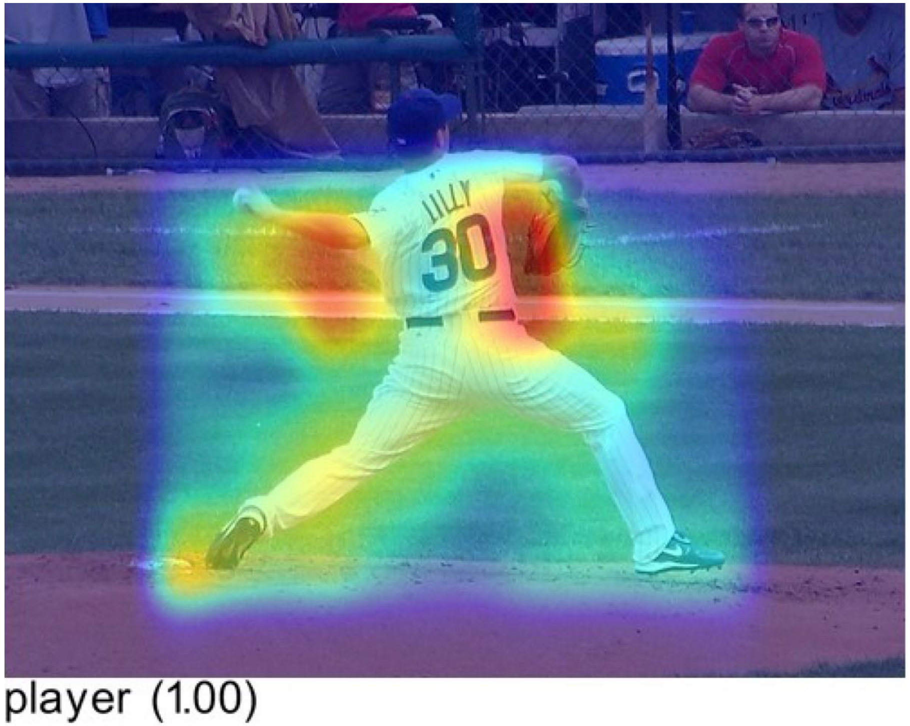
Our system not only generates the caption, but can also interpret it.

Interpretability

a **baseball**

Our system not only generates the caption, but can also interpret it.

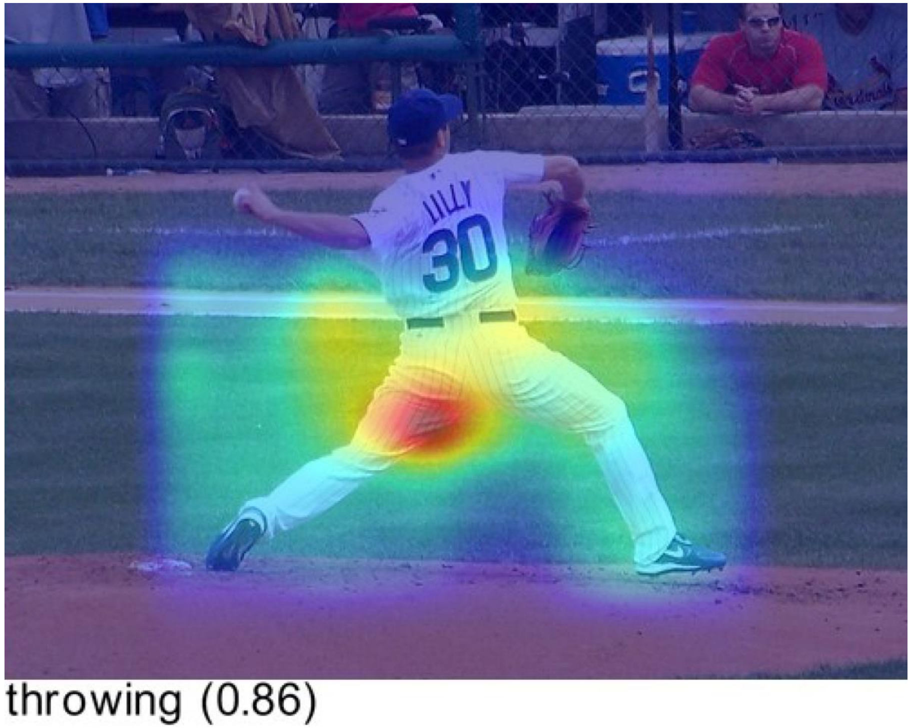
Interpretability



a baseball **player**

Our system not only generates the caption, but can also interpret it.

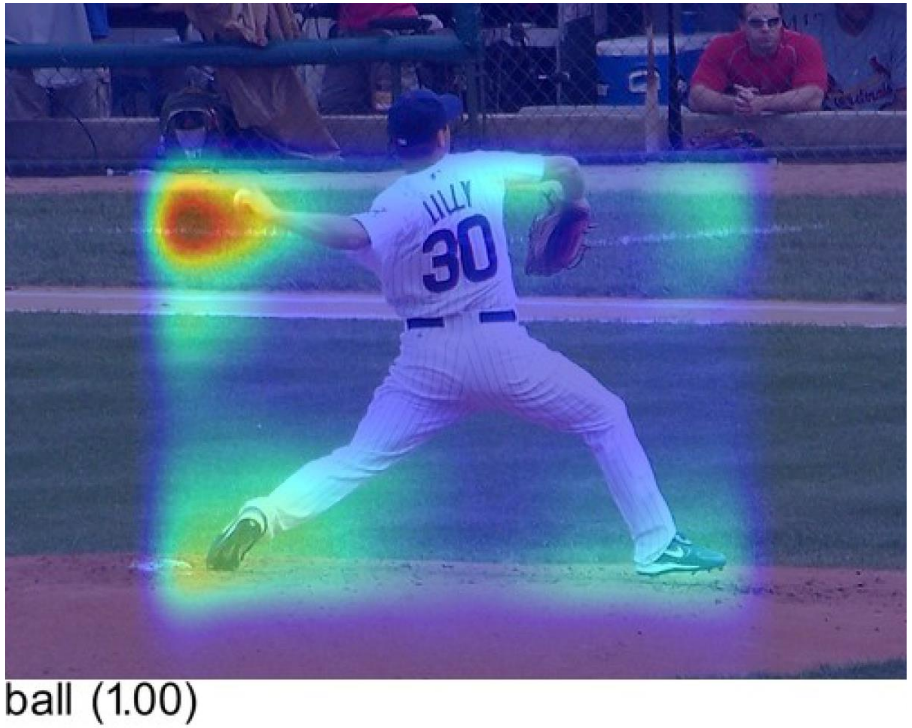
Interpretability



a baseball player **throwing**

Our system not only generates the caption, but can also interpret it.

Interpretability



a baseball player throwing a **ball**

Our system not only generates the caption, but can also interpret it.

Interpretability

Our system not only generates the caption, but can also interpret it.

Interpretability

Our system not only generates the caption, but can also interpret it.

Interpretability

man (0.93)

a **man**

Our system not only generates the caption, but can also interpret it.

Interpretability

a man **sitting**

Our system not only generates the caption, but can also interpret it.

Interpretability

a man sitting in a **couch**

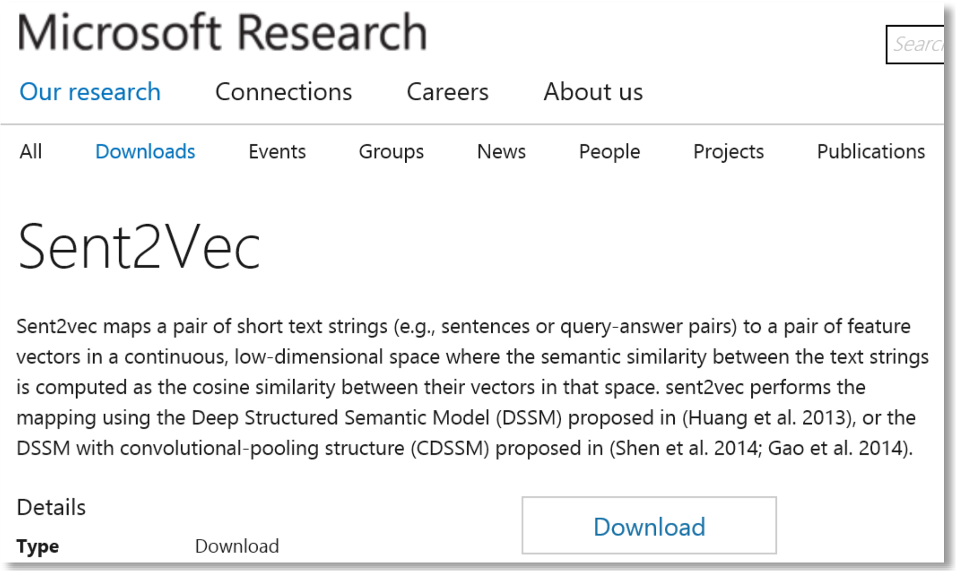
Our system not only generates the caption, but can also interpret it.

Interpretability

dog (1.00)

a man sitting in a couch with a **dog**

Interim summary



Sent2Vec

Sent2vec maps a pair of short text strings (e.g., sentences or query-answer pairs) to a pair of feature vectors in a continuous, low-dimensional space where the semantic similarity between the text strings is computed as the cosine similarity between their vectors in that space. sent2vec performs the mapping using the Deep Structured Semantic Model (DSSM) proposed in (Huang et al. 2013), or the DSSM with convolutional-pooling structure (CDSSM) proposed in (Shen et al. 2014; Gao et al. 2014).

Details

Type Download

Learn Sent2Vec by DSSM

- similarity driven deep semantic model
- superior performance in a range of NL tasks

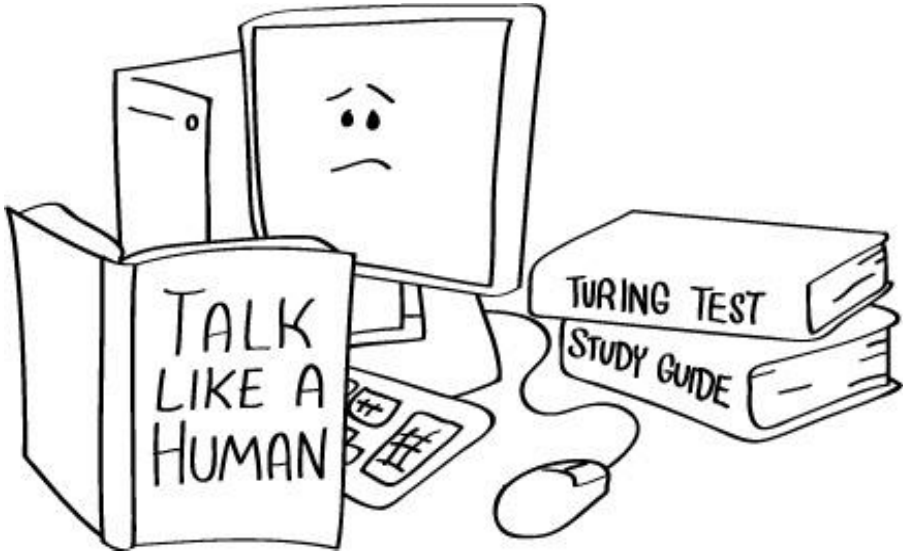
Tool kit available online: http://aka.ms/sent2vec/

Part IV

Natural Language Understanding

Natural Language Understanding

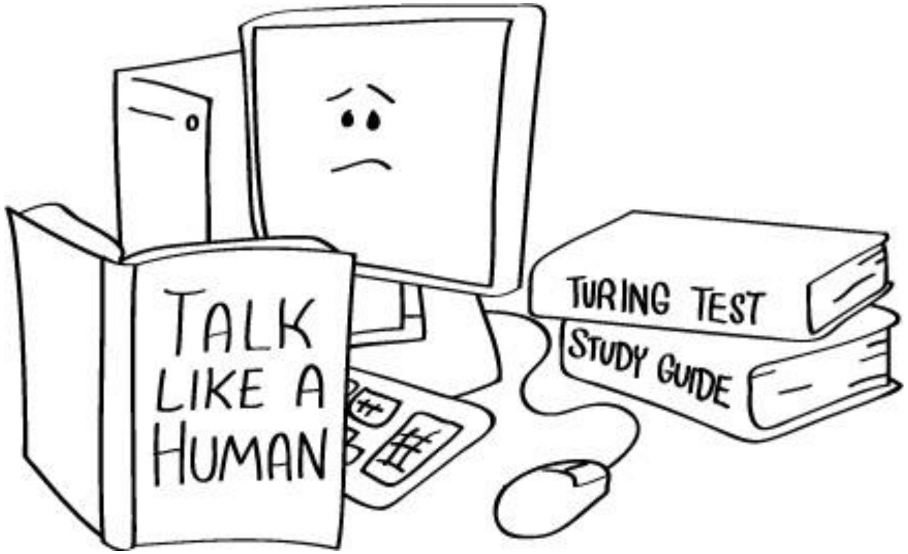
- Build an intelligent system that can interact with human using natural language
- Research challenge
  - **Meaning representation of text**
  - **Support useful inferential tasks**



http://csunplugged.org/turing-test

Natural Language Understanding

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



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]

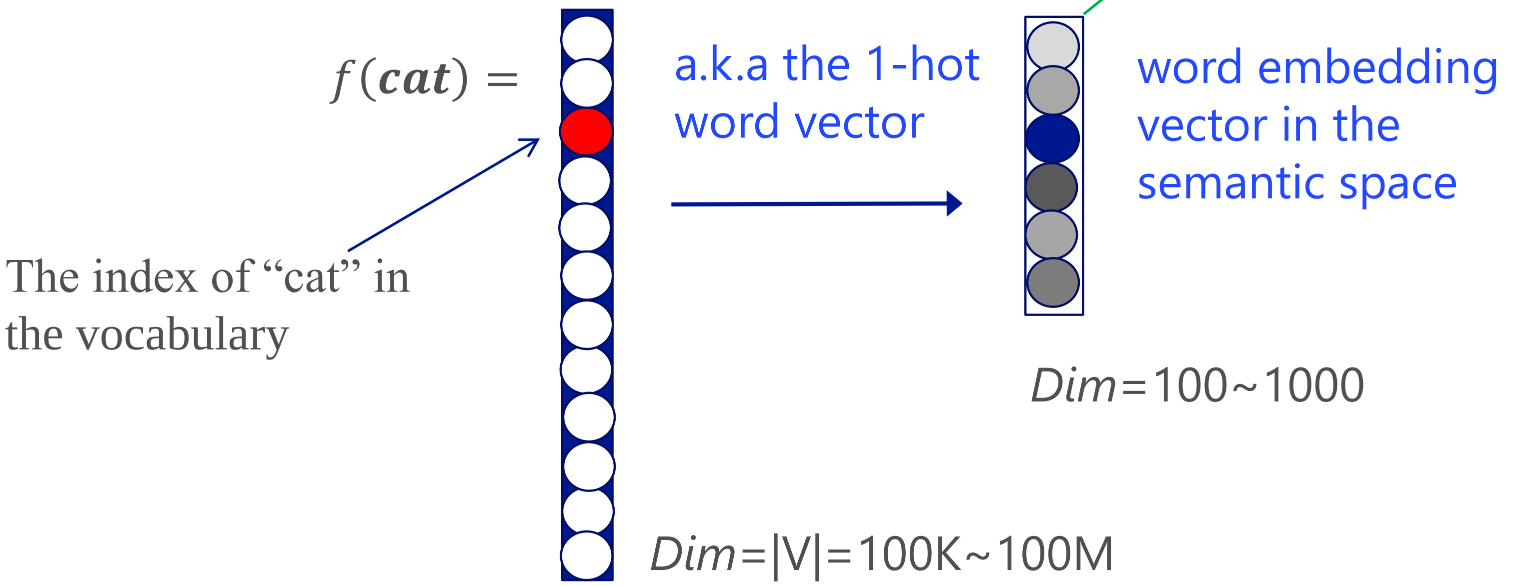
- Encode term co-occurrence information
- Measure semantic similarity well

