Deep Learning for Natural Language Processing

Xiaodong He

DLTC, Microsoft Research
Redmond, WA, USA

Machine Learning Summer School
Shenzhen Research Institute of Big Data
The Chinese University of Hong Kong (Shenzhen)

August 4th, 2016

About me

Research interests:

Artificial Intelligence: deep learning, natural language, vision, speech, information retrieval, knowledge representation.

Published mainly at ACL, EMNLP, NAACL, CVPR, ICASSP, SIGIR, WWW, CIKM, ICLR, NIPS, IEEE TASLP, IEEE SPM, Proc. IEEE

Part of this tutorial is based on

- Xiaodong He, Jianfeng Gao, Li Deng. "*Deep Learning for Natural Language Processing*," Tutorial, CIKM 2014, Shanghai, USA
- Scott Yih, Xiaodong He, Jianfeng Gao. "*Deep Learning and Continuous Representations for Natural Language Processing*," Tutorial, NAACL, 2015, San Diego, USA
- Scott Yih, Xiaodong He, Jianfeng Gao. "*Deep Learning and Continuous Representations for Natural Language Processing*," Tutorial, IJCAI, 2016, New York City, USA

And with materials based on collaborations with many colleagues.

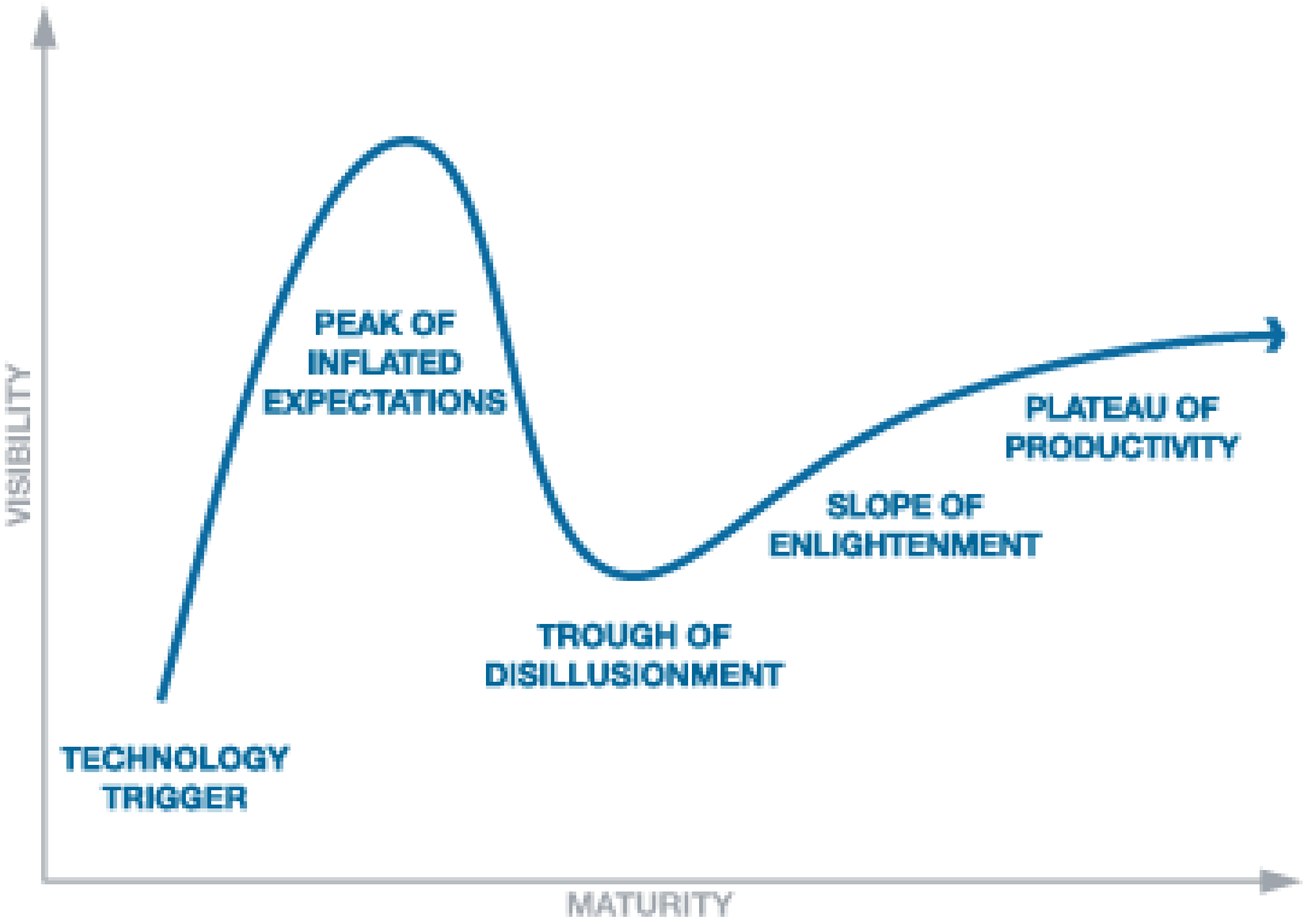
Tutorial Outline

- Part I: Introduction to Deep Learning
- Part II: Deep learning in statistical machine translation and conversation
- Part III: Deep Structured Semantic Models (DSSM) and IR/NL Applications
- Part IV: NLU: Knowledge Base representation and Question answering
- Part V: Deep reinforcement learning in NLP
- Part VI: Image-language multimodal learning and inference
- Part VII: Conclusion

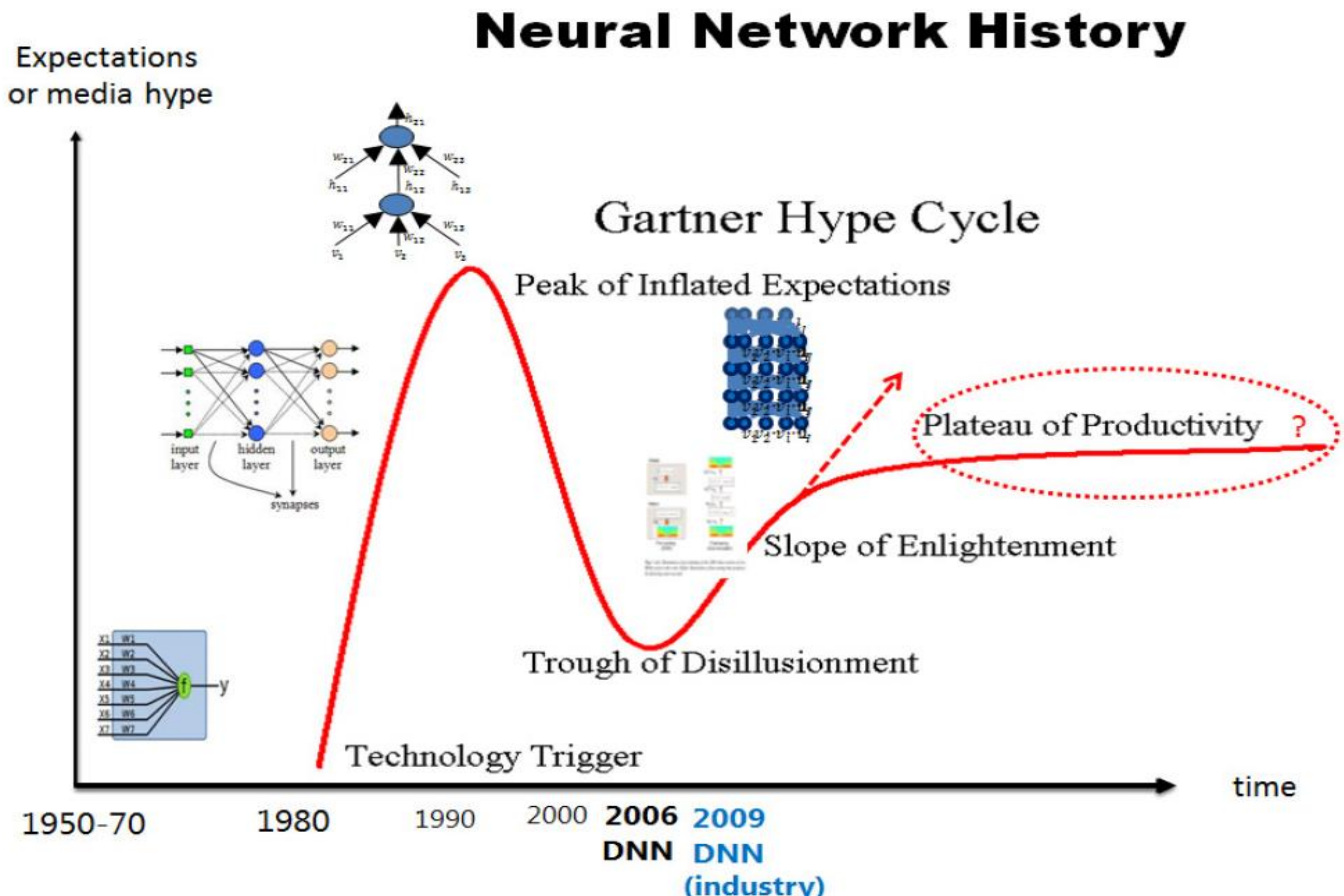
Part I

Background

Gartner hype cycle



A brief history of deep neural networks (DNN)