Semantic Embedding

Project raw text into a continuous semantic space

e.g., word embedding

Captures the word meaning in a semantic space



$f(\boldsymbol{cat}) =$

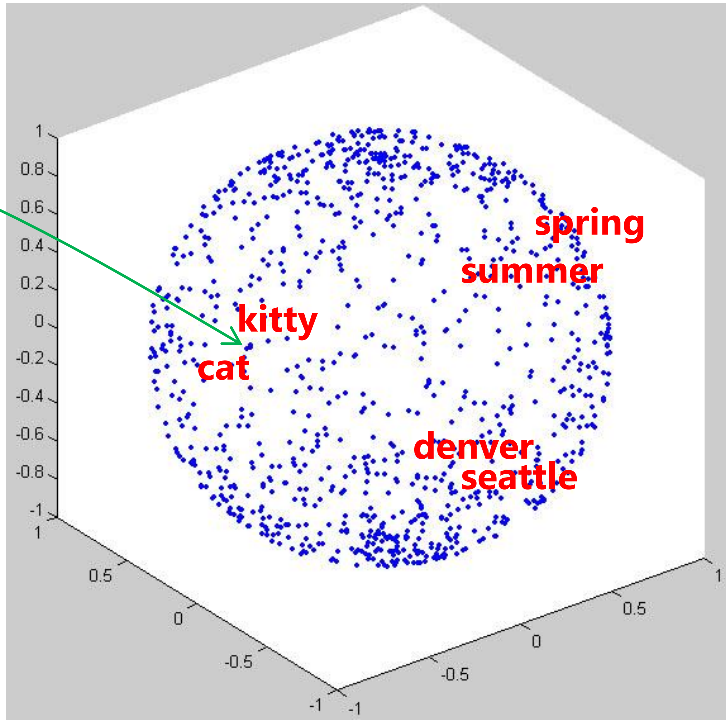
The index of “cat” in the vocabulary

a.k.a the 1-hot word vector

word embedding vector in the semantic space

Dim=100~1000

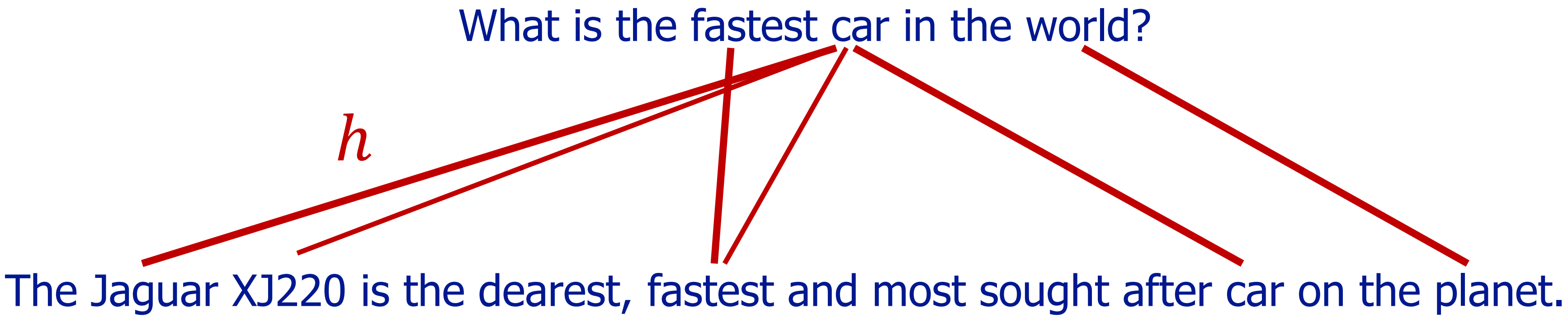
Dim=|V|=100K~100M



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

Why is Word Embedding Useful?

- Lexical semantics – semantic word similarity
  - Used as features in many NLP applications
  - e.g., Question/Sentence matching [Yih+ ACL-13; Jansen+ ACL-14]



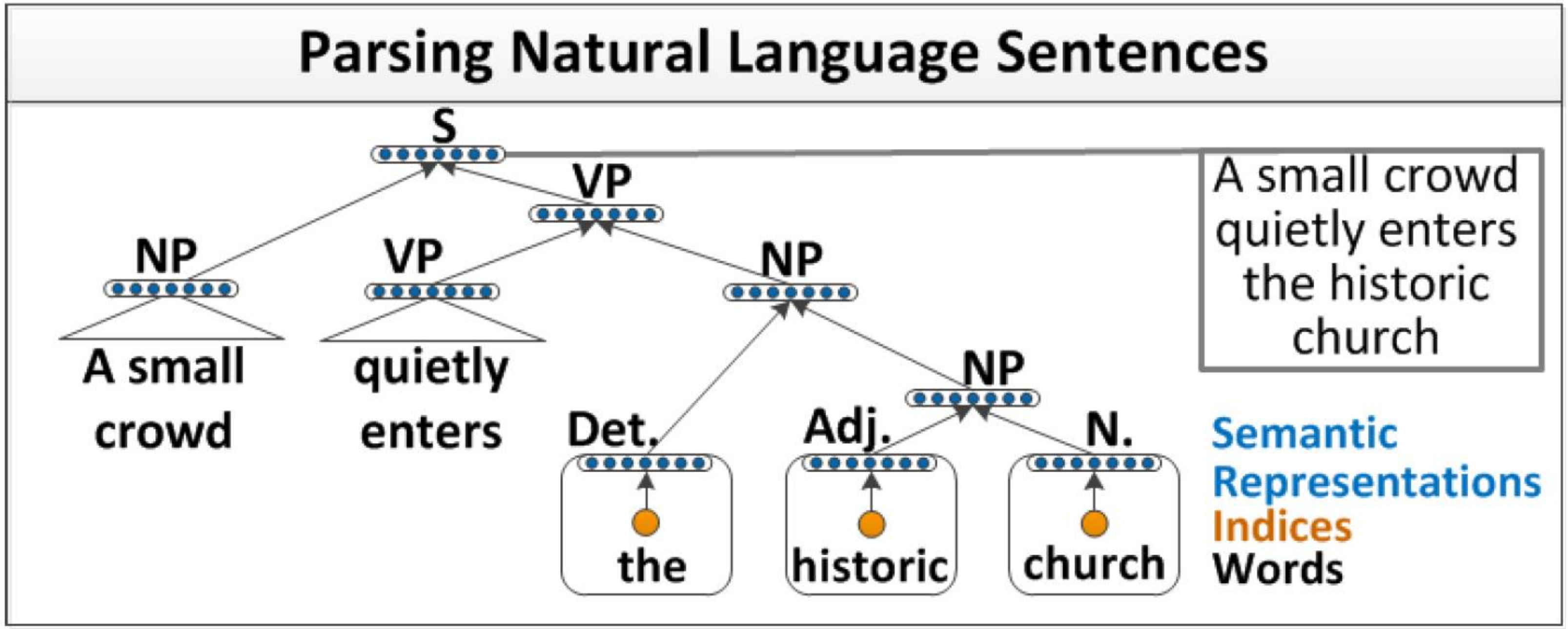
What is the fastest car in the world?

The Jaguar XJ220 is the dearest, fastest and most sought after car on the planet.

- Simple semantic representation of text
  - Represent longer text using average of the word vectors
  - e.g., entity [Socher+ NIPS-13], question [Berant&Liang ACL-14]

Why is Word Embedding Useful? (Cont'd)

- "Pre-training" of a neural-network model
  - Take word vectors trained on a general corpus as input
  - e.g., Recursive NN for parsing [Socher+ ICML-11]



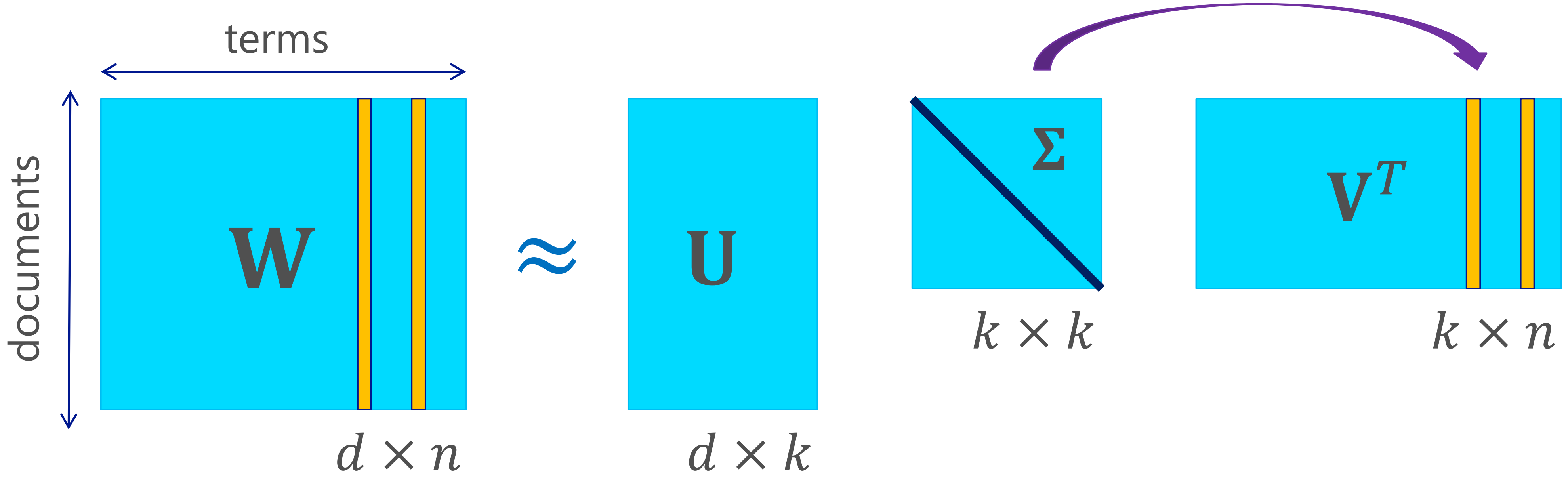
Roadmap – Continuous Word Representations

- Samples of word embedding models
  - Latent Semantic Analysis (LSA), Recurrent Neural Networks
  - SENNA, CBOW/Skip-gram, DSSM

- Evaluation
  - Semantic word similarity
  - Relational similarity (word analogy)

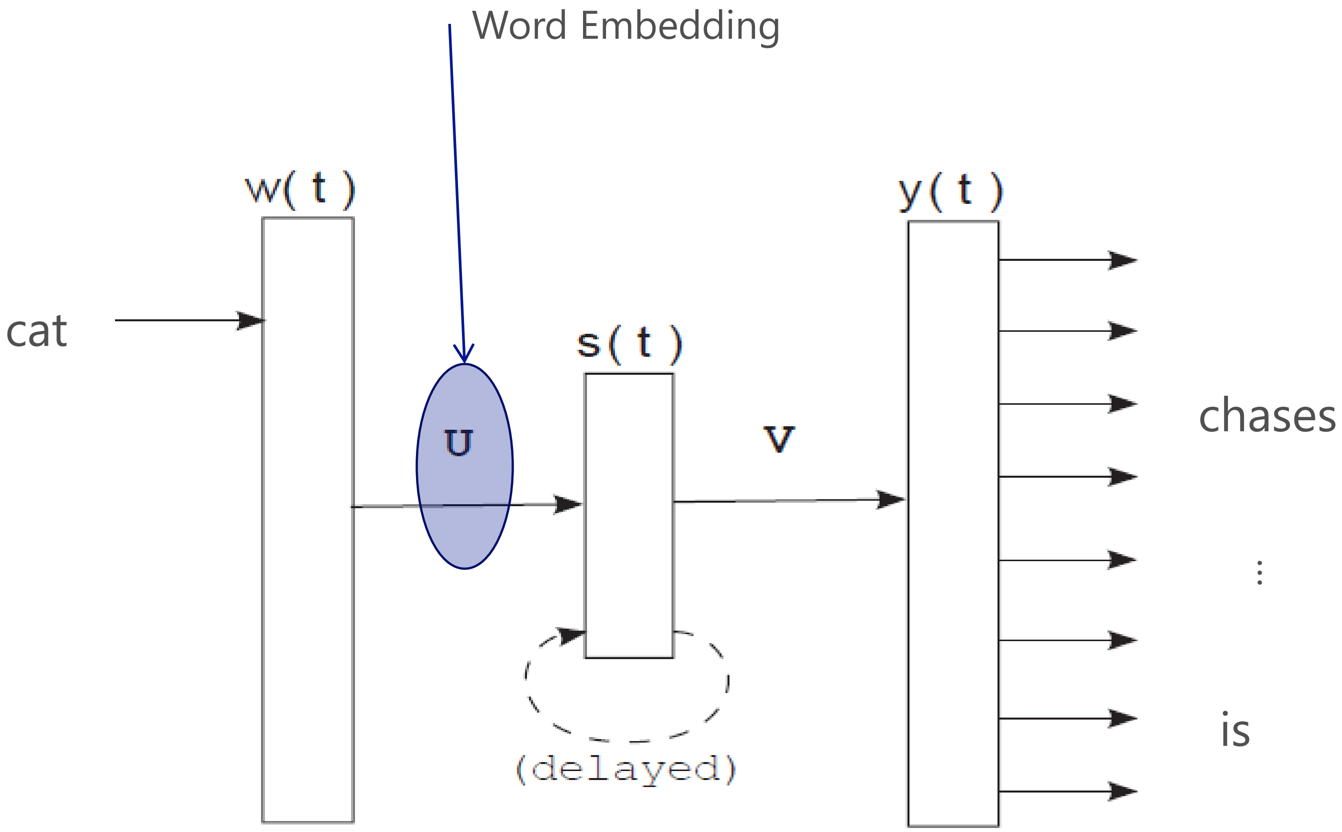
- Related work
  - Model different word relations
  - Other word embedding models

Latent Semantic Analysis



- 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 $\mathbf{\Sigma V}^T$

RNN-LM Word Embedding



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

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

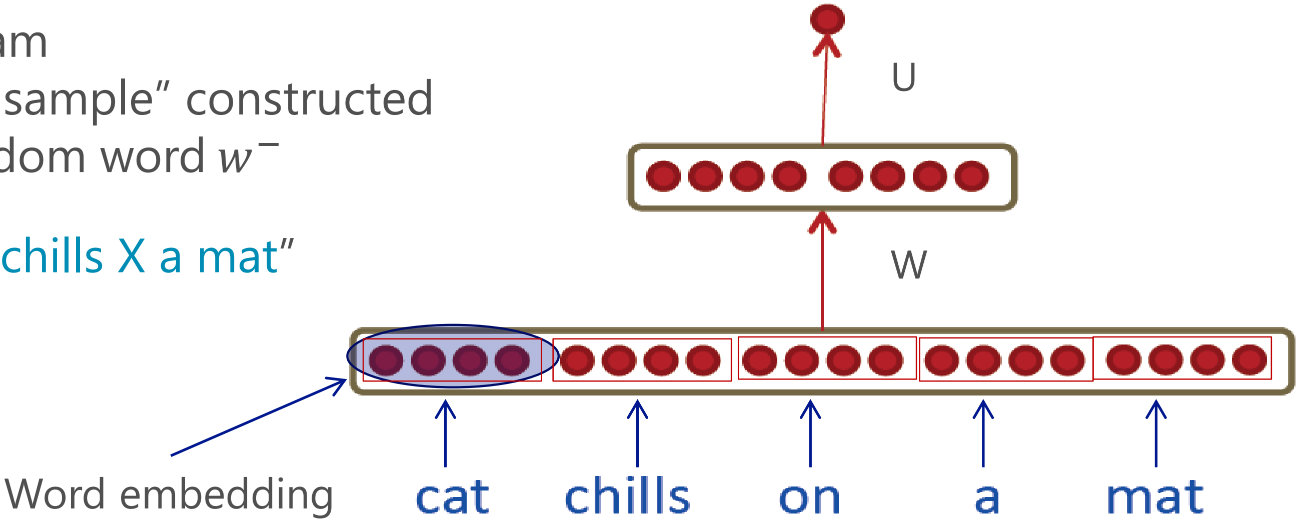
$$S^+ = Score(w_1, w_2, w_3, w_4, w_5)$$
$$S^- = Score(w_1, w_2, w^-, w_4, w_5)$$

And

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

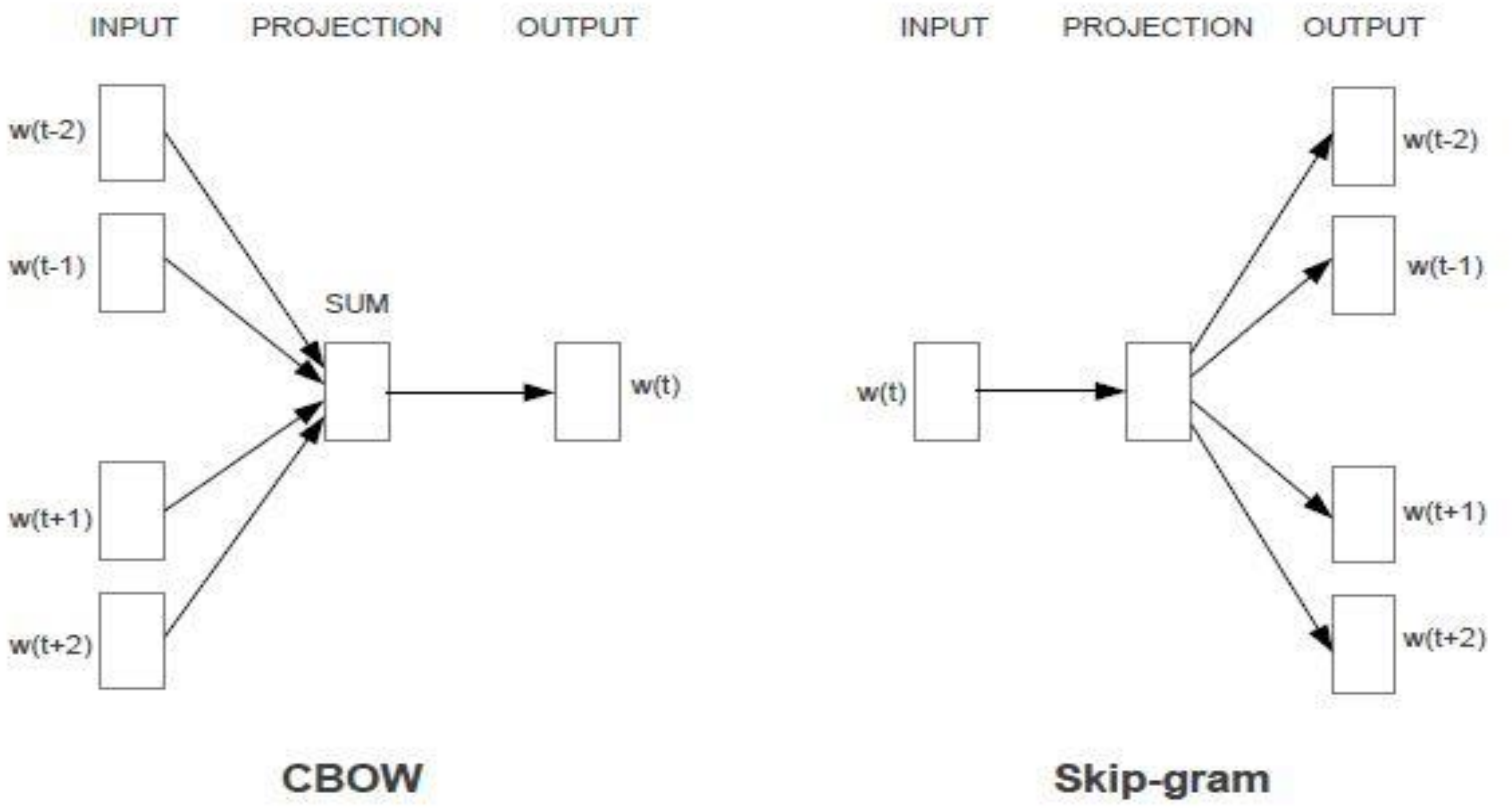
$< w_1, w_2, w^-, w_4, w_5 >$ is a "negative sample" constructed by replacing the word w_3 with a random word w^-

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



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

CBOW/Skip-gram Word Embeddings



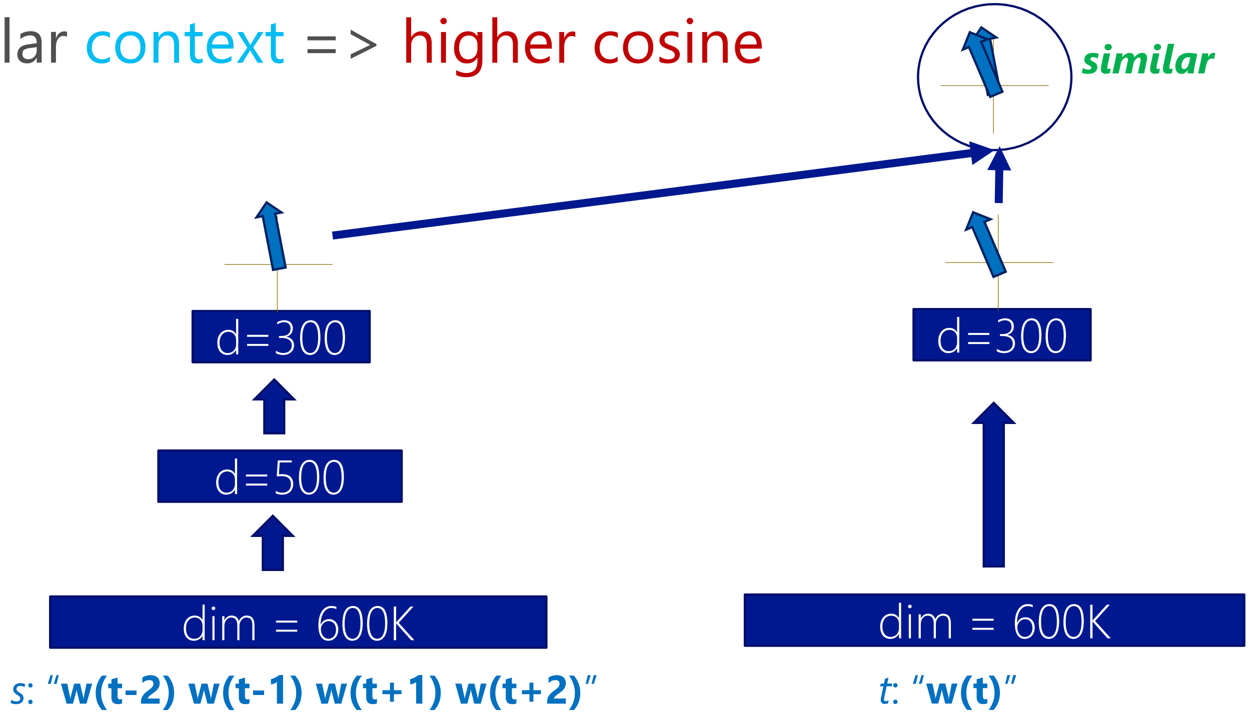
Continuous Bag-of-Words

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

DSSM: Learning Word 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)



s: "**w(t-2) w(t-1) w(t+1) w(t+2)**"

t: "**w(t)**"

[Song, He, Gao, Deng, 2014]

Evaluation: Semantic Word Similarity

- Data: word pairs with human judgment (e.g., WS-353, RG-65)

Word 1	Word 2	Human Score (mean)
midday	noon	9.3
tiger	jaguar	8.0
cup	food	5.0
forest	graveyard	1.9
…	…	…

- Correlation of the *ranking* of word similarity and human judgment
  - **Spearman's rank correlation coefficient** ρ
- Word embedding models individually usually do not achieve the state-of-the-art results (cf. ACL Wiki Similarity (State-of-the-art))

Evaluation: Relational Similarity (Word Analogy)

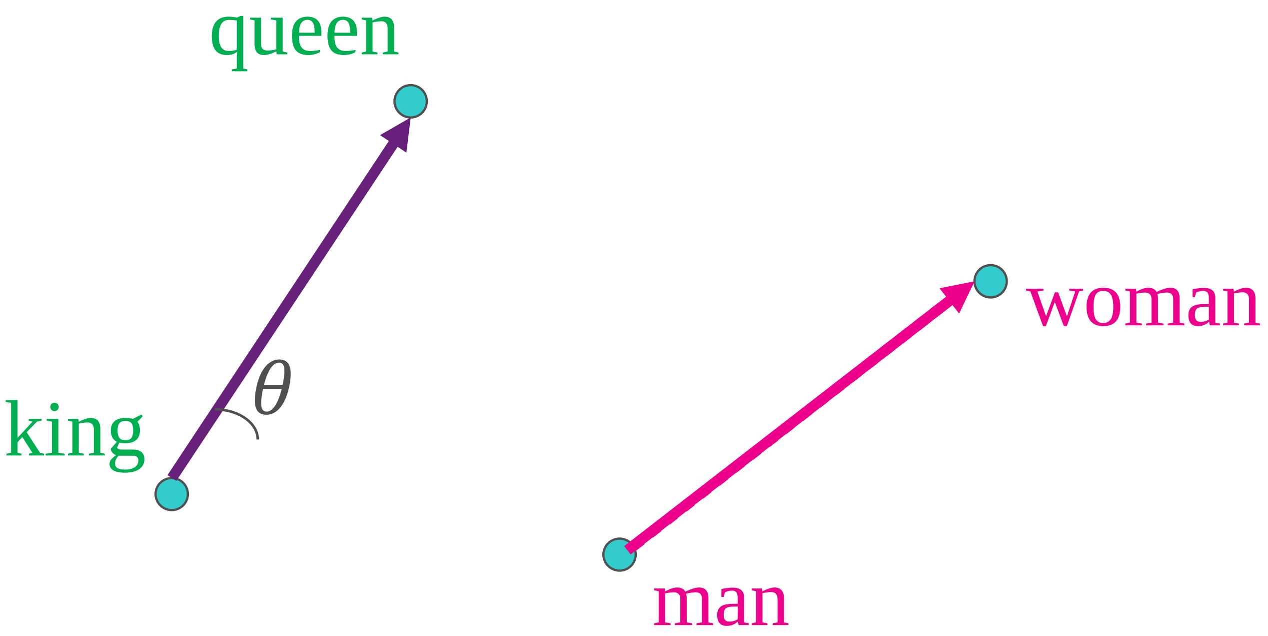
king : queen $\overset{?}{=}$ man : woman

- 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

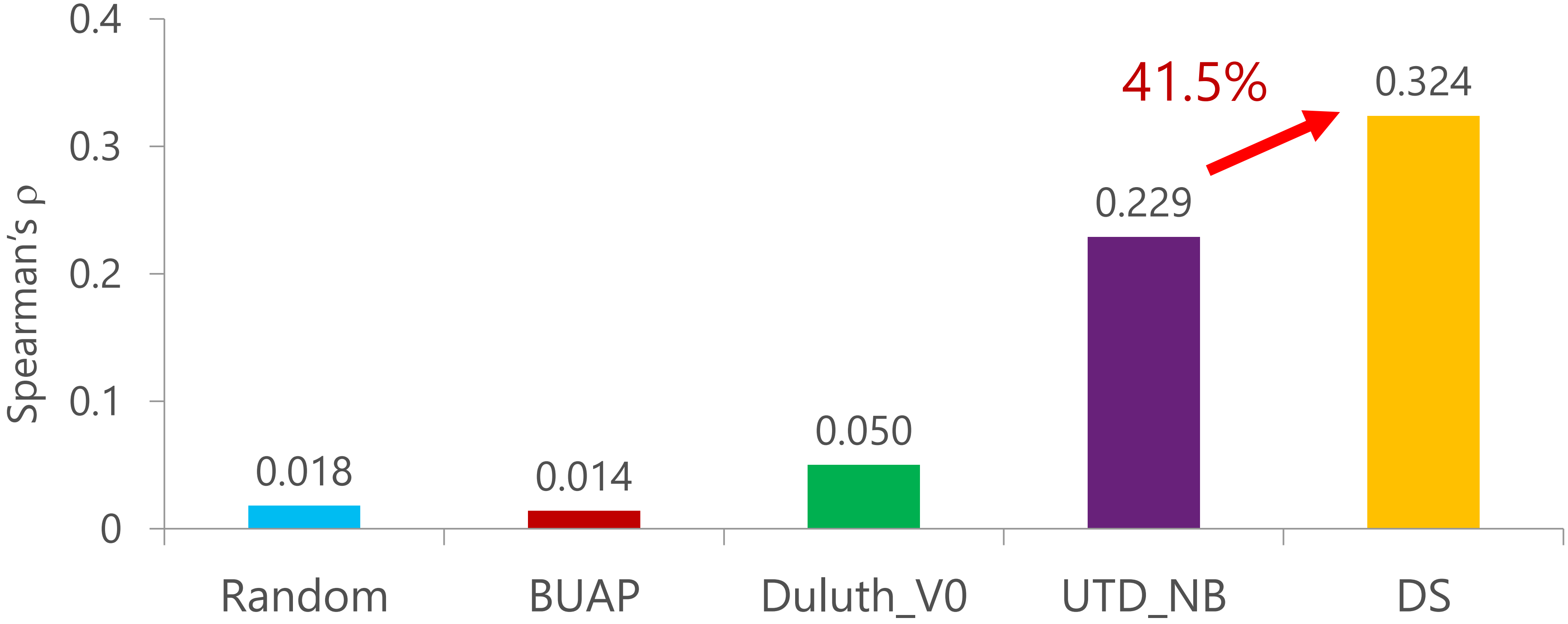
- Word embedding taken from recurrent neural network language model (RNN-LM) [Mikolov 2011]



- 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

$$\arg\max_{x}(\cos(x, king - man + woman)) =$$
$$\arg\max_{x}(\underset{\Uparrow}{\cos(x, king)} - \underset{\Downarrow}{\cos(x, man)} + \underset{\Uparrow}{\cos(x, woman)})$$

Related Work – Model Different Word Relations

Tomorrow
will be rainy.

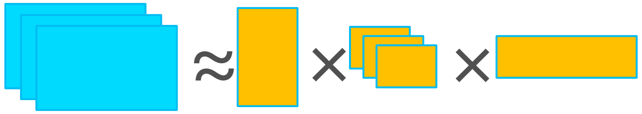
Tomorrow
will be sunny.

~~$similar$(rainy, sunny)?~~

$antonym$(rainy, sunny)?