[Deng & Yu 14]

10 BREAKTHROUGH TECHNOLOGIES 2013

MIT Technology Review

Introduction | The 10 Technologies | Past Years

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

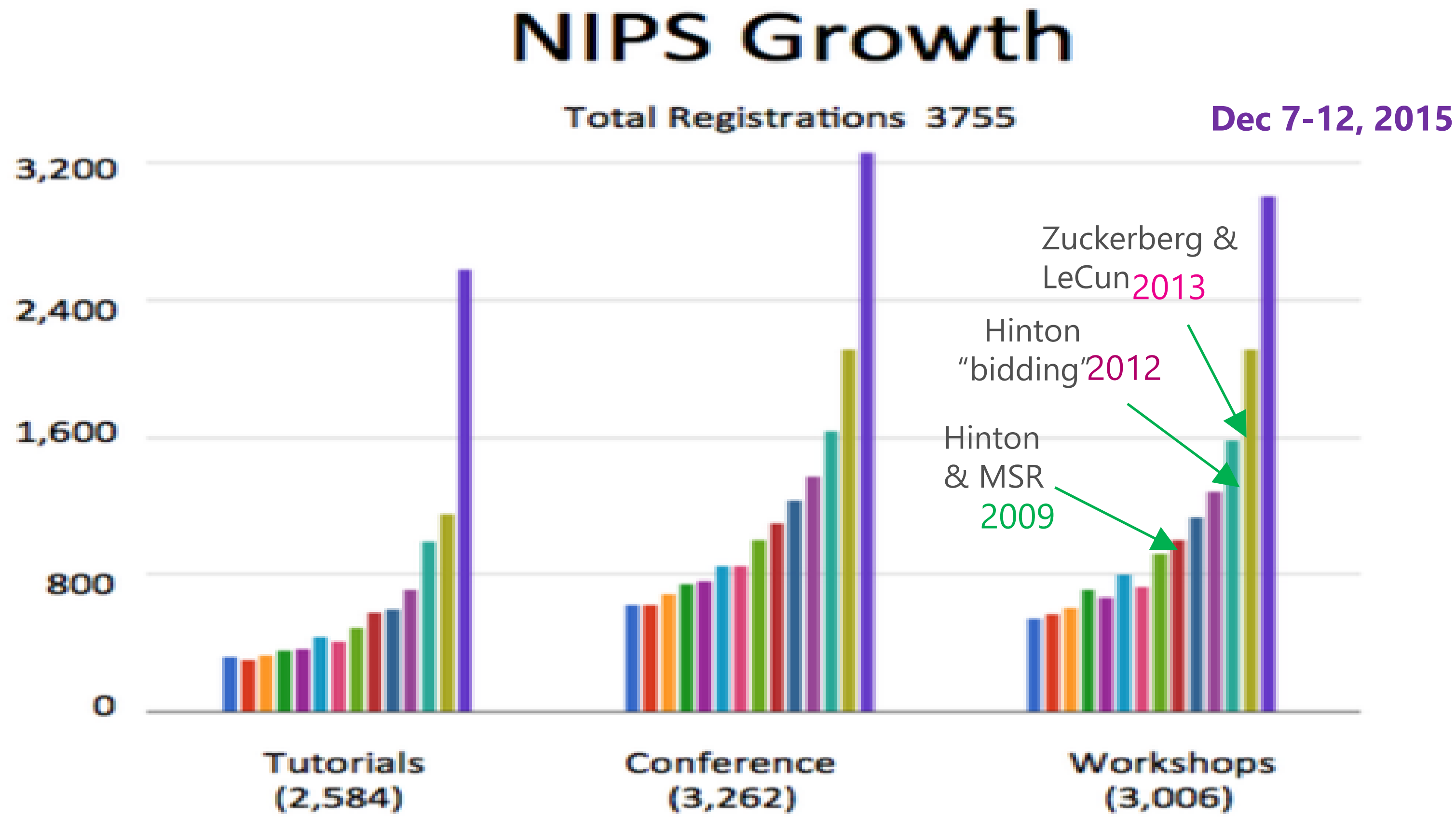
Collecting and analyzing information from simple cell phones can provide surprising insights into how people move about and behave – and even help us understand the spread of diseases.

Supergrids

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

Deep learning in academia: centered at NIPS 2015

Neil Lawrence @lawrennd · Dec 7
#NIPS2015 attendance numbers. Massive growth across the board but over 100% in tutorials.



Geoff Hinton

The New York Times

Scientists See Promise in Deep-Learning Programs
John Markoff November 23, 2012

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

The universal translator on “Star Trek” comes true…

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

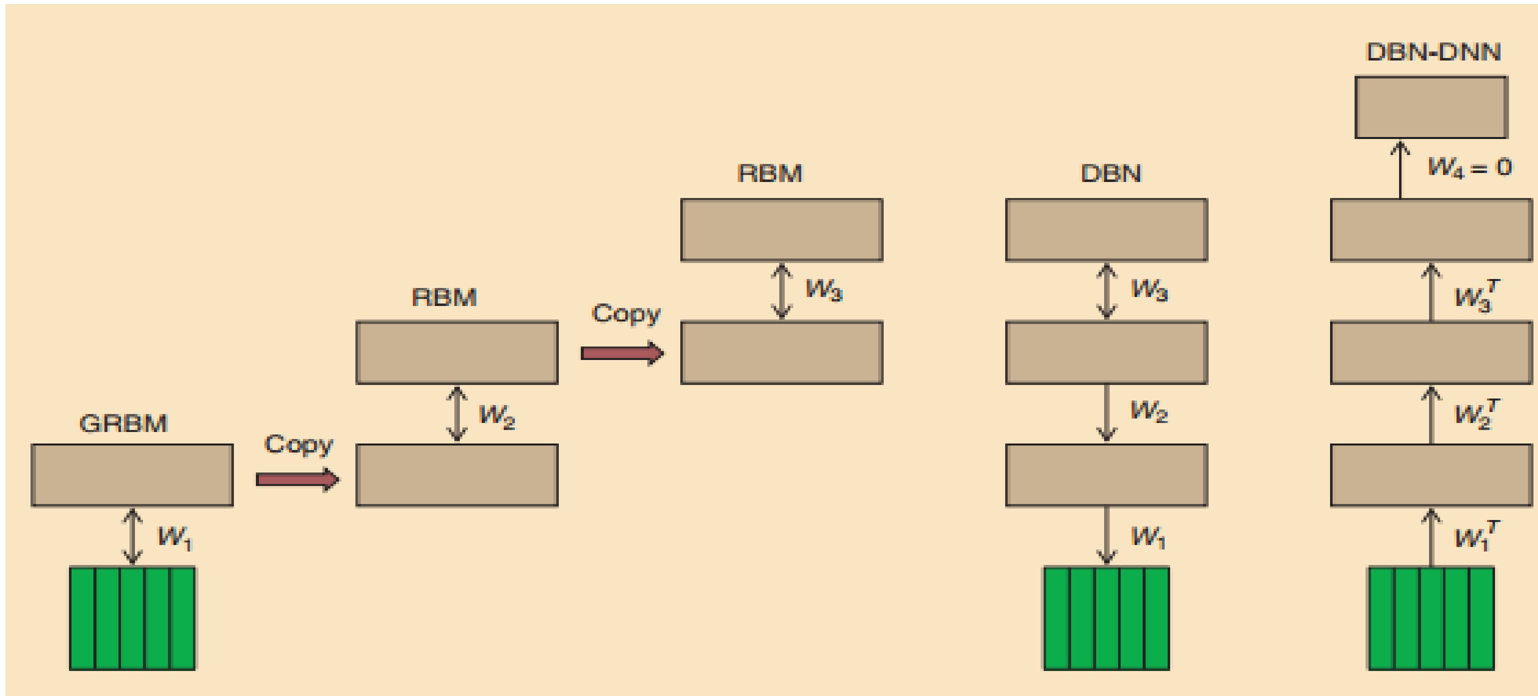
Geoff Hinton

Li Deng

Dong Yu

DNN: (Fully-Connected) Deep Neural Networks

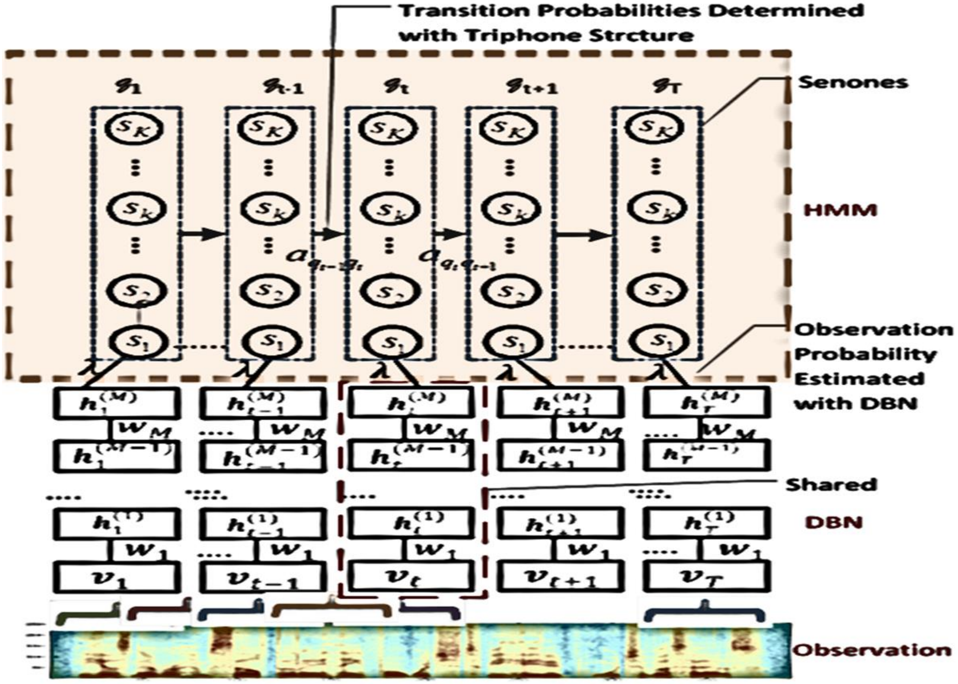
Hinton, Deng, Yu, et al., DNN for AM in speech recognition, *IEEE SPM*, 2012



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.