- Multi-Relational Latent Semantic Analysis [Chang+ EMNLP-04]

f_{rel}(●, ●)



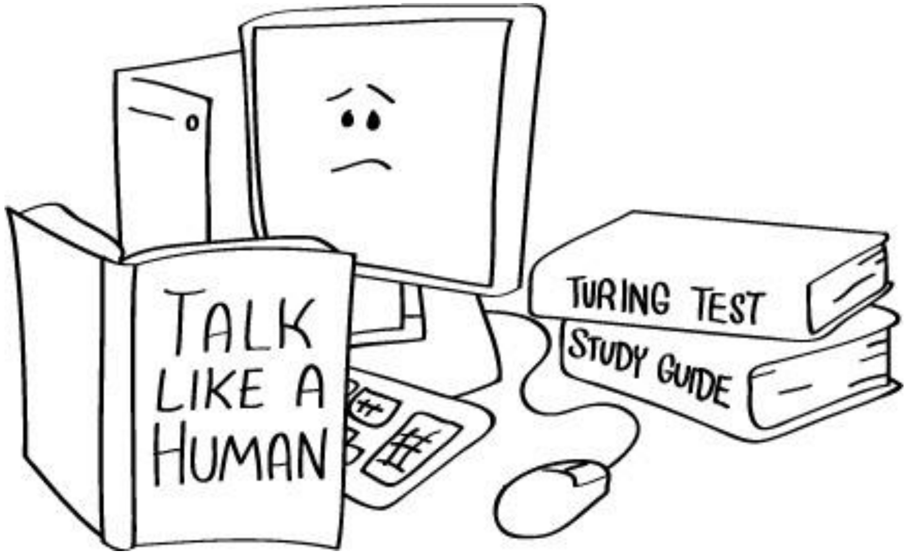
Related Work – Word Embedding Models

- Other word embedding models
  - GloVe [Pennington+ EMNLP-14], [Wang+ EMNLP-14], [Bian+ ECML/PKDD-14], [Xu+, CIKM-14], [Faruqui+ NAACL-15], [Yogatama+ ICML-15], [Faruqui+ ACL-15]

- Analysis of Word2Vec and Directional Similarity
  - Linguistic Regularities in Sparse and Explicit Word Representations [Levy & Goldberg CoNLL-14]
  - Neural Word Embedding as Implicit Matrix Factorization [Levy & Goldberg NIPS-14]
  - Yoav Goldberg's blog on comparing word embedding models (invited speaker of CVSC-2015)

Natural Language Understanding

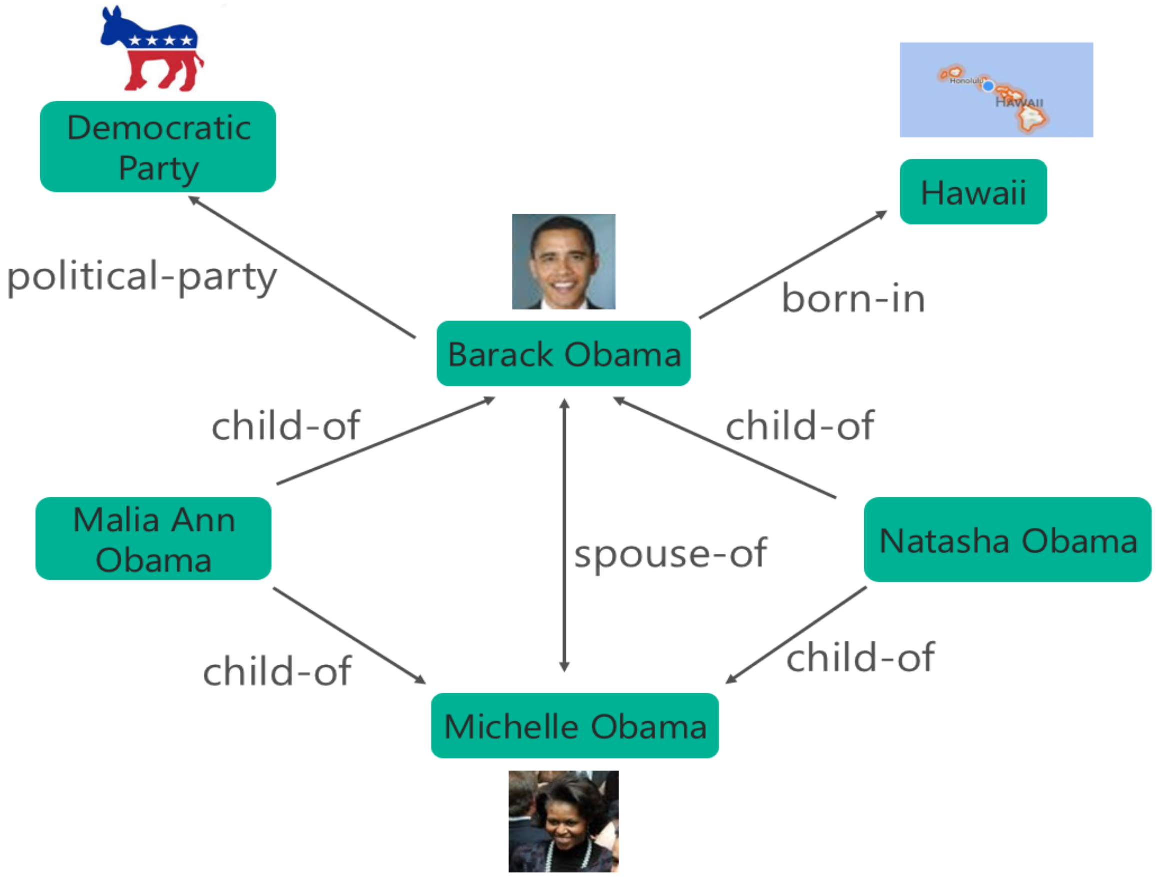
- Continuous Word Representations & Lexical Semantics
- Knowledge Base Embedding
- Semantic Parsing & Question Answering



http://csunplugged.org/turing-test

Knowledge Base

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



Democratic Party

Hawaii

political-party

born-in

Barack Obama

child-of

child-of

Malia Ann Obama

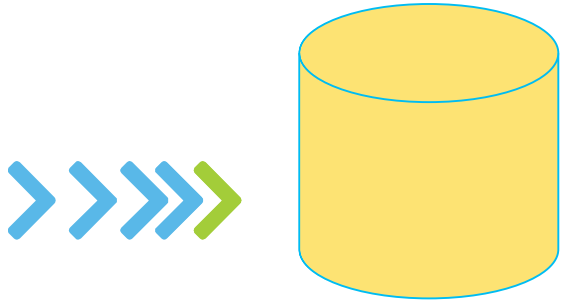
spouse-of

Natasha Obama

child-of

child-of

Michelle Obama



Freebase
DBpedia
YAGO
NELL
OpenIE/ReVerb

Current KB Applications in NLP & IR

- Question Answering

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

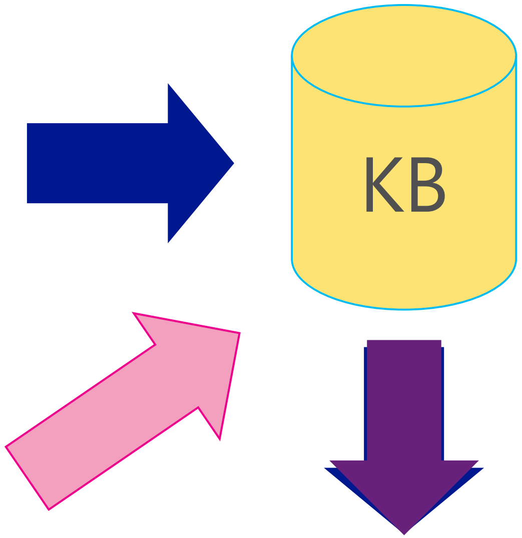
  $\lambda x.parent(Obama, x) \wedge gender(x, Female)$

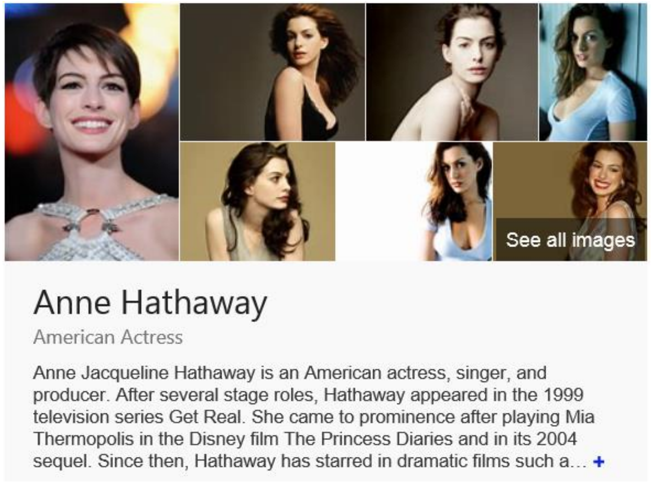
- Information Extraction
  - *"Hathaway was born in Brooklyn, New York."*

  $bornIn(Hathaway, Brooklyn)$

  $contains(New\ York, Brooklyn)$

- Web Search
  - Identify entities and relationships in queries





Anne Hathaway

American Actress

Anne Jacqueline Hathaway is an American actress, singer, and producer. After several stage roles, Hathaway appeared in the 1999 television series Get Real. She came to prominence after playing Mia Thermopolis in the Disney film The Princess Diaries and in its 2004 sequel. Since then, Hathaway has starred in dramatic films such a… +

Reasoning with Knowledge Base

- Knowledge base is never complete!
  - Predict new facts: $Nationality(Natasha\ Obama, ?)$
  - Mine rules: $BornInCity(a,b) \wedge CityInCountry(b,c) \Rightarrow Nationality(a,c)$

- 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

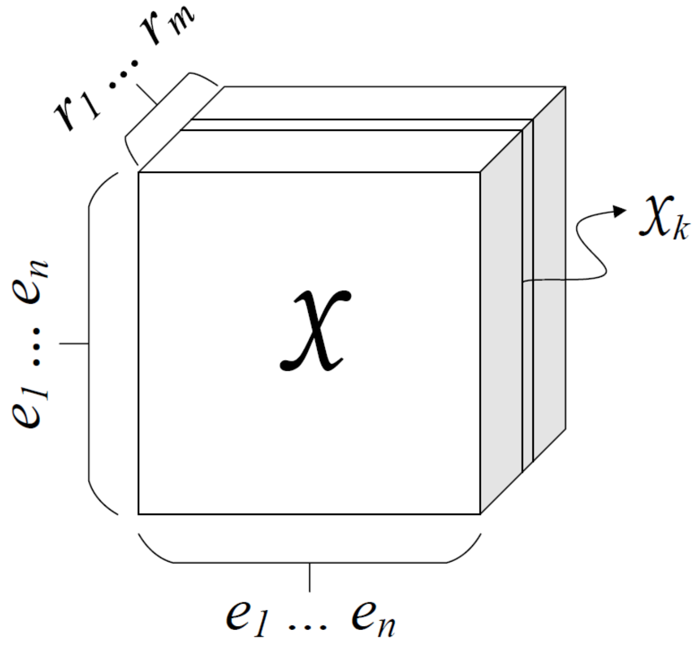
- Each entity in a KB is represented by an $\boldsymbol{R}^{d}$ vector
- Predict whether (e_1, r, e_2) is true by $f_r\left(\boldsymbol{v}_{e_1}, \boldsymbol{v}_{e_2}\right)$

- 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]

Tensor Decomposition:
Knowledge Base Representation (1/2)

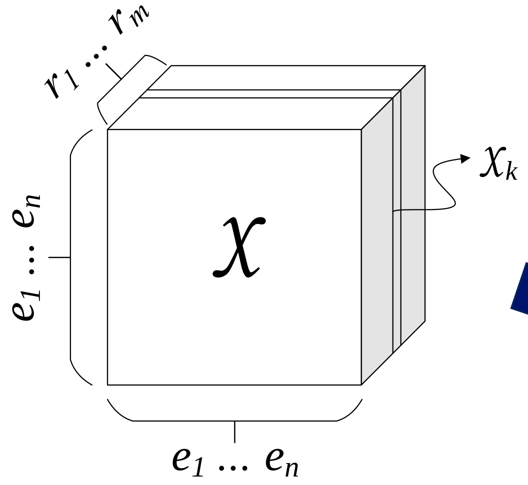
- Collection of **subj-pred-obj** triples – (e_1, r, e_2)

Subject	Predicate	Object
Obama	BornIn	Hawaii
Bill Gates	Nationality	USA
Bill Clinton	SpouseOf	Hillary Clinton
Satya Nadella	WorkAt	Microsoft
…	…	…

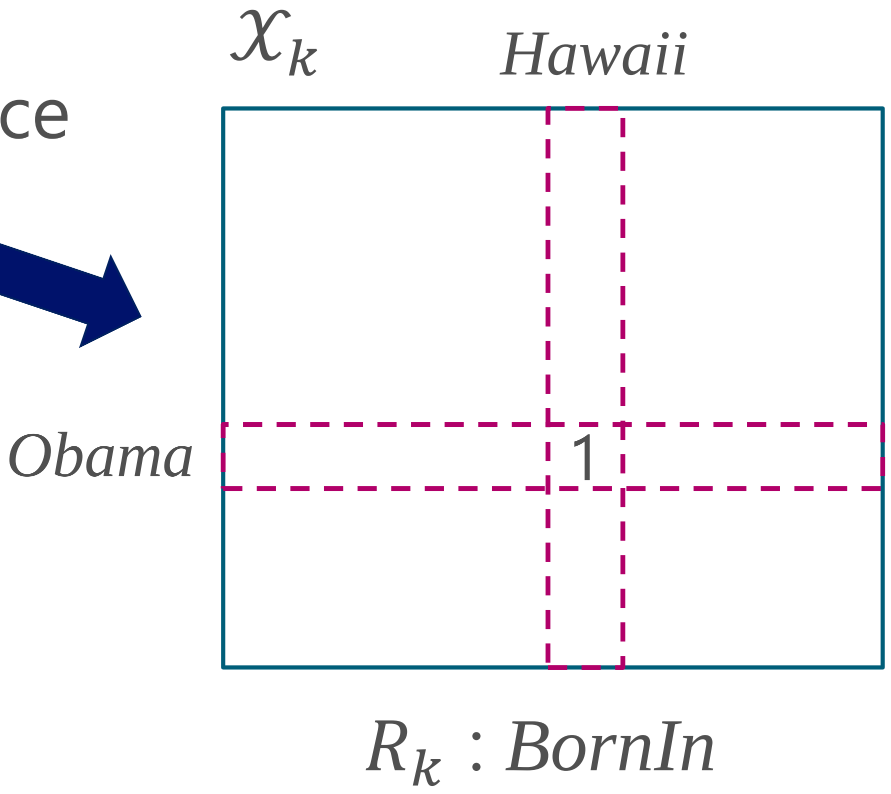


n: # entities, m: # relations

Tensor Decomposition: Knowledge Base Representation (2/2)