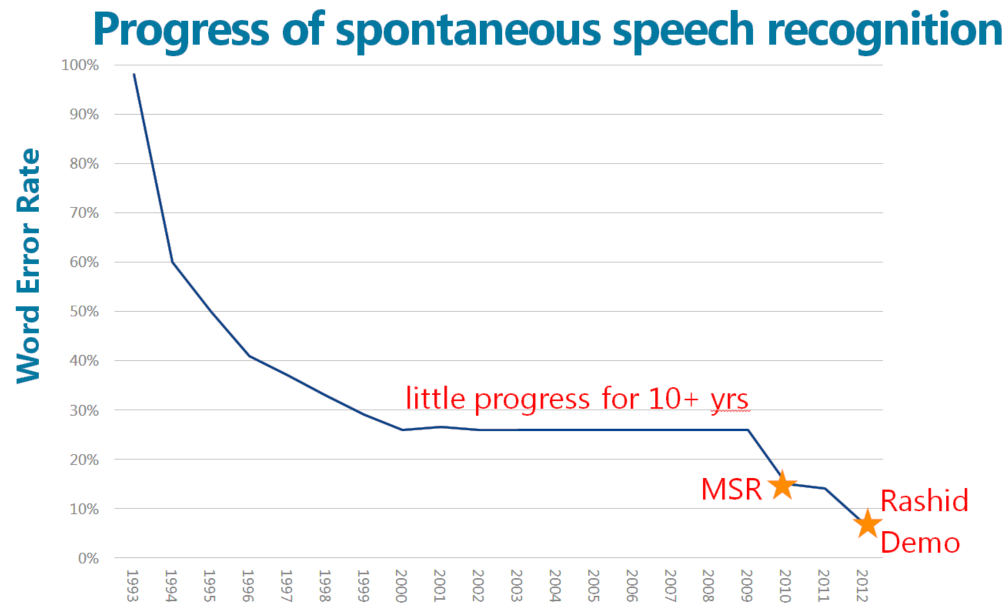
Then add outputs and train the DNN with backprop.



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)!

Skype to get 'real-time' translator

Analysts say the translation feature could have wide ranging applications

/ MOBILE

Ina Fried

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

By Ina Fried

ETHICS BIO ARTICLES

May 27, 2014, 5:48 PM PDT

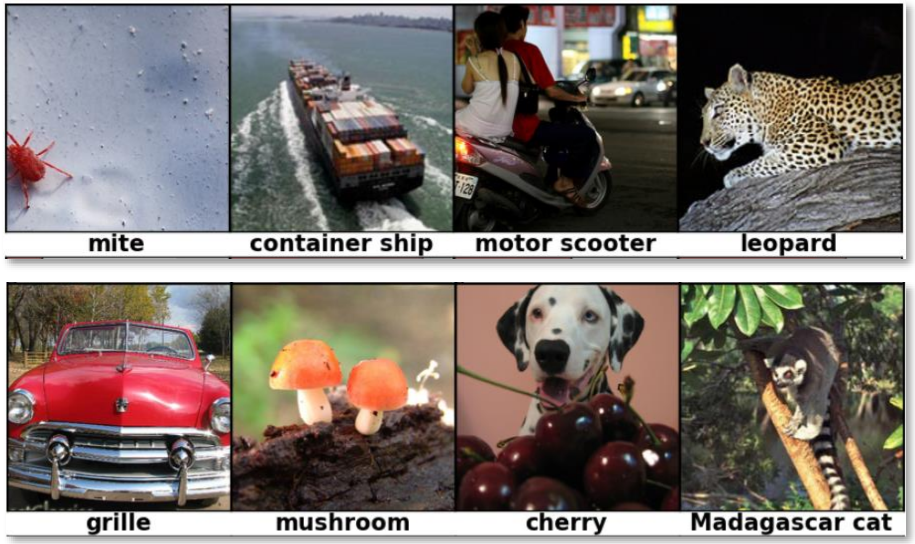
Remember the universal translator on Star Trek? The gadget that let Kirk and Spock talk to aliens?

Deep learning in computer vision

ImageNet: large scale image recognition bencvhmark

E.g., ImageNet provides hundreds to thousands of images for each category, aka **synset**, in the WordNet.

IMAGENET

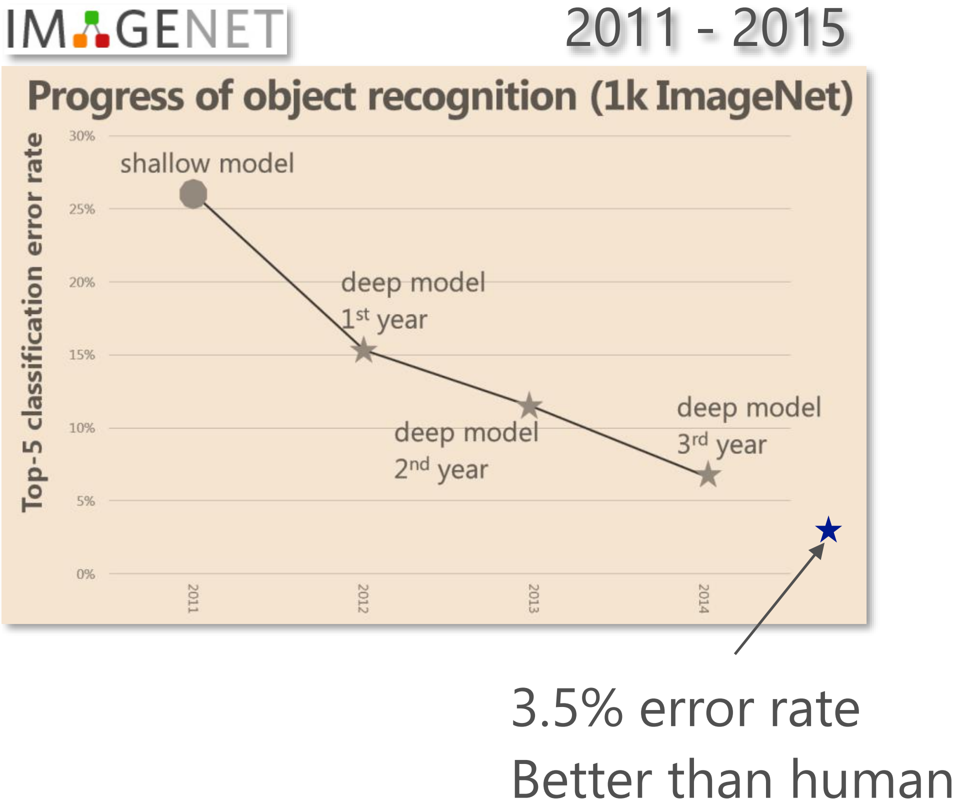


[Russakovsky, Deng, Fei-Fei, et al., 2014]

Great success on ImageNet

Dramatic progress in recent years thanks to deep CNN [LeCun, Bottou, Bengio, Haffner, 1998, Krizhevsky, Sutskever, Hinton, 2012].