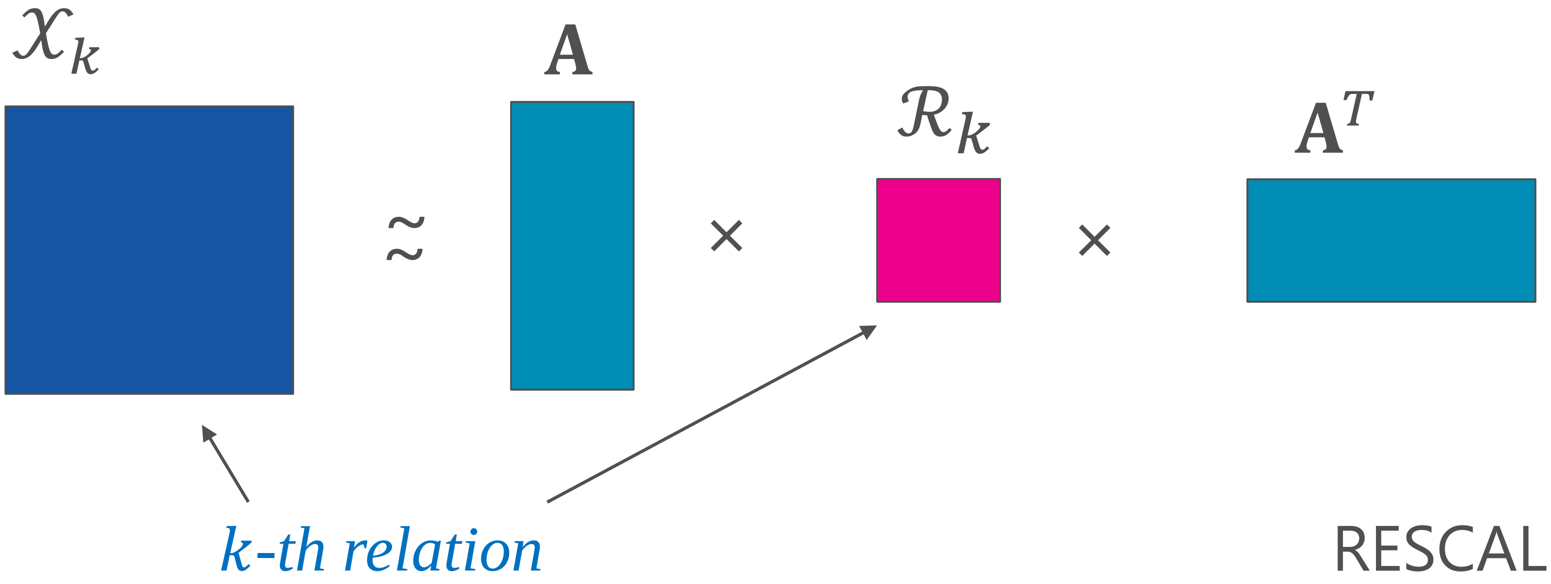
k-th slice



A zero entry means either:
- Incorrect (*false*)
- Unknown

Tensor Decomposition Objective

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

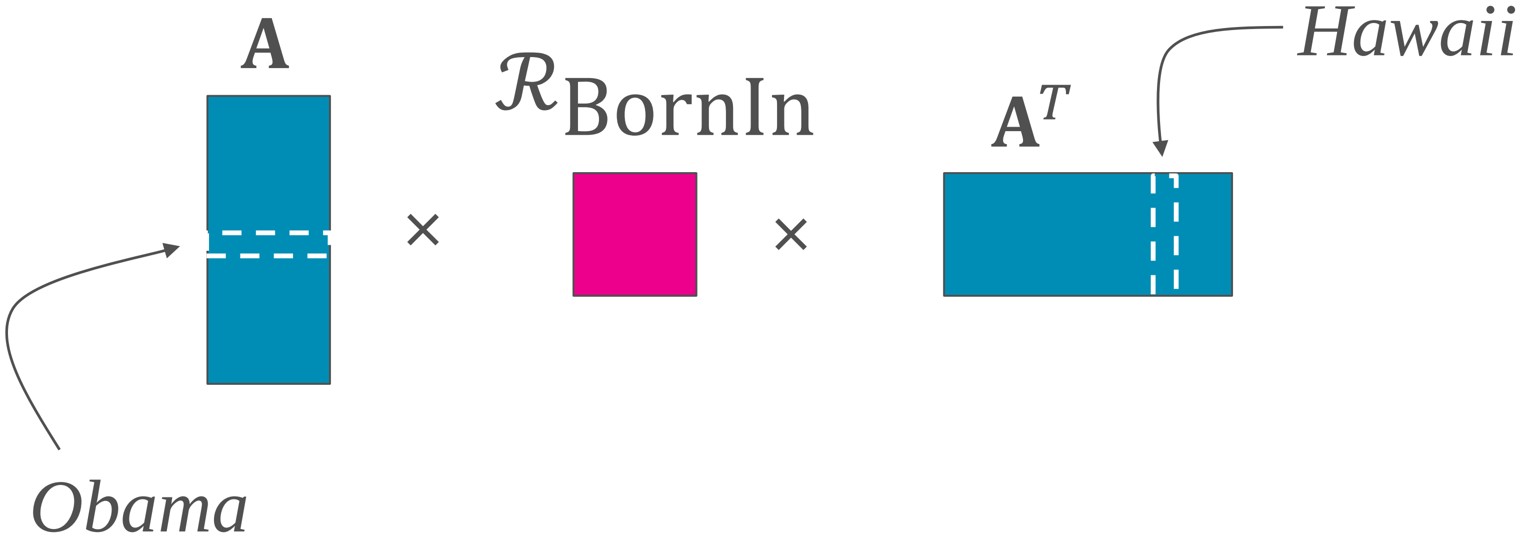


RESCAL [Nickel+, ICML-11]

Measure the Degree of a Relationship

$$f_{\text{BornIn}}(\text{Obama}, \text{Hawaii})$$

$$= \mathbf{A}_{\text{Obama},:} \mathcal{R}_{\text{BornIn}} \mathbf{A}^{\text{T}}_{\text{Hawaii},:}$$

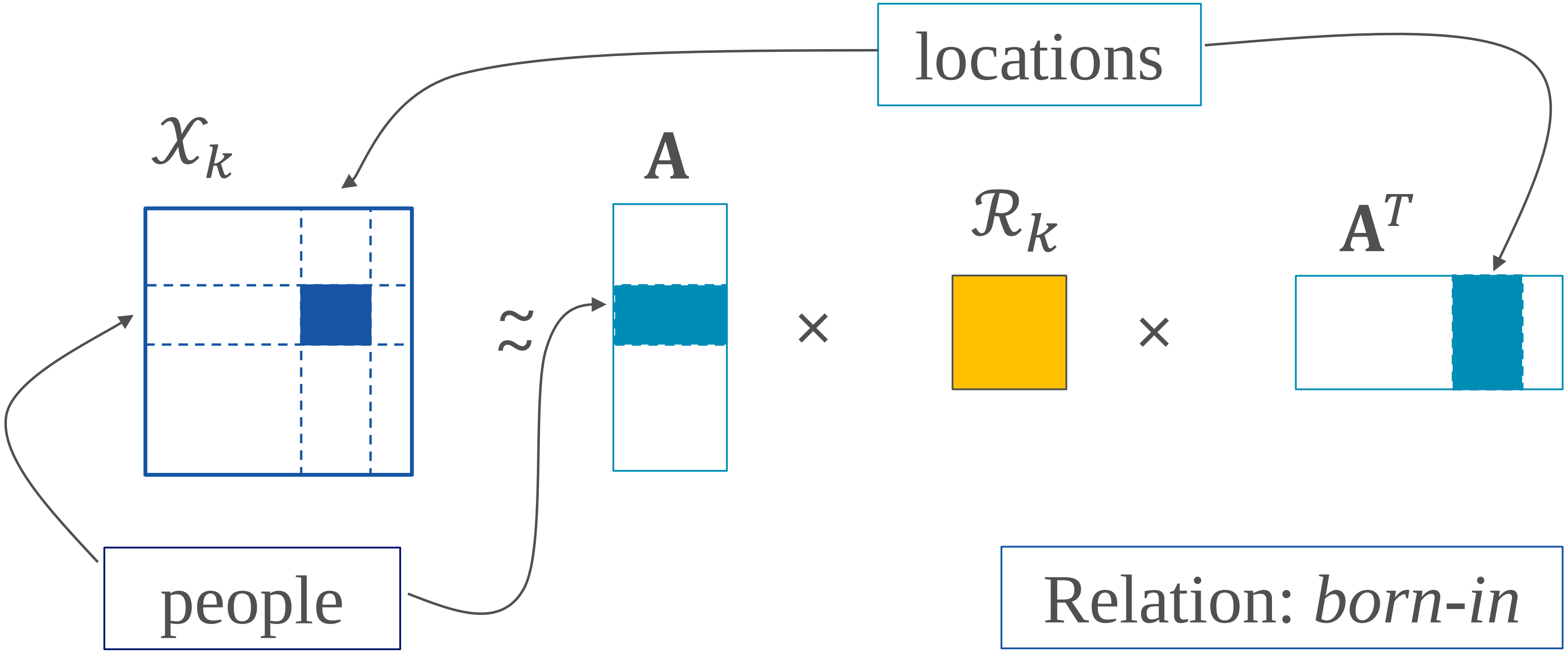


Typed Tensor Decomposition – TRESCAL
[Chang+ EMNLP-14]