First time surpassed human-level performance, now top5 err = 3.5% on ImageNet classification [He, Zhang, Ren, Sun, 2015]



The focus of this tutorial

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

A query classification problem

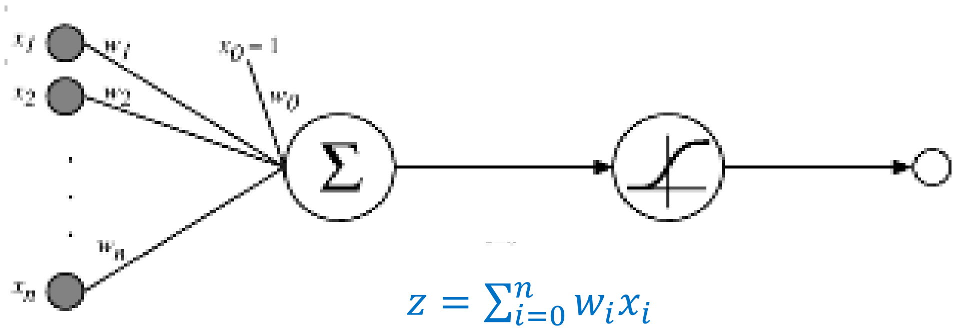
- Given a search query $\boldsymbol{q}$, e.g., “denver sushi downtown”
- Identify its domain $\boldsymbol{c}$ e.g.,
  - Restaurant
  - Hotel
  - Nightlife
  - Flight
  - etc.
- So that a search engine can tailor the interface and result to provide a richer user experience

A single neuron model

- For each domain 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

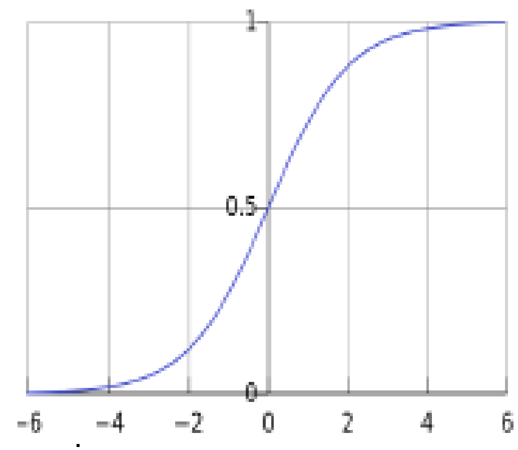


$z = \sum_{i=0}^{n} w_i x_i$

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

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

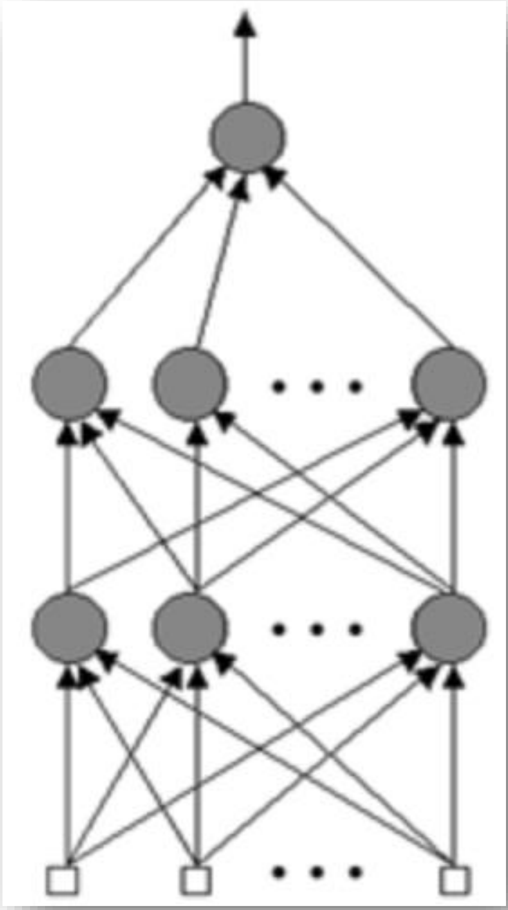
- w: weight vector to be learned
- z: weighted sum of input features
- σ: the logistic function
  - Turn a score to a probability
  - non-linear activation function, essential in DNN models



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 (e.g., 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 $y^1 = \sigma(\mathbf{W}_1 x)$

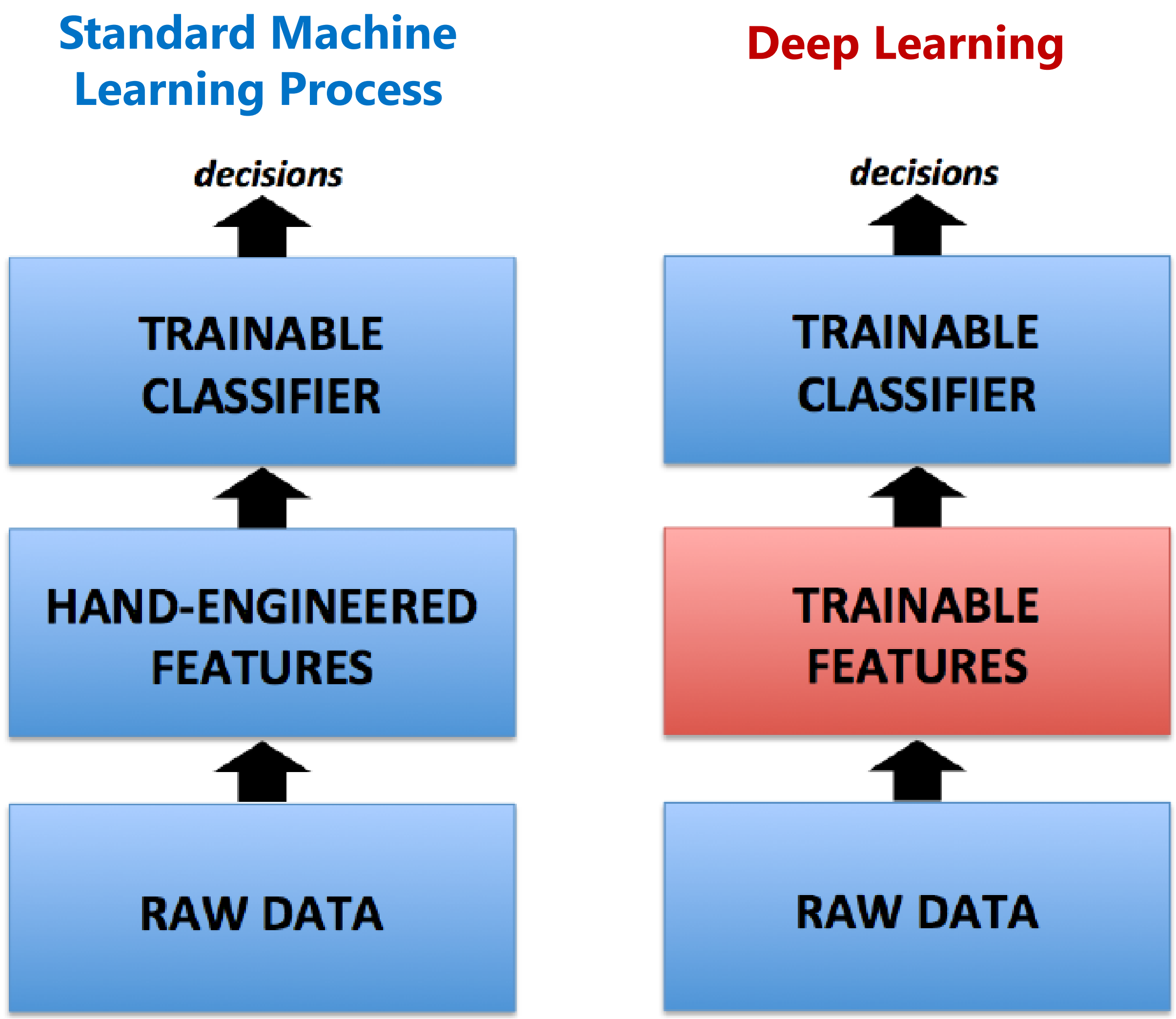
Projection matrix $\mathbf{W}_1$

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).

Use back propagation (BP) algorithm for training

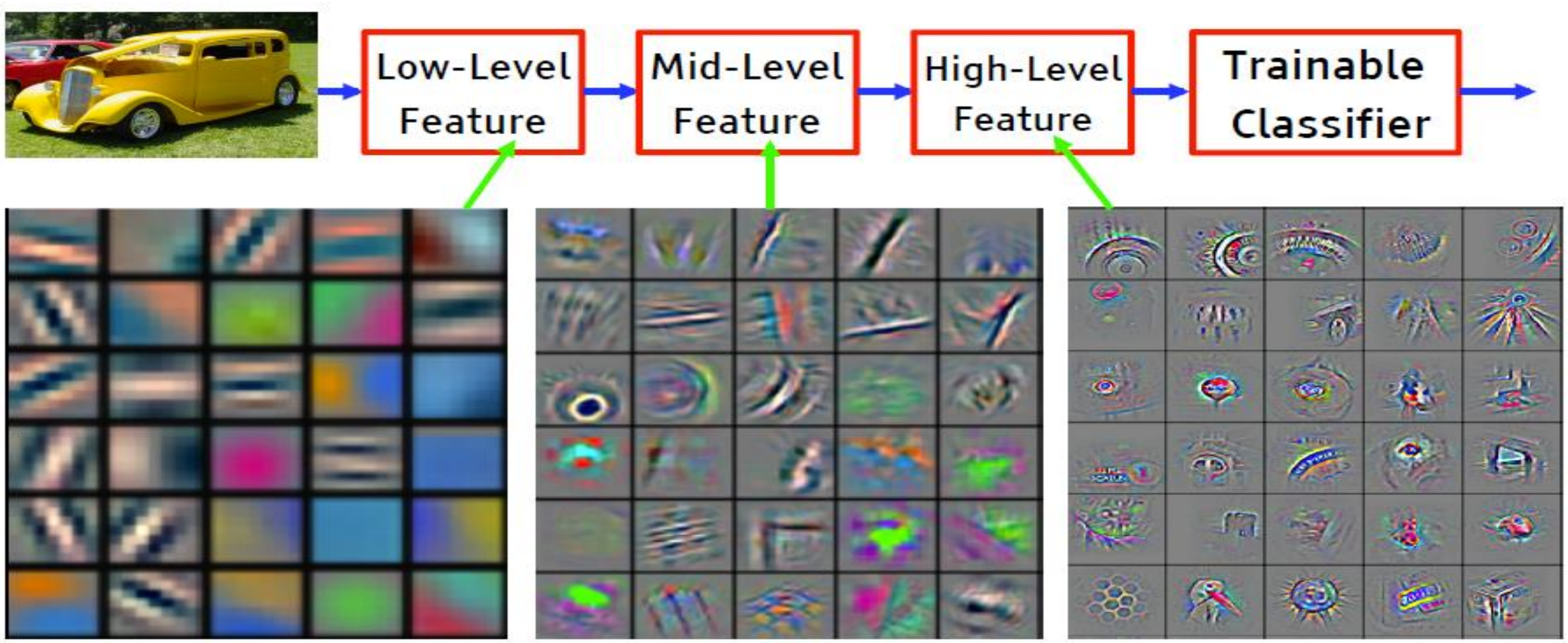


Adapted from [Duh 14]

Why Multiple Layers?