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

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

$$\mathbf{vec}(\mathcal{R}_k)$$

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

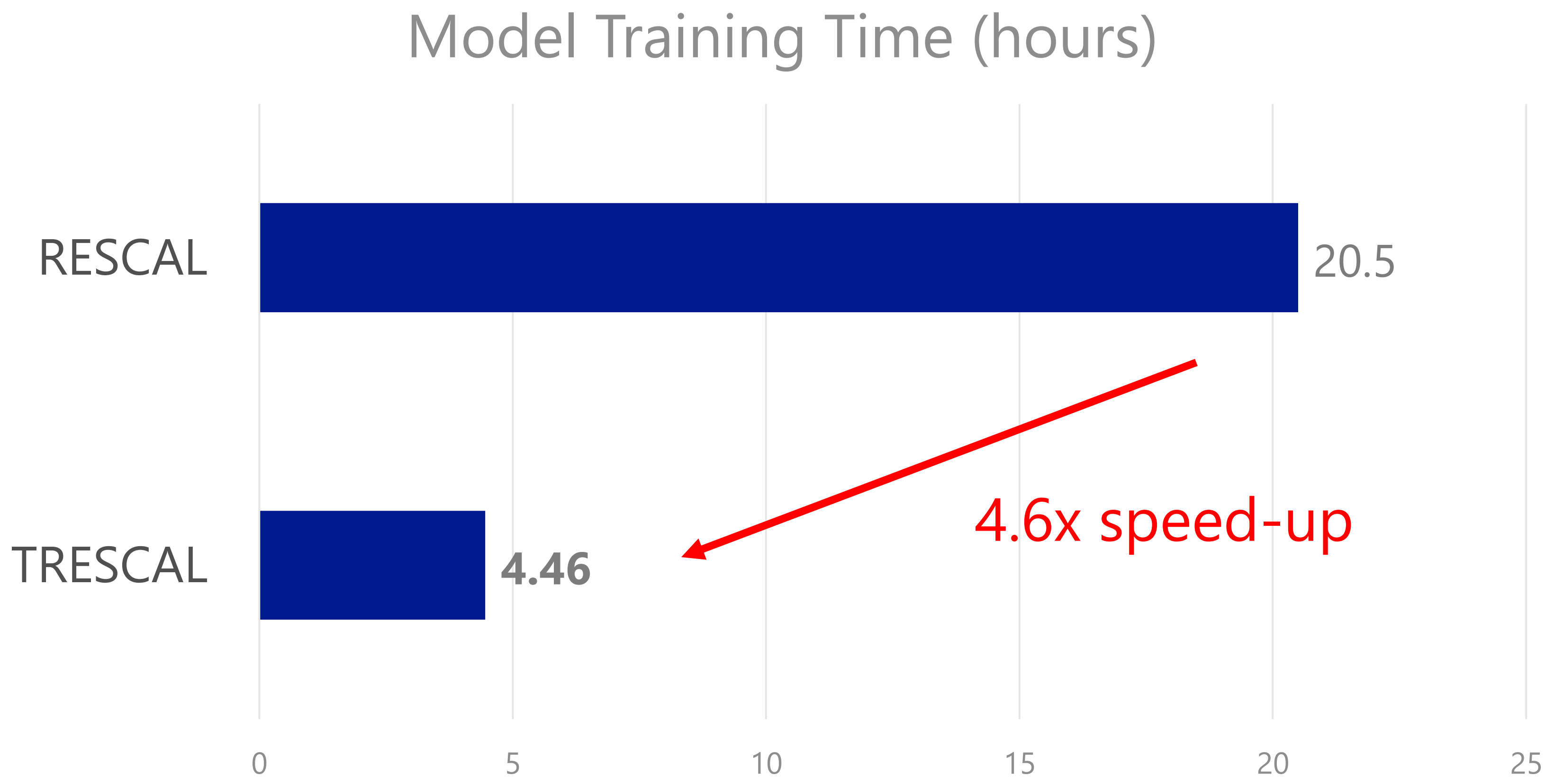
Experiments – KB Completion

- 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

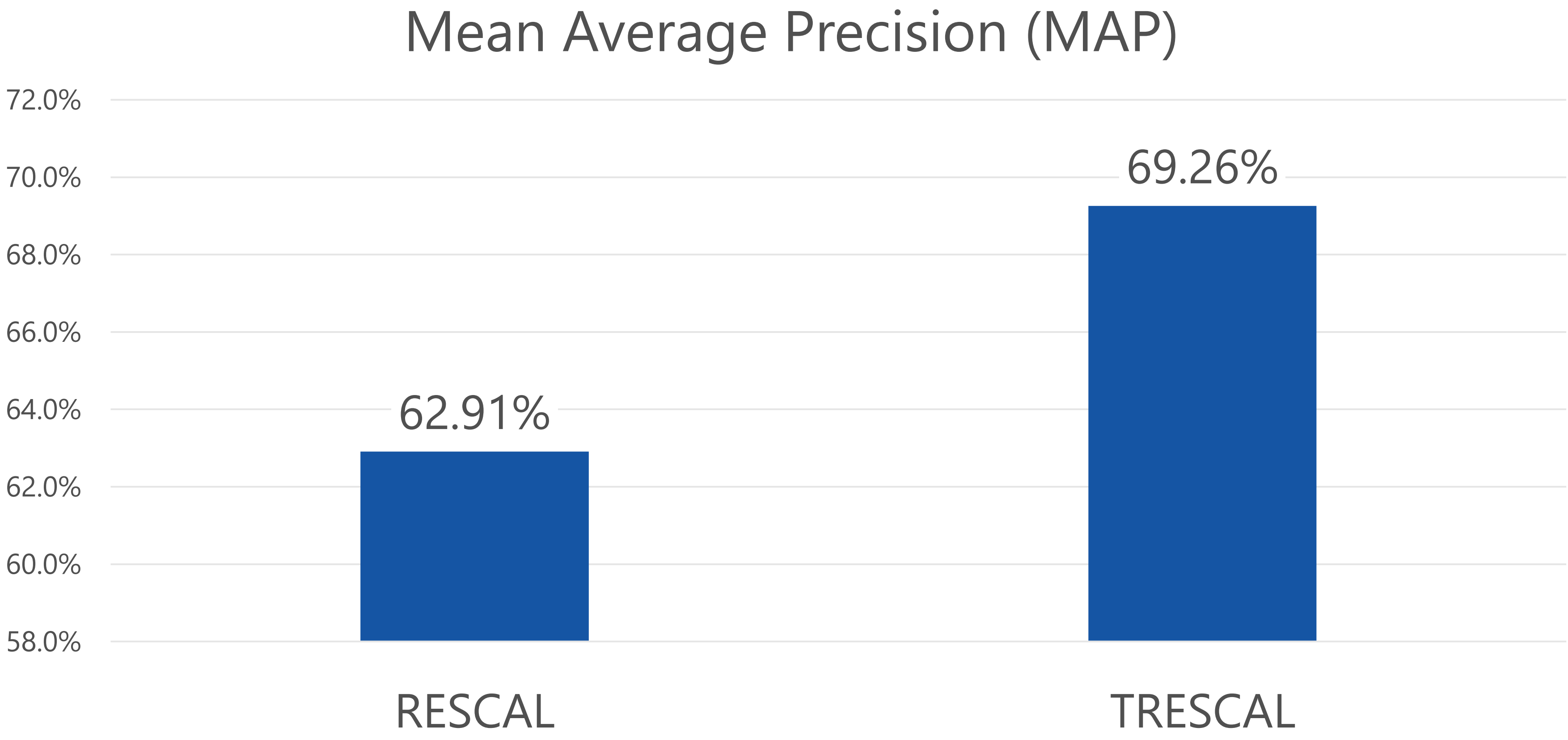
Training Time Reduction



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

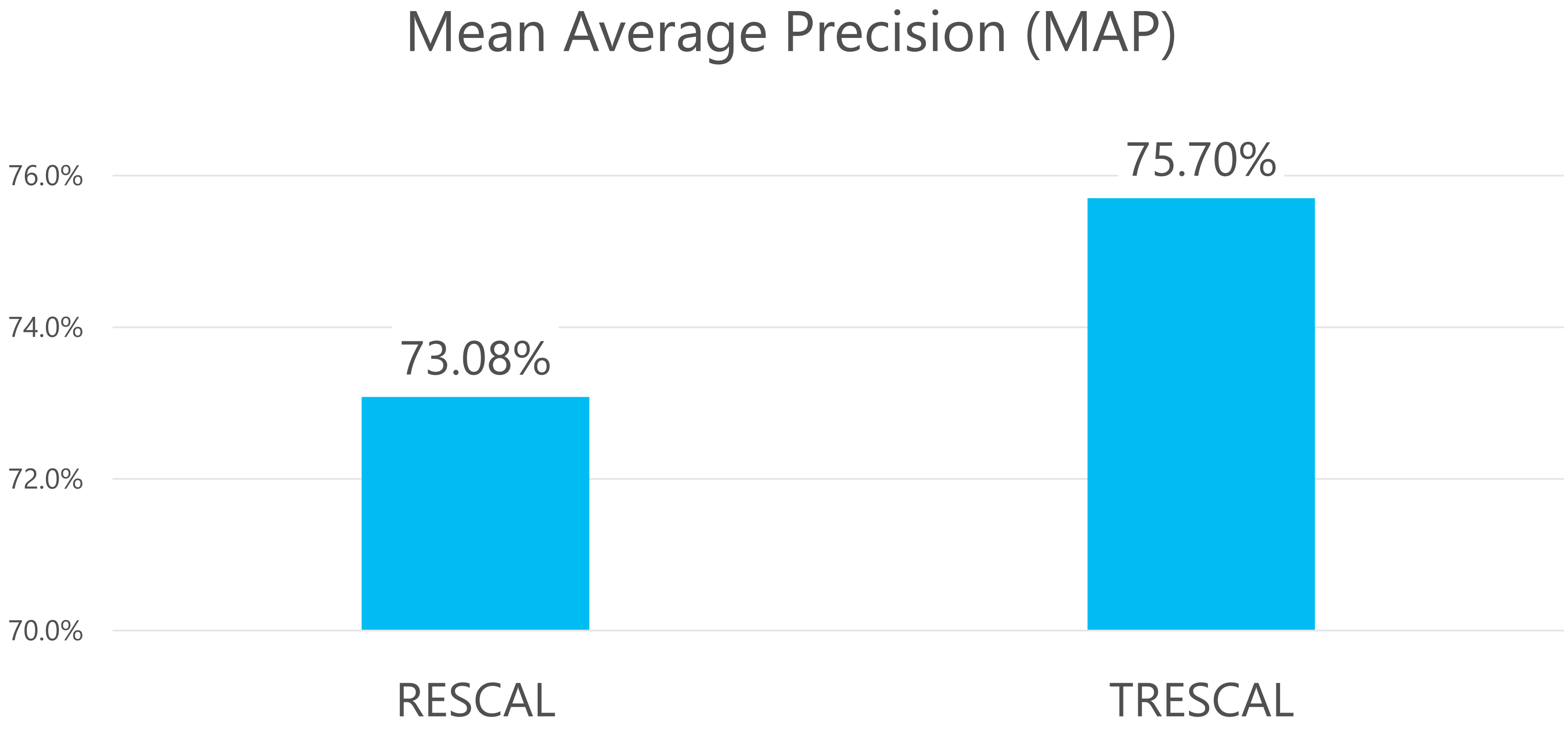
Entity Retrieval $(e_i, r_k, ?)$

- One positive entity with 100 negative entities

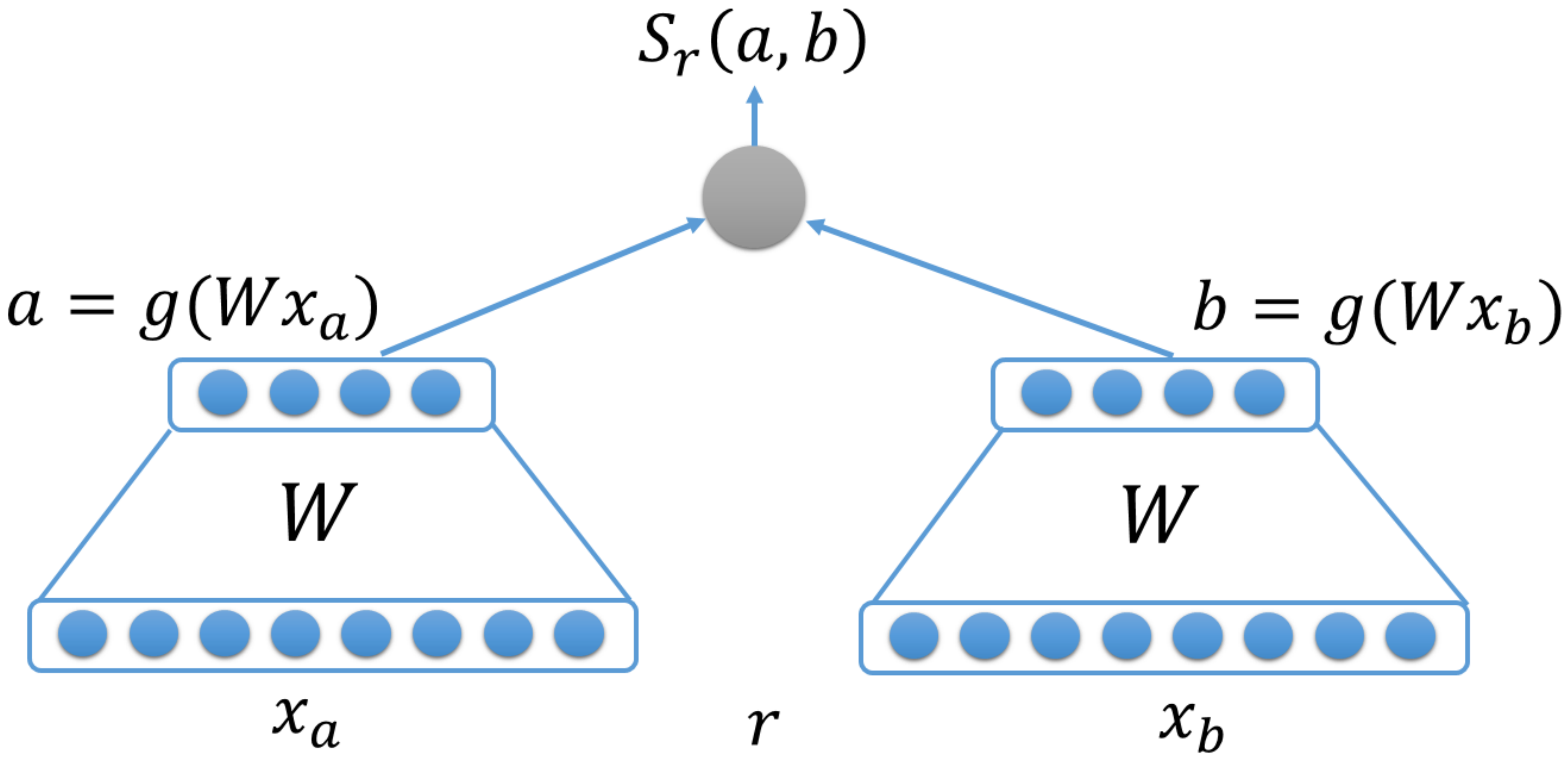


Relation Retrieval $(e_i, ?, e_j)$

- Positive entity pairs with equal number of negative pairs



Neural Knowledge Base Embedding



$S_r(a, b)$

$a = g(Wx_a)$

$b = g(Wx_b)$

W

W

x_a r x_b

Nicole Kidman *Nationality* *Australia*

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+ 2015)	$a^T diag(M_r) b$	$O(n_r \times k)$

n_r: #predicates, k: #dimensions of entity vectors, d: #layers

Empirical Comparisons of NN-based KB Embedding Methods [Yang+ ICLR-2015]

- Models with fewer parameters tend to perform better (for the datasets FB-15k and WN).
- The bilinear operator ($\boldsymbol{a}^T \boldsymbol{M}_r \boldsymbol{b}$) plays an important role in capturing entity interactions.
- With the same model complexity, multiplicative operations are superior to additive operations in modeling relations.
- Initializing entity vectors with pre-trained phrase embedding vectors can significantly boost performance.

Mining Horn-clause Rules

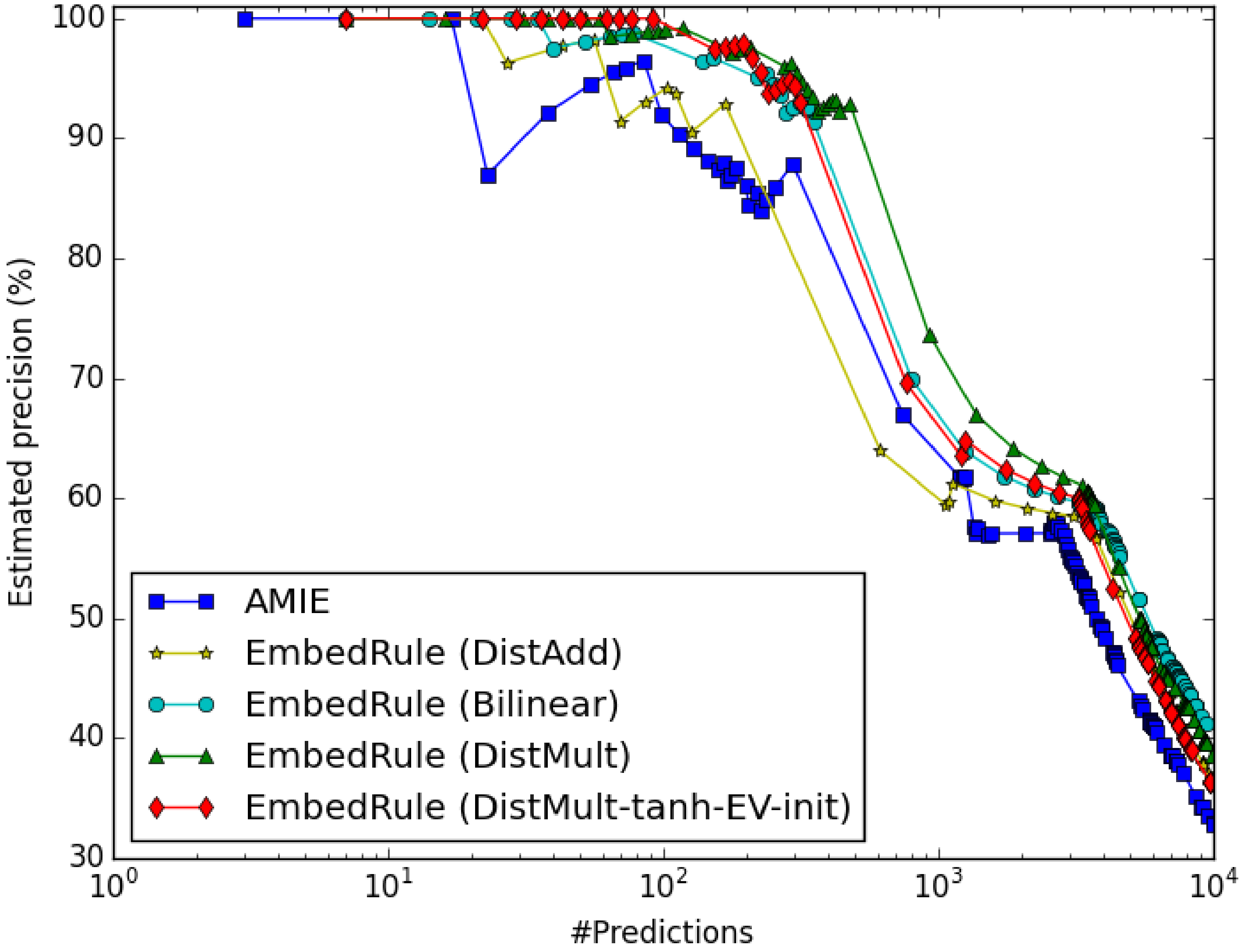
- Can relation embedding capture relation composition?

$$BornInCity(a,b) \wedge CityInCountry(b,c) \Rightarrow Nationality(a,c)$$

- Embedding-based Horn-clause rule extraction
  - For each relation r, find a chain of relations $r_1 \cdots r_n$, such that:
$$dist(M_r, M_1 \circ M_2 \circ \cdots \circ M_n) < \theta$$
  - $r_1(e_1,e_2) \wedge r_2(e_2,e_3) \cdots \wedge r_n(e_n,e_{n+1}) \rightarrow r(e_1,e_{n+1})$

- Advantages vs. Inductive Logic Programming
  - Search the relation space instead of instance space

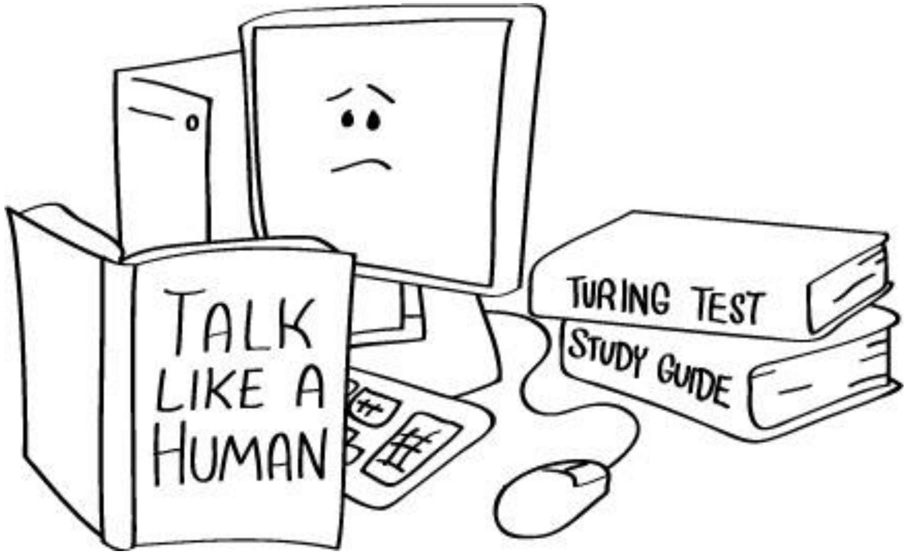
Aggregated Precision of Top Length-2 Rules



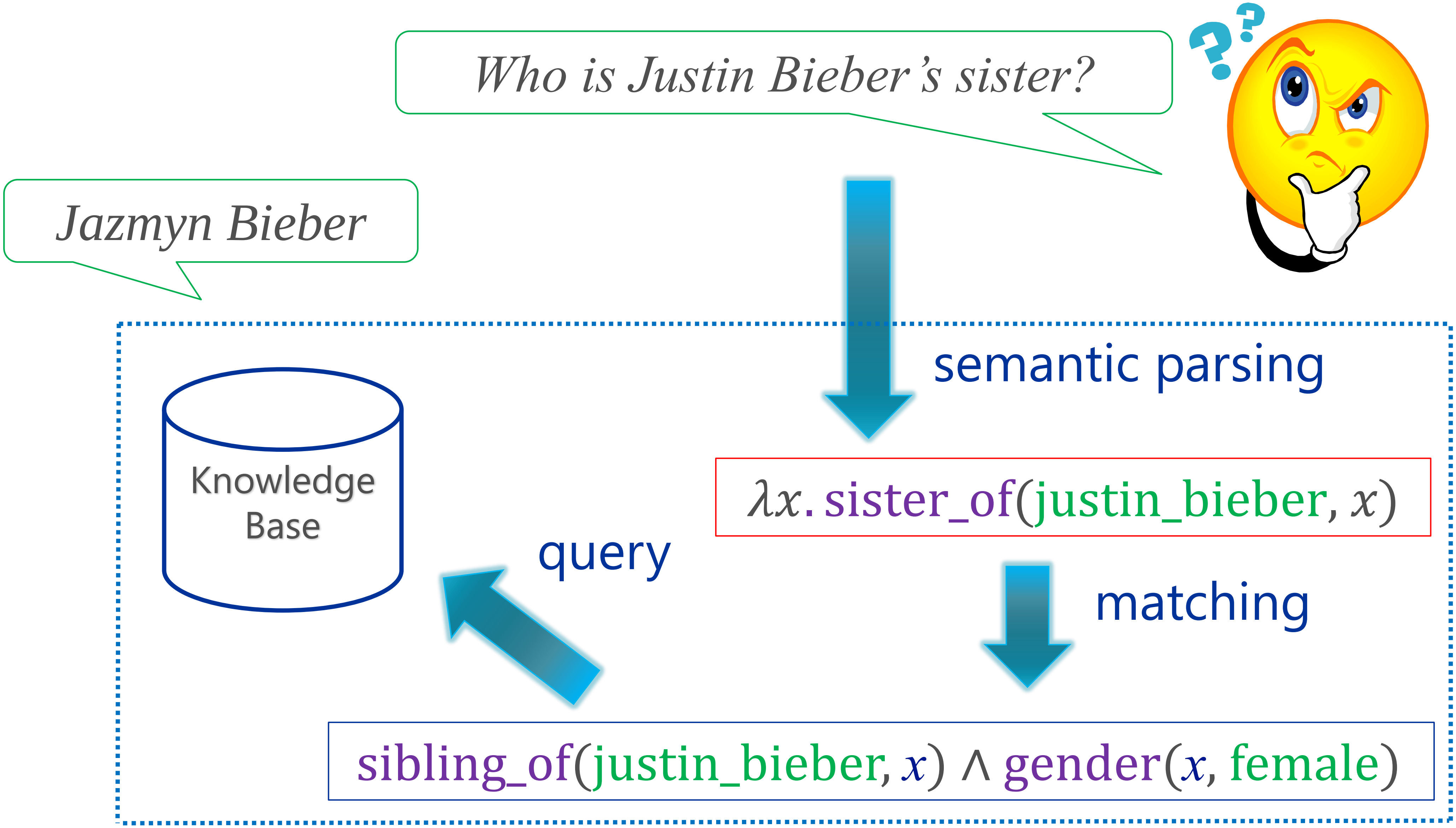
- AMIE [Galárraga+, WWW-2013] is an association rule-mining approach for large-scale KBs.
- Data: FB15k-401
- Execution time:
  - AMIE: 9 min.
  - EmbedRule: 2 min.

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?

Jazmyn Bieber

semantic parsing

$\lambda x.\,\text{sister\_of}(\text{justin\_bieber}, x)$

matching

$\text{sibling\_of}(\text{justin\_bieber}, x) \wedge \text{gender}(x, \text{female})$

query

Knowledge Base

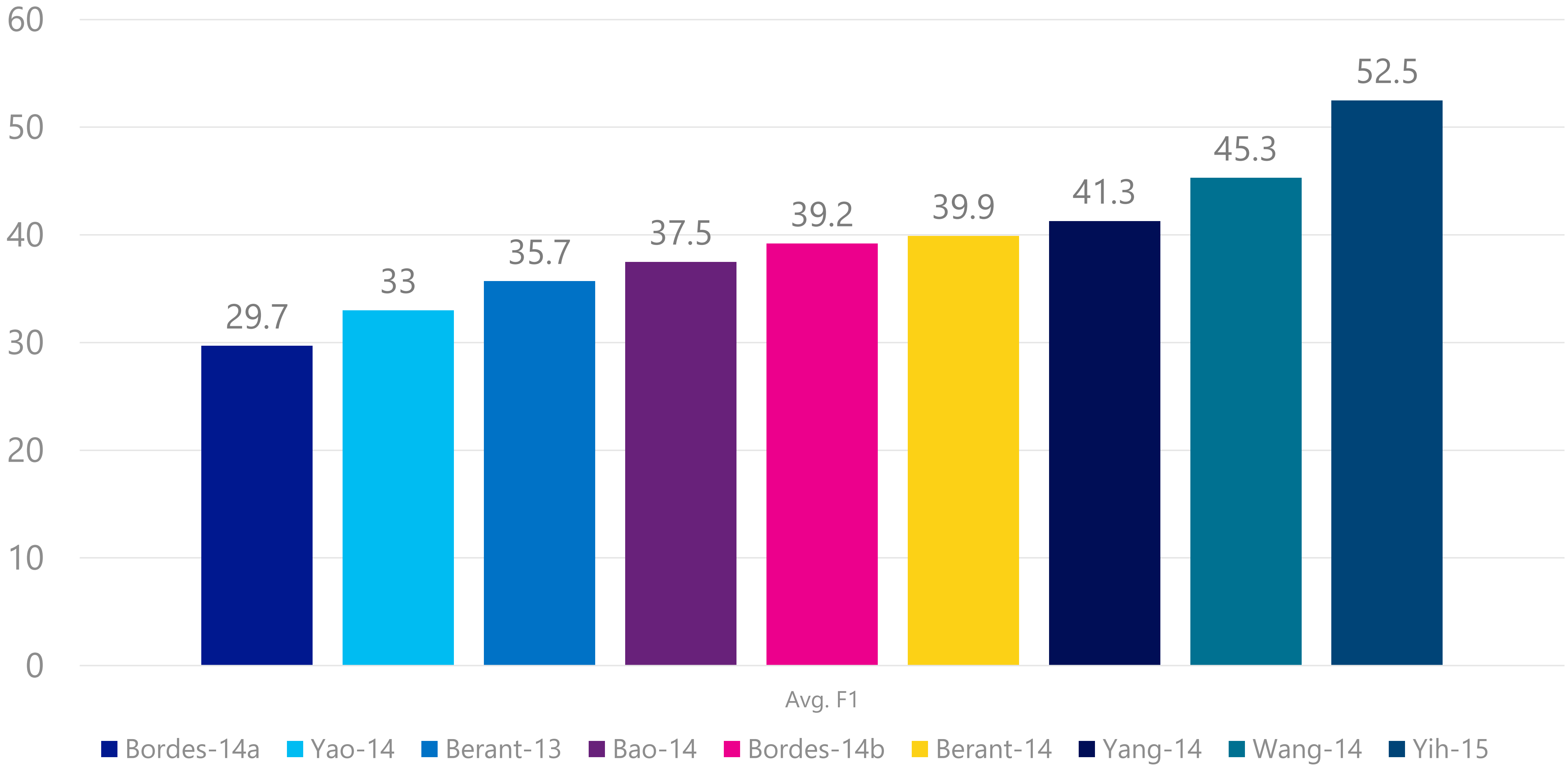
WebQuestions Dataset [Berant+ EMNLP-2013]

- *What character did Natalie Portman play in Star Wars?* ⇒ Padme Amidala
- *What kind of money to take to Bahamas?* ⇒ Bahamian dollar
- *What currency do you use in Costa Rica?* ⇒ Costa Rican colon
- *What did Obama study in school?* ⇒ political science
- *What do Michelle Obama do for a living?* ⇒ writer, lawyer
- *What killed Sammy Davis Jr?* ⇒ throat cancer

[Examples from Berant]

- 5,810 questions crawled from Google Suggest API and answered using Amazon MTurk
  - 3,778 training, 2,032 testing
  - A question may have multiple answers → using Avg. F1 (~accuracy)

Avg. F1 (Accuracy) on WebQuestions Test Set



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?"*

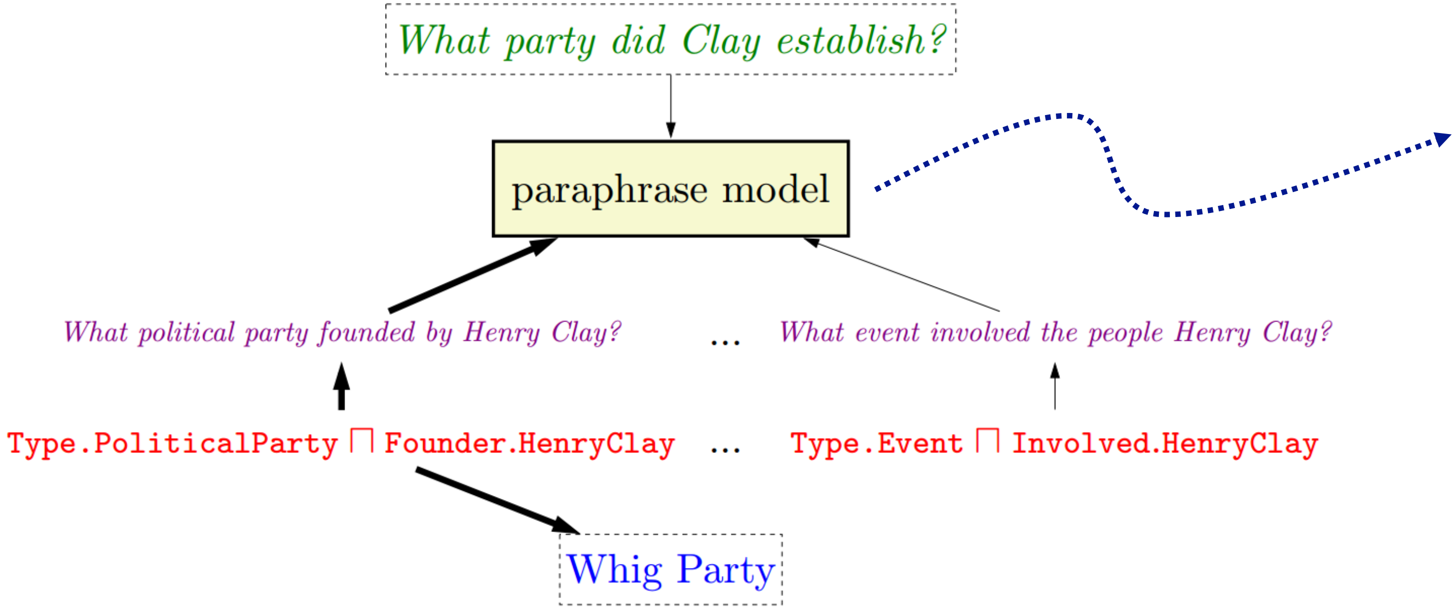
- Need to map them to the predicate defined in KB
    - location.dated_location.date_founded

Matching Question and Relation

- Similar text can map to very different relations
  - Q = *Who is the father of King George VI?*
  - R = people.person.parents
  - Q = *Who is the father of the Periodic Table?*
  - R = law.invention.inventor
- Estimate $P(R|Q)$ using naïve Bayes [Yao&VanDurme ACL-14]
  - $P(R|Q) \propto P(Q|R)P(R) \approx \prod_w P(w|R)P(R)$
  - Use ClueWeb09 dataset with Freebase entity annotations to create a "relation – sentence" parallel corpus
  - Derive $P(w|R)$ and $P(R)$ from the word alignment model (IBM Model 1)
  - Top words for film.film.directed_by: won, start, among, show.

Matching Questions

- Semantic Parsing via Paraphrasing [Berant&Liang ACL-14]



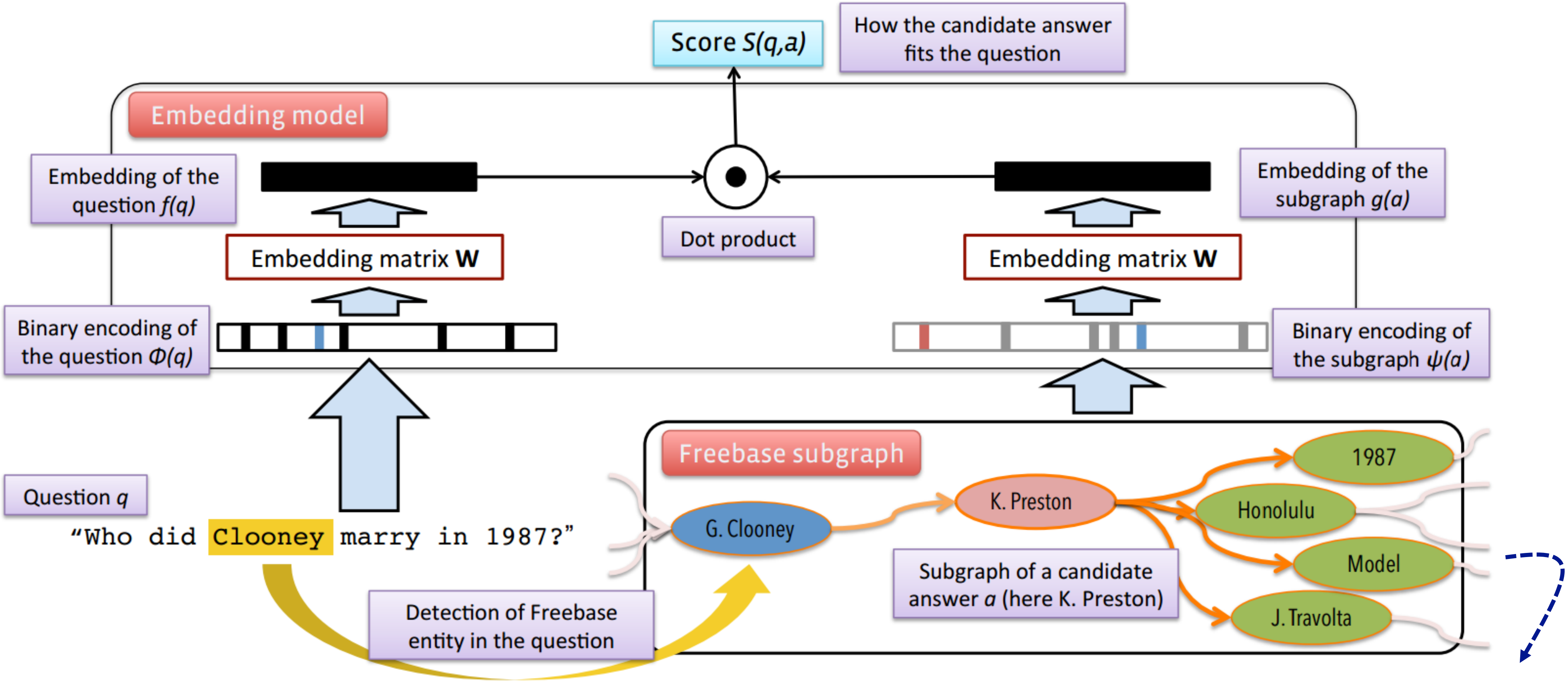
Learned using question paraphrases in Paralex [Fader+ ACL-13]

- Create phrase matching features using phrase table derived from word alignment results
- Represent questions as vectors (avg. of word vectors)

Subgraph Embedding [Bordes+ EMNLP-2014]

- Basic idea: map question and answer to vectors
  - q: question (Who did Clooney marry in 1987?)
  - a: answer candidate (K. Preston)
  - $S(q,a) = f(q)^{\mathrm{T}} g(a)$, where $f(q) = \mathbf{W}\phi(q), g(a) = \mathbf{W}\psi(a)$
- Answer candidate generation
  - Assume the topic entity (Clooney → G. Clooney) in q is given
  - All neighboring entities 1 or 2 edges away from topic entity
- Input encoding
  - $\phi(q)$: bag-of-word binary vectors
  - $\psi(a)$: binary encoding of the answer entity

Subgraph Embedding [Bordes+ EMNLP-2014]

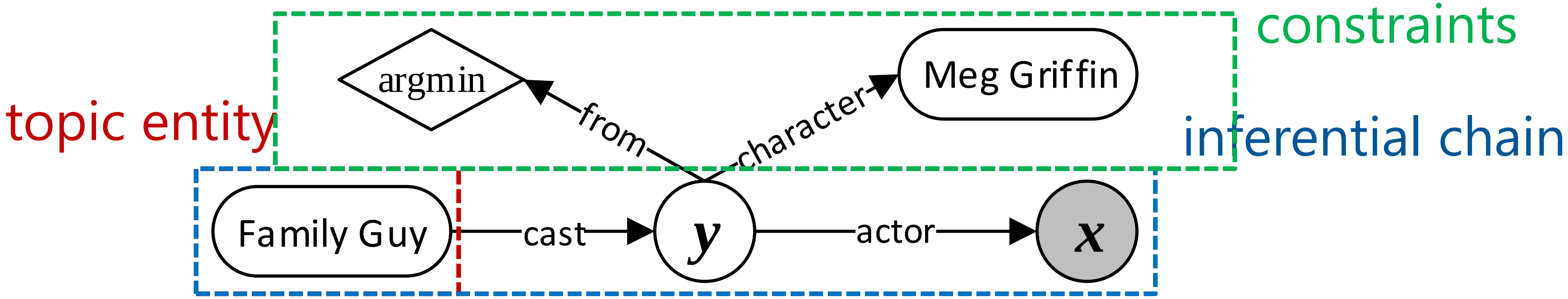


Other candidate answer encoding that includes the path, or other neighboring entities (subgraph)

Staged Query Graph Generation [Yih+ ACL-15]

- Query graph
  - Resembles subgraphs of the knowledge base
  - Can be directly mapped to a logical form in λ-calculus
  - Semantic parsing: a search problem that *grows* the graph through actions