DL tutorial at NIPS'2015

- Hierarchy of representations with increasing level of abstraction
- Each layer is a trainable feature transform
- Image recognition: pixel → edge → texton → motif → part → object
- ?? Text: character → word → word group → clause → sentence → story



Different forms of DNN

- Classification task – label X by Y
  - Multi-Layer Perceptron
  - Convolutional NN
- Ranking task – compute the sim btw X and Y
  - Siamese neural network [Bromley et al. 1993]
  - Deep Semantic Similarity Model (DSSM)
- (Text) Generation task – generate Y from X
  - Seq2Seq (RNN/LSTM)
  - Memory Network

Deep Semantic Similarity Model (DSSM)

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

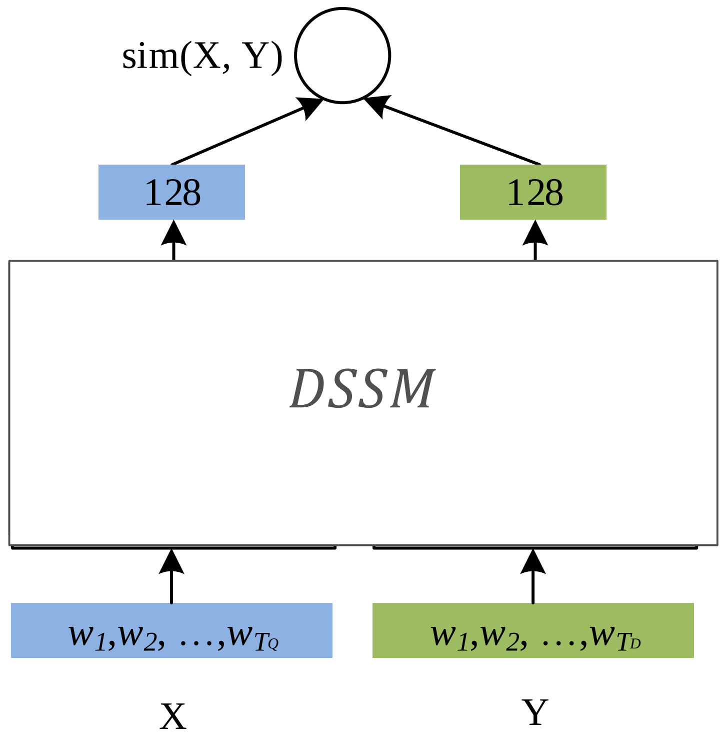
- Compute semantic similarity btw text strings X and Y
  - Map X and Y to feature vectors in a latent semantic space via deep neural net
  - Compute the cosine similarity between the feature vectors
  - Also called "Deep Structured Similarity Model" in [Huang+ 13]

Tasks	**X**	**Y**	**Ref**
Machine translation	*Text in language A*	*Translation in language B*	[Gao+ 14a]
Web search	*Search query*	*Web document*	[Huang+ 13; Shen+ 14]
Image captioning	*Image*	*Text caption*	[Fang+ 15]
Question Answering	*Question*	*Answer*	[Yih+ 15]
Contextual entity linking	*Mention (in text)*	*Entities (in Satori)*	[Gao+ 14b]
Ad selection	*Search query*	*Ad keywords*	
…	…	…	

Sent2Vec (DSSM) http://aka.ms/sent2vec

DSSM: Compute Similarity in Semantic Space

Relevance measured by cosine similarity

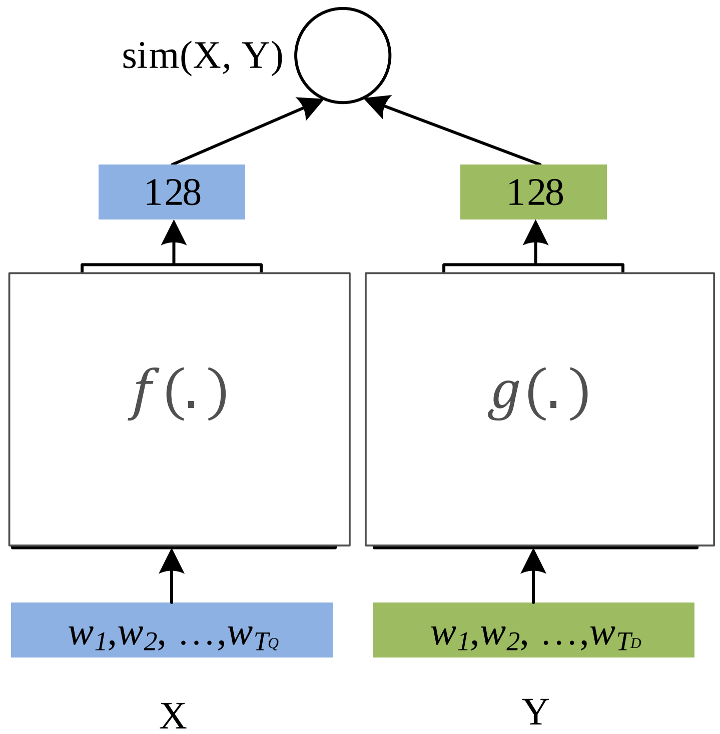


Word sequence x_t

Learning: maximize the similarity between X (source) and Y (target)

DSSM: Compute Similarity in Semantic Space

Relevance measured by cosine similarity



Word sequence x_t

Learning: maximize the similarity between X (source) and Y (target)

Representation: use DNN to extract abstract semantic representations

Convolutional DSSM [Gao+ 14b; Shen+ 14]

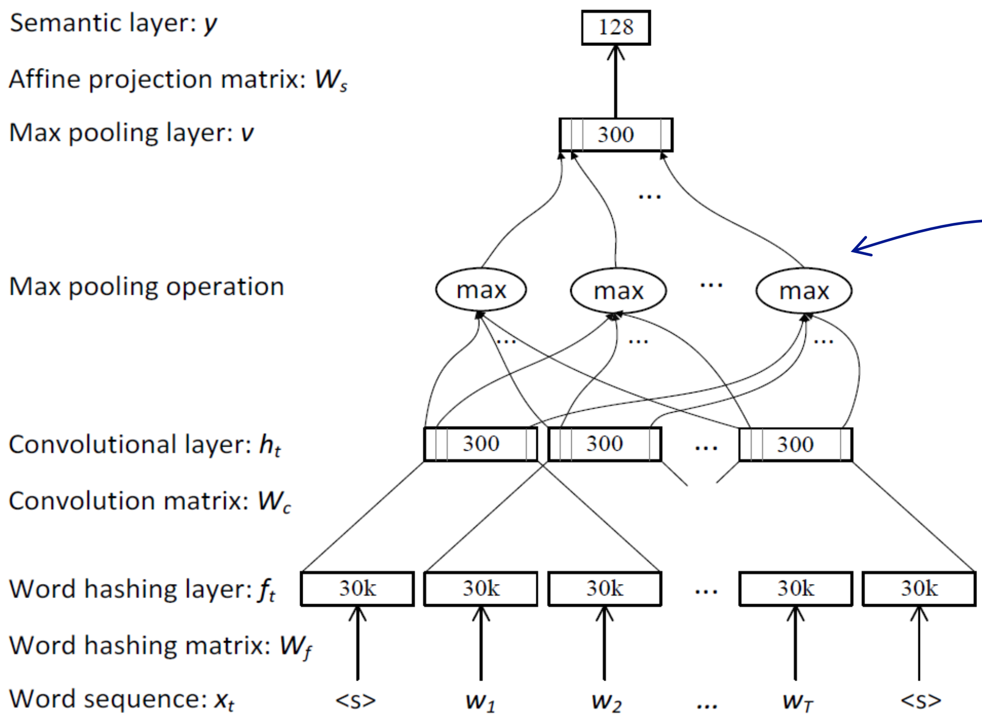
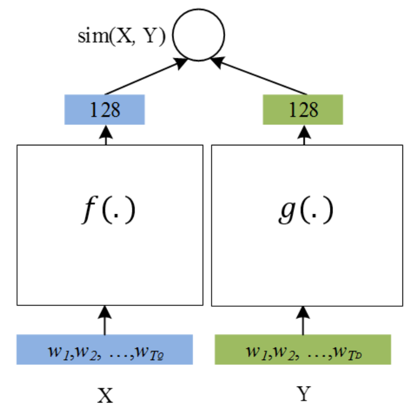
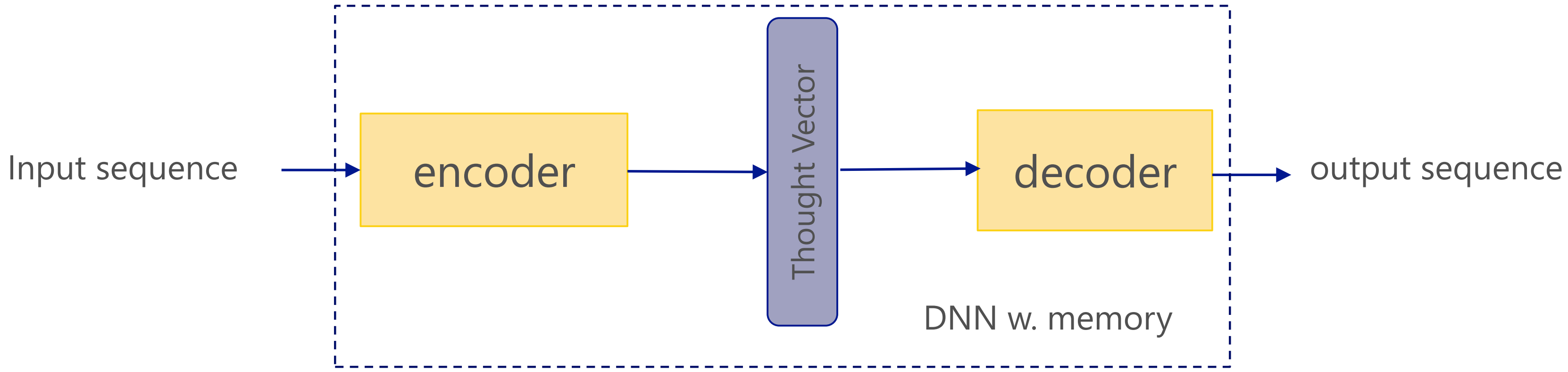


Figure 1: Illustration of the C-DSSM. A convolutional layer with the window size of three is illustrated.

- Model local context at the convolutional layer
- Model global context at the pooling layer
- Ident... pts in X (and Y)



Sequence-to-Sequence Tasks



- Statistical Machine translation (SMT):
  - A sentence in source language → A sentence in target language
- Conversation (chitchat):
  - Context + message → response
- Question answering + recommendation dialog:
  - Knowledge base + context + question → answer/recommendation

QA + Recommendation Dialog [Dodge+ 16]

Information/sentences retrieved from Knowledge bases, e.g., personal profile, Satori etc.

Conversation context

Query

Long-Term Memories	h_i	Shaolin Soccer directed_by Stephen Chow Shaolin Soccer written_by Stephen Chow Shaolin Soccer starred_actors Stephen Chow Shaolin Soccer release_year 2001 Shaolin Soccer has_genre comedy Shaolin Soccer has_tags martial arts, kung fu soccer, stephen chow Kung Fu Hustle directed_by Stephen Chow Kung Fu Hustle written_by Stephen Chow Kung Fu Hustle starred_actors Stephen Chow Kung Fu Hustle has_genre comedy action Kung Fu Hustle has_imdb_votes famous Kung Fu Hustle has_tags comedy, action, martial arts, kung fu, china, soccer, hong kong, stephen chow The God of Cookery directed_by Stephen Chow The God of Cookery written_by Stephen Chow The God of Cookery starred_actors Stephen Chow The God of Cookery has_tags hong kong Stephen Chow From Beijing with Love directed_by Stephen Chow From Beijing with Love written_by Stephen Chow From Beijing with Love starred_actors Stephen Chow, Anita Yuen . . . <and more> . . .
Short-Term Memories	c_1^u c_1^r	1) I'm looking a fun comedy to watch tonight, any ideas? 2) Have you seen Shaolin Soccer? That was zany and great.. really funny but in a whacky way.
Input	c_2^u	3) Yes! Shaolin Soccer and Kung Fu Hustle are so good I really need to find some more Stephen Chow films I feel like there is more awesomeness out there that I haven't discovered yet ...
Output	y	4) God of Cookery is pretty great, one of his mid 90's hong kong martial art comedies.

End-to-End Memory Networks (MemNN)

[Sukhbaatar+ 15]

- Retrieving long-term mem x
- Embedding input

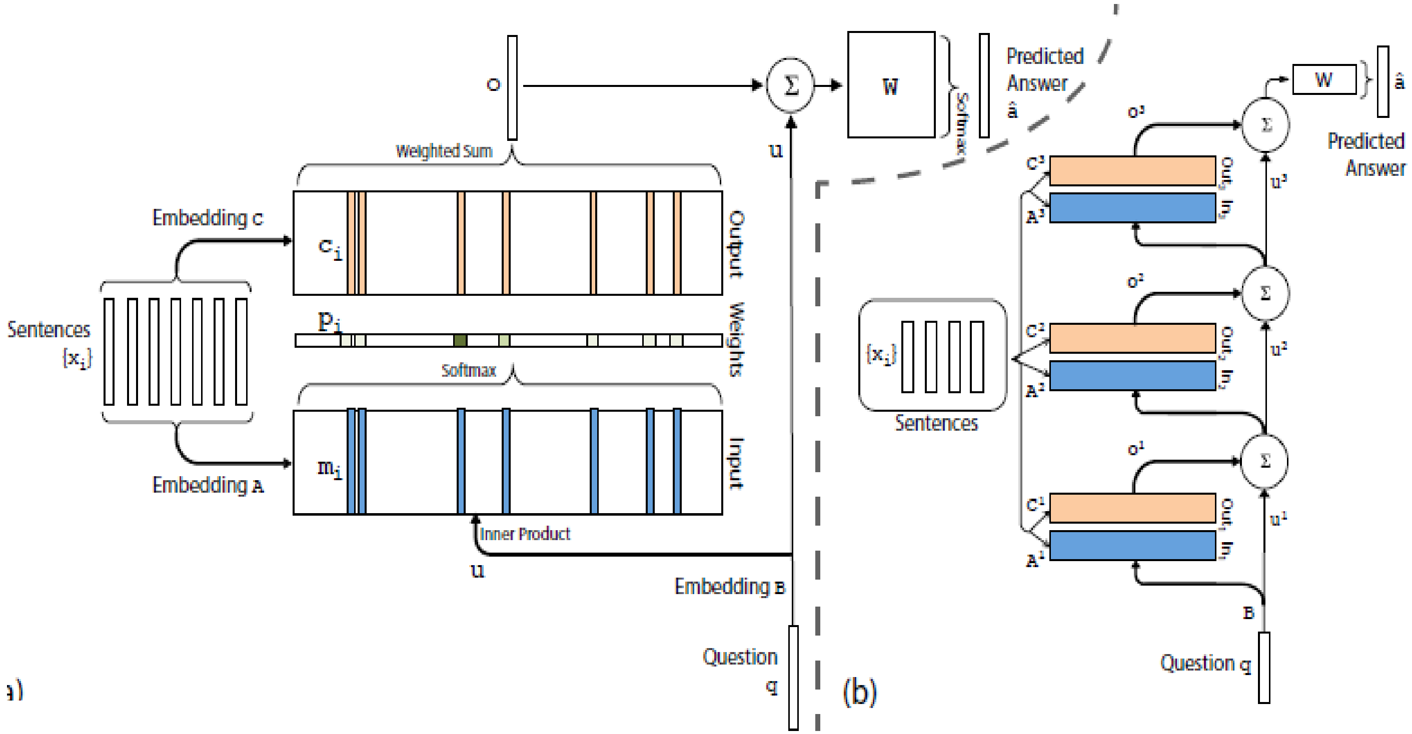
$$m_i = Ax_i$$
$$c_i = Cx_i$$
$$u = Bq$$

- Attention over memories

$$p_i = \mathrm{softmax}(u^T m_i)$$

- Generating (ranking) the final answer

$$o = \sum_i p_i c_i$$
$$a = \mathrm{softmax}(W(o + u))$$



Part II

Deep learning in statistical machine translation (SMT) and Conversation

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
  - Neural conversation models
- Part III: Deep Semantic Similarity Model and IR/NL Applications
- Part IV: NLU: Knowledge Base representation and Question answering
- Part V: Reinforcement learning in NLP
- Part VI: Image-language multimodal learning and inference
- Part VII: 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: $T^* = \underset{T}{\operatorname{argmax}} P(T|S)$
- Source-channel model: $T^* = \underset{T}{\operatorname{argmax}} P(S|T)P(T)$
- 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)

[Koehn 2009]

Phrase-based SMT

救援人员在倒塌的房屋里寻找生还者 *Chinese*

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:

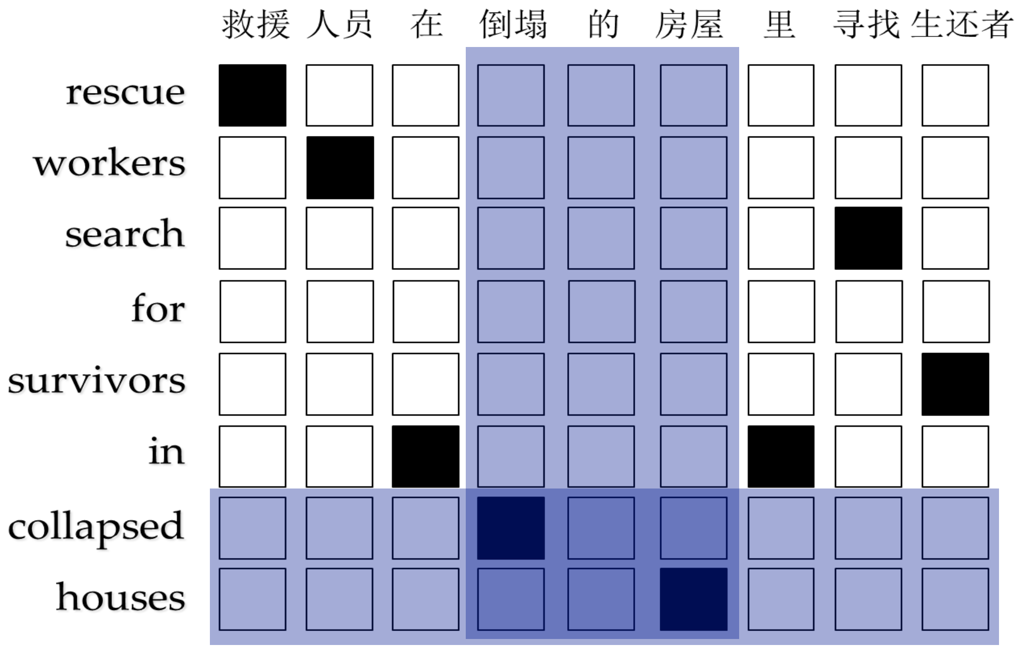
- Word Align: [Yang et al., 2013, Tamura et al., 2014, Songyot and Chiang, 2014]
- Adaptation / Topic Context: [Duh et al., 2013, Cui et al., 2014]
- Multilingual Embeddings: [Klementiev et al., 2012, Lauly et al., 2013, Zou et al., 2013, Kočiský et al., 2014, Faruqui and Dyer, 2014, Hermann and Blunsom, 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 (NMT)
  - 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]
  - NMT surpassed the best result on a WMT task [Luong et al. 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)

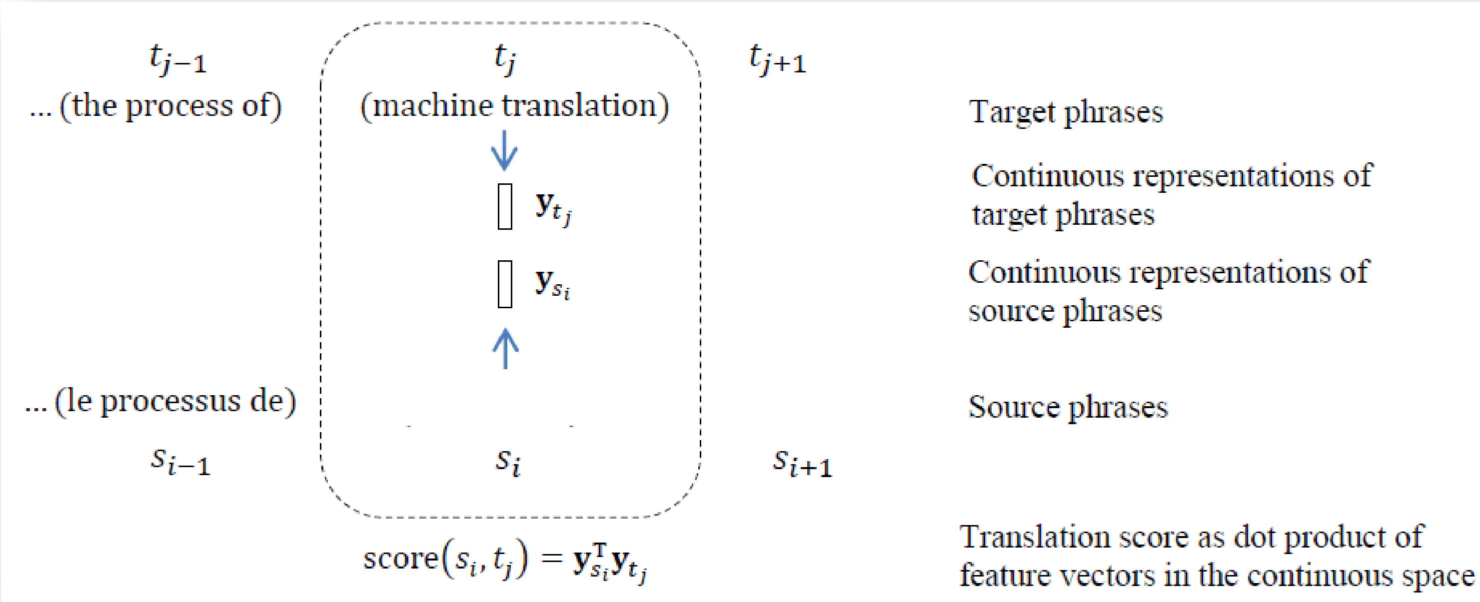
[Huang+ 13; Gao+ 14a; Gao+ 14b; Shen+ 14, Yih+ 15]

- 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, He, Yih, Deng, 2014]

- 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
  - $\text{score}(s,t) \equiv \text{sim}_{\boldsymbol{\theta}}(\mathbf{x}_s, \mathbf{x}_t) = \mathbf{y}_s^{\mathrm{T}} \mathbf{y}_t$
- 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: $\boldsymbol{\theta}^{new} = \boldsymbol{\theta} - \eta \frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$,
  - 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
  - $P(w_1 w_2 \ldots w_n) = P(w_1)P(w_2|w1)\prod_{i=2\ldots n} P(w_i|w_{i-2}w_{i-1})$
  - Cannot capture long-distance dependency

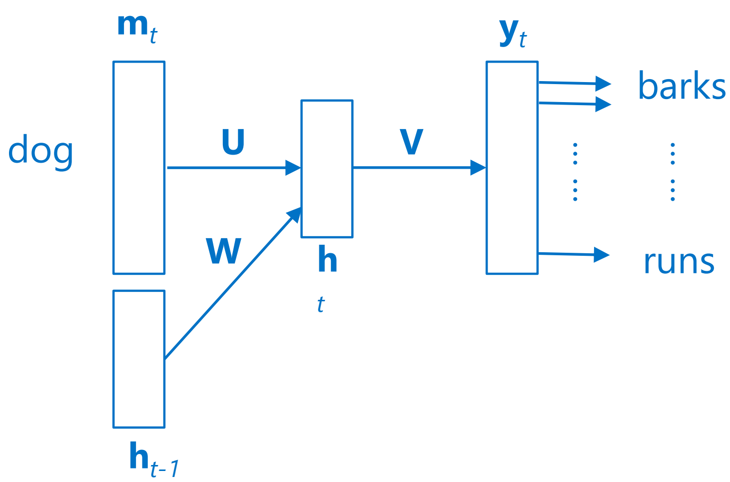
the dog of our neighbor barks

- 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 $P(w_3\|w_1w_2)$	8×10^{12}
4-gram $P(w_4\|w_1w_2w_3)$	1.6×10^{17}

[Manning & Schütze 99]

Recurrent neural net for language modeling



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

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

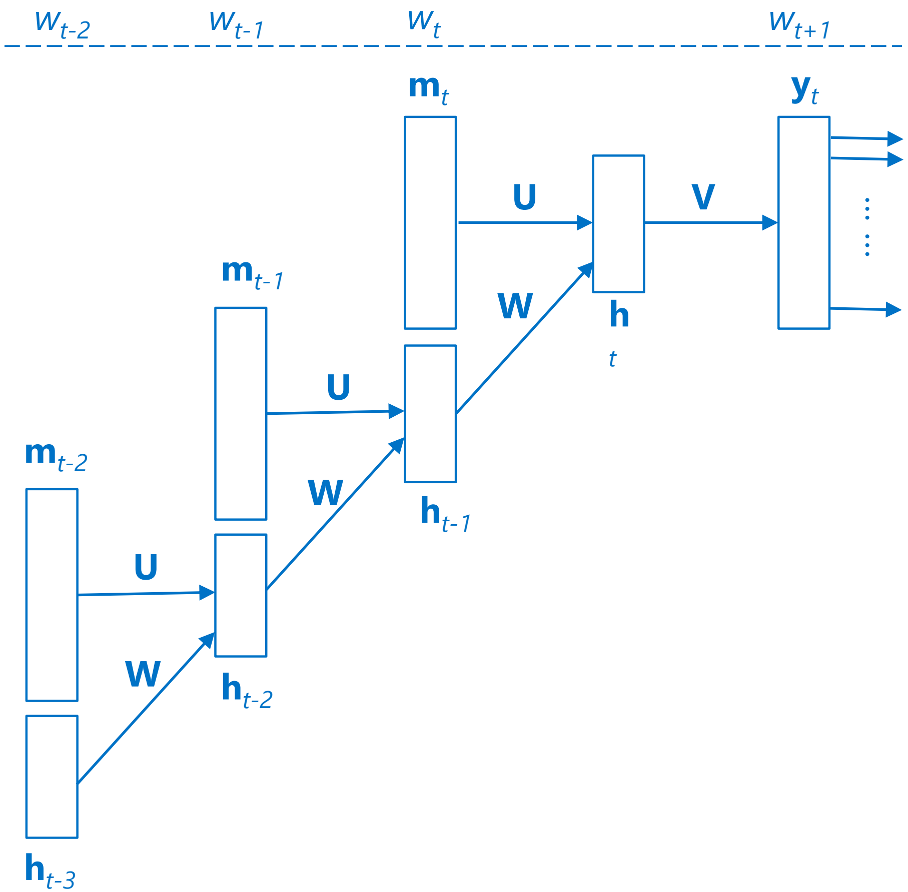
$g(.)$ is called the *softmax* function

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]