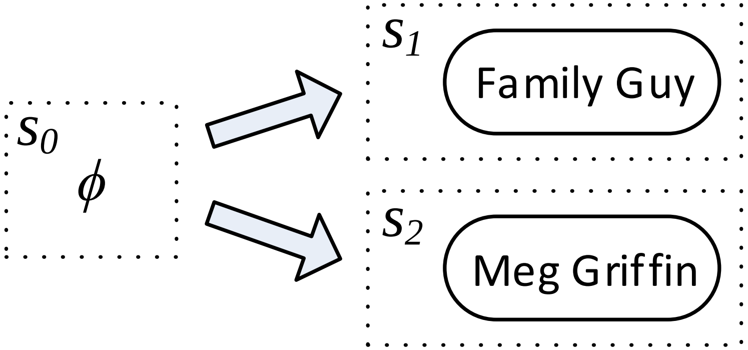
- Who first voiced Meg on Family Guy?
- $\lambda x. \exists y. \mathrm{cast}(\mathrm{FamilyGuy}, y) \wedge \mathrm{actor}(y, x) \wedge \mathrm{character}(y, \mathrm{MegGriffin})$



Graph Generation Stages

- **Who first voiced Meg on Family Guy?**

1. Topic Entity Linking [Yang&Chang ACL-15]

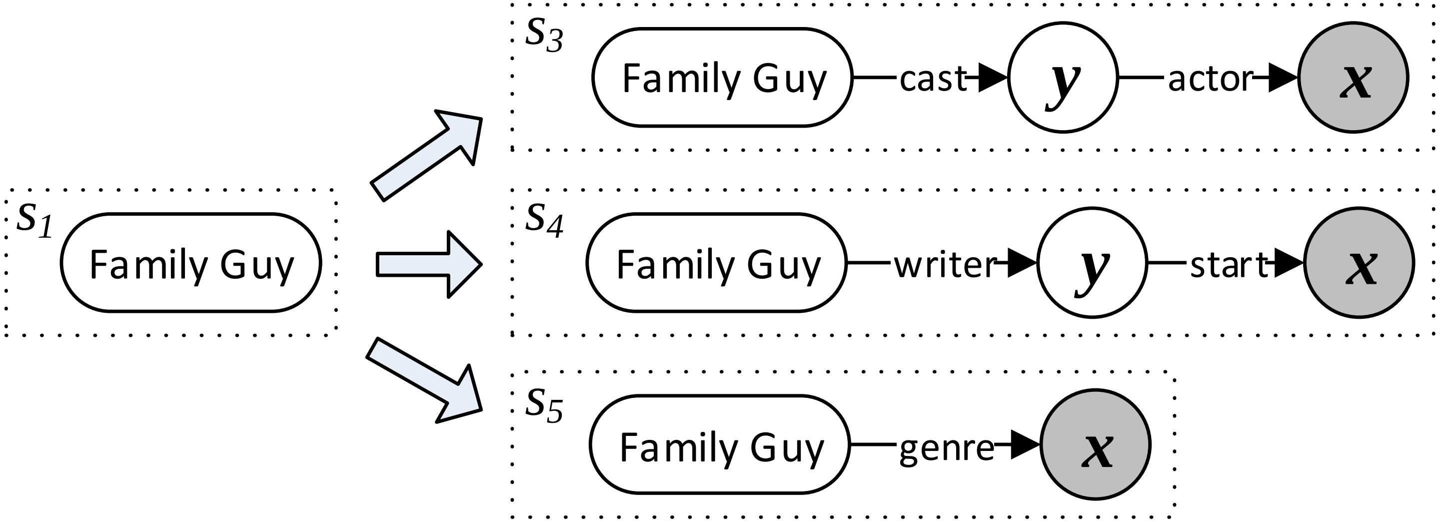


s_0 ϕ

s_1 Family Guy

s_2 Meg Griffin

2. Identify the core inferential chain



s_1 Family Guy

s_3 Family Guy —cast→ y —actor→ x

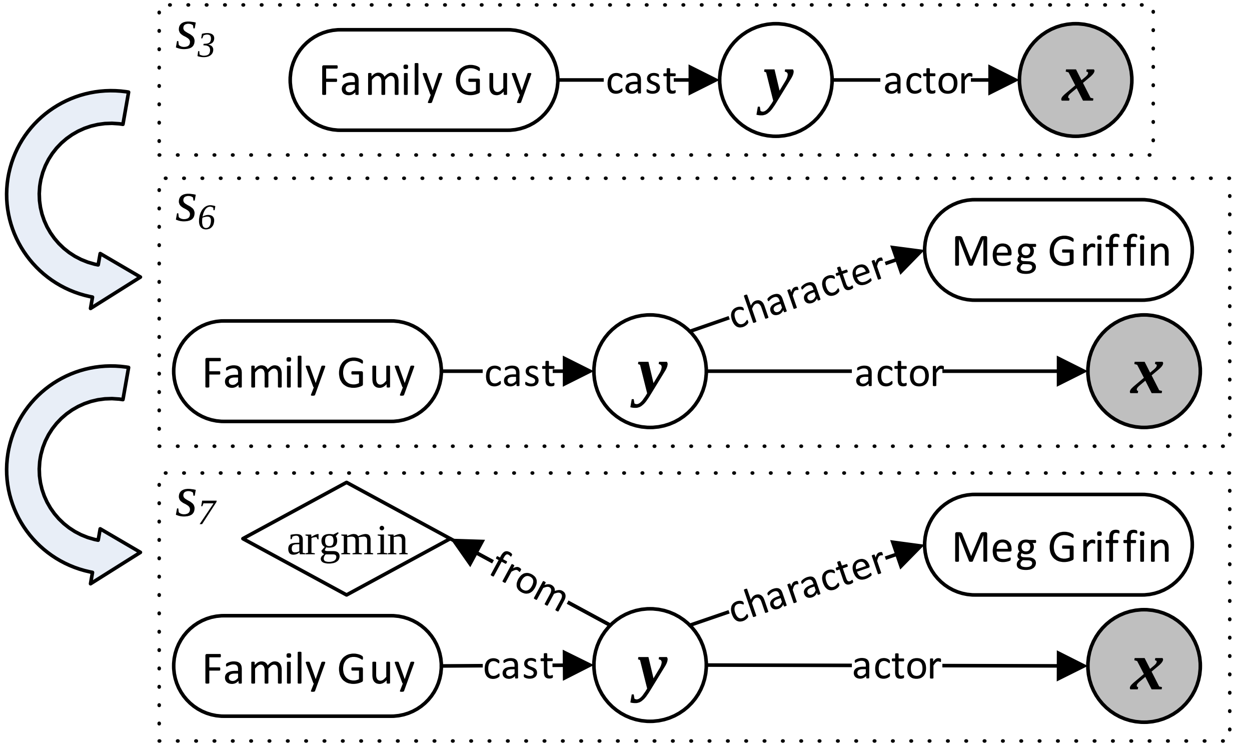
s_4 Family Guy —writer→ y —start→ x

s_5 Family Guy —genre→ x

Graph Generation Stages (cont'd)

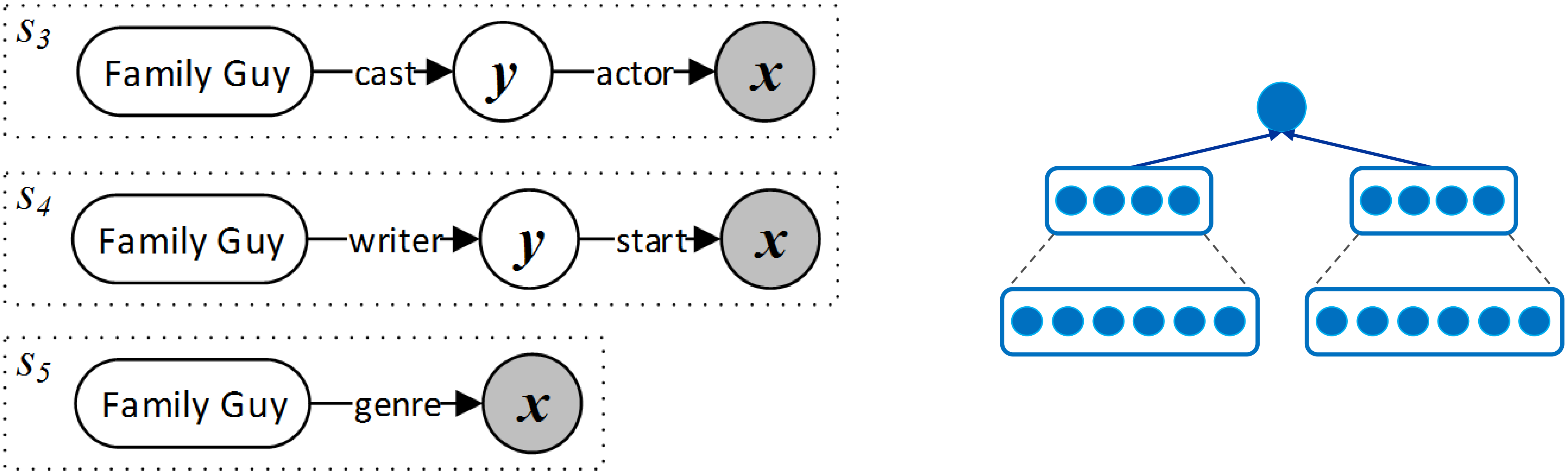
- Who first voiced Meg on Family Guy?

3. Augment constraints



Identify Inferential Chain using DSSM

- Who first voiced Meg on Family Guy?



- Semantic match (**“Who first voiced Meg on ⟨*e*⟩”**, “cast-actor”)
- Single pattern/relation matching model: 49.6% F_1 (vs. 52.5% F_1 Full)

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

Data & tools (partial list)

- Word2Vec https://code.google.com/p/word2vec/
- GloVe http://nlp.stanford.edu/projects/glove/
- MSR Continuous Space Text Representation http://aka.ms/msrcstr
- Knowledge base embedding, Semantic Parsing QA (to be released)

Conclusions

- Exciting advances in NN and continuous representations
  - Text processing & Knowledge reasoning

- Looking forward
  - Building an universal intelligence space
    - Text, Knowledge, Reasoning, …
    - Sent2Vec (DSSM) http://aka.ms/sent2vec
  - From component models to end-to-end solutions

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.
- Bahdanau, D., Cho, K., and Bengio, Y. 2015. Neural machine translation by joingly learning to align and translate, in ICLR 2015.
- 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).
- Chandar, A. P. S., Lauly, S., Larochelle, H., Khapra, M. M., Ravindran, B., Raykar, V., and Saha, A. (2014). An autoencoder approach to learning bilingual word representations. In NIPS.
- 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.
- Cui, L., Zhang, D., Liu, S., Chen, Q., Li, M., Zhou, M., and Yang, M. (2014). Learning topic representation for smt with neural networks. In ACL.
- 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.

References

- 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.
- Deng, L. and Yu, D. 2014. Deeping learning methods and applications. Foundations and Trends in Signal Processing 7:3-4.
- 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.
- Duh, K. 2014. Deep learning for natural language processing and machine translation. Tutorial. 2014.
- Duh, K., Neubig, G., Sudoh, K., and Tsukada, H. (2013). Adaptation data selection using neural language models: Experiments in machine translation. In ACL.
- 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
- Faruqui, M. and Dyer, C. (2014). Improving vector space word representations using multilingual correlation. In EACL.
- 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., Deng, L., and Shen, Y. 2014b. 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.

References

- Hermann, K. M. and Blunsom, P. (2014). Multilingual models for compositional distributed semantics. In ACL.
- 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., Osindero, S., and The, Y-W. 2006. A fast learning algorithm for deep belief nets. *Neural Computation*, 18: 1527-1554.
- 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.
- Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models., in EMNLLP
- Kiros, R., Zemel, R., and Salakhutdinov, R. 2013. Multimodal Neural Language Models, Proc. NIPS Deep Learning Workshop.
- Klementiev, A., Titov, I., and Bhattarai, B. (2012). Inducing crosslingual distributed representations of words. In COLING.
- Kocisky, T., Hermann, K. M., and Blunsom, P. (2014). Learning bilingual word representations by marginalizing alignments. In ACL.
- 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.
- Lao, N., Mitchell, T., and Cohen, W. 2011. Random walk inference and learning in a large scale knowledge base. In EMNLP.
- Lauly, S., Boulanger, A., and Larochelle, H. (2013). Learning multilingual word representations using a bag-of-words autoencoder. In NIPS.
- 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.

References

- Li, P., Hastie, T., and Church, K.. 2006. Very sparse random projections, in Proc. SIGKDD.
- Li, P., Liu, Y., and Sun, M. (2013). Recursive autoencoders for ITG-based translation. In EMNLP.
- Li, P., Liu, Y., Sun, M., Izuha, T., and Zhang, D. (2014b). A neural reordering model for phrase-based translation. In COLING.
- Liu, S., Yang, N., Li, M., and Zhou, M. (2014). A recursive recurrent neural network for statistical machine translation. In ACL.
- Liu, L., Watanabe, T., Sumita, E., and Zhao, T. (2013). Additive neural networks for statistical machine translation. In ACL.
- Lu, S., Chen, Z., and Xu, B. (2014). Learning new semi-supervised deep auto-encoder features for statistical machine translation. In ACL.
- Maskey, S., and Zhou, B. 2012. Unsupervised deep belief feature for speech translation, in ICASSP.
- 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.
- Niehues, J. and Waibel, A. (2013). Continuous space language models using Restricted Boltzmann Machines, in IWLT.
- Reddy, S., Lapata, M., and Steedman, M. 2014. Large-scale semantic parsing without question-answer pairs. Transactions of the Association for Computational Linguistics (TACL).
- 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. 2012. Continuous space translation models for phrase-based statistical machine translation, in COLING.

References

- Schwenk, H., Rousseau, A., and Attik, M., 2012. Large, pruned or continuous space language models on a gpu for statistical machine translation, in NAACL-HLT 2012 Workshop.
- 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
- Son, L. H., Allauzen, A., and Yvon, F. (2012). Continuous space translation models with neural networks. In NAACL.
- 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.
- Songyot, T. and Chiang, D. (2014). Improving word alignment using word similarity. In EMNLP.
- Sundermeyer, M., Alkhouli, T., Wuebker, J., and Ney, H. (2014). Translation modeling with bidirectional recurrent neural networks, in EMNLP.
- Tamura, A., Watanabe, T., and Sumita, E. (2014). Recurrent neural networks for word alignment model. In ACL.
- Tran, K. M., Bisazza, A., and Monz, C. (2014). Word translation prediction for morphologically rich languages with bilingual neural networks. In EMNLP.
- 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. Songyot, T. and Chiang, D. (2014). Improving word alignment using word similarity. In EMNLP.
- Vaswani, A., Zhao, Y., Fossum, V., and Chiang, D. 2013. Decoding with large-scale neural language models improves translation, in EMNLP.
- 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.
- Wu, H., Dong, D., Hu, X., Yu, D., He, W., Wu, H., Wang, H., and Liu, T. (2014a). Improve statistical machine translation with context-sensitive bilingual semantic embedding model. In EMNLP.
- Wu, Y., Watanabe, T., and Hori, C. (2014b). Recurrent neural network-based tuple sequence model for machine translation. In COLING.

References

- Yang, B., Yih, W., He, X., Gao, J., and Deng L. 2014. In NIPS-2014 Workshop Learning Semantics.
- Yang, N., Liu, S., Li, M., Zhou, M., and Yu, N. (2013). Word alignment modeling with context dependent deep neural network. In ACL.
- 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.
- Yih, W., Chang, M-W., He, X., Gao, J. 2015. Semantic parsing via staged query graph generation: question answering with knowledge base, In ACL.
- Zeiler, M. and Fergus, R. 2013. Visualizing and understanding convolutional networks, arXiv:1311.2901, pp. 1-11.
- Zhang, J., Liu, S., Li, M., Zhou, M., and Zong, C. (2014). Bilingually-constrained phrase embeddings for machine translation. In ACL.
- Zou, W. Y., Socher, R., Cer, D., and Manning, C. D. (2013). Bilingual word embeddings for phrase-based machine translation. In EMNLP.