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]

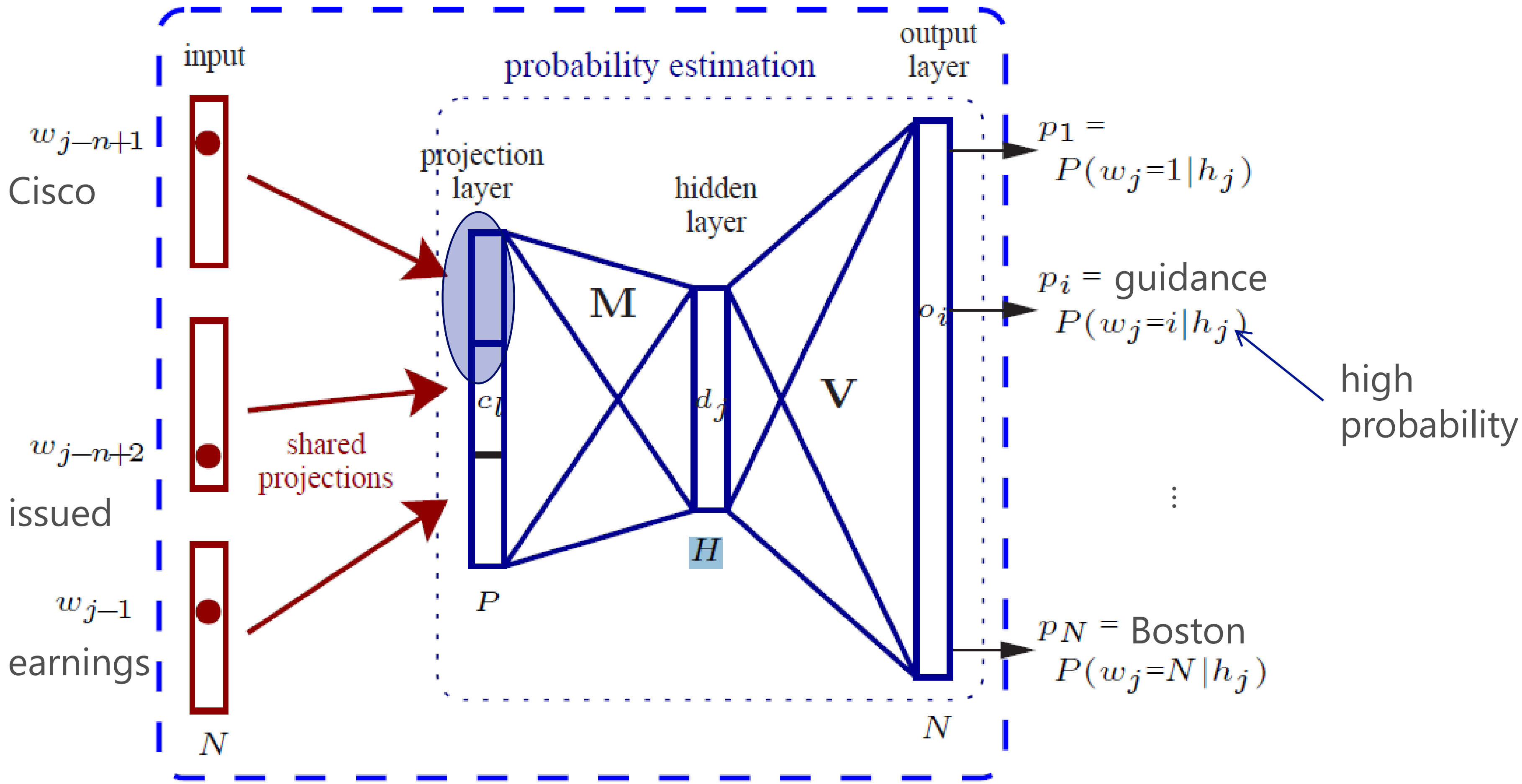
- RNN LMs require history going back to start-of-sentence. Harder to do dynamic programming.
- To score new words, each decoder state needs to maintain *h*. For recombination, merge hypotheses by traditional n-gram context and the best *h*

	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

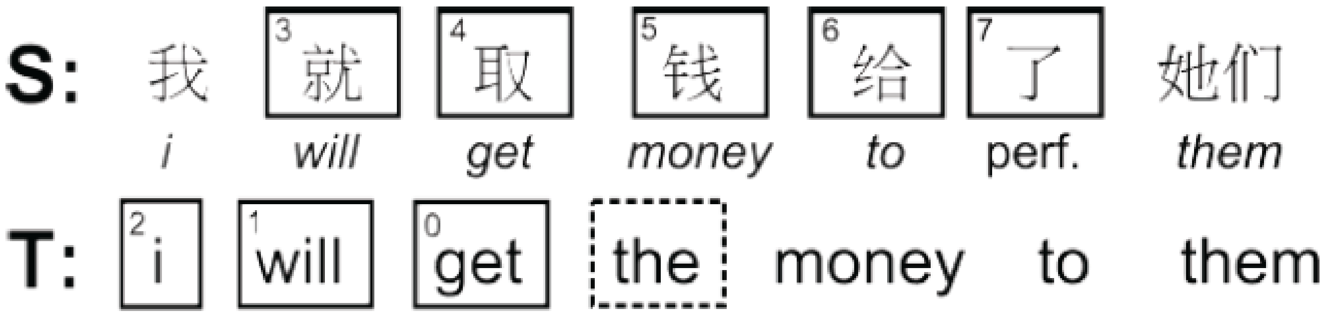
Joint model: language model with source

- $P(t_i|t_{i-2}t_{i-1}, S)$
- How to model 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.

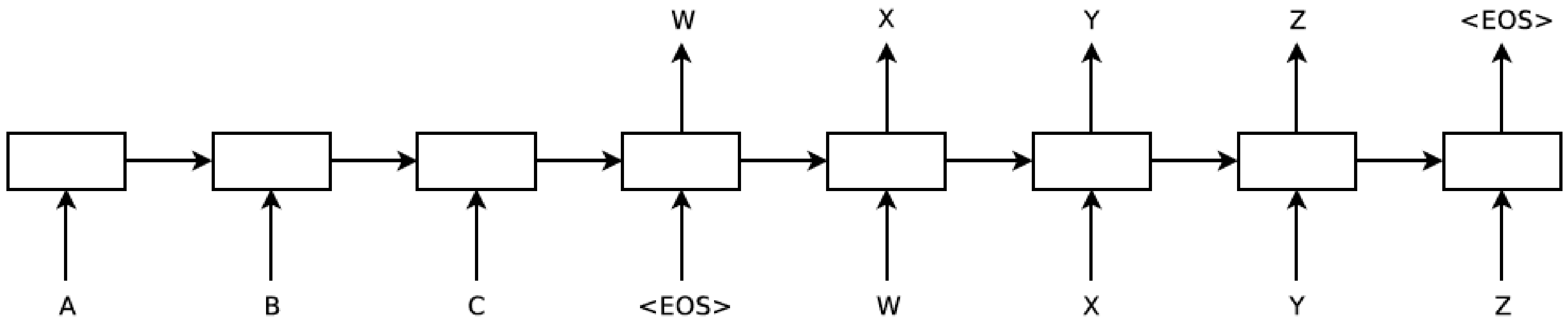
Neural machine translation
[Sutskever+ 14; Cho+ 14; Bahdanau+ 15]

- 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]

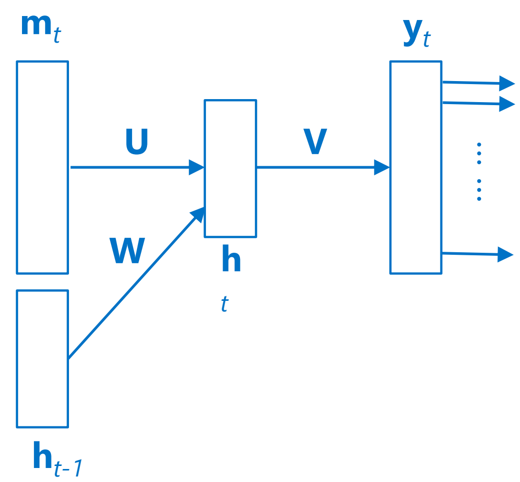
- "A B C" is source sentence; "W X Y Z" is target sentence

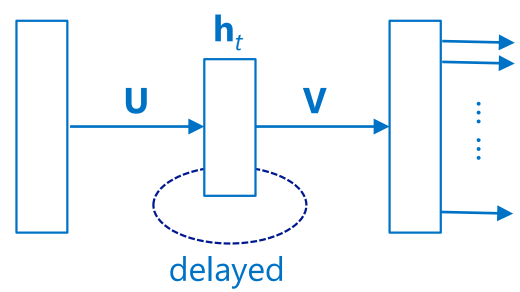


- Treat MT as general sequence-to-sequence transduction
  - Read source; accumulate hidden state; generate target
  - <EOS> token stops the recurrent process
  - In practice, read source sentence in reverse leads to better MT results
- Train on bi-text; optimize target likelihood using SGD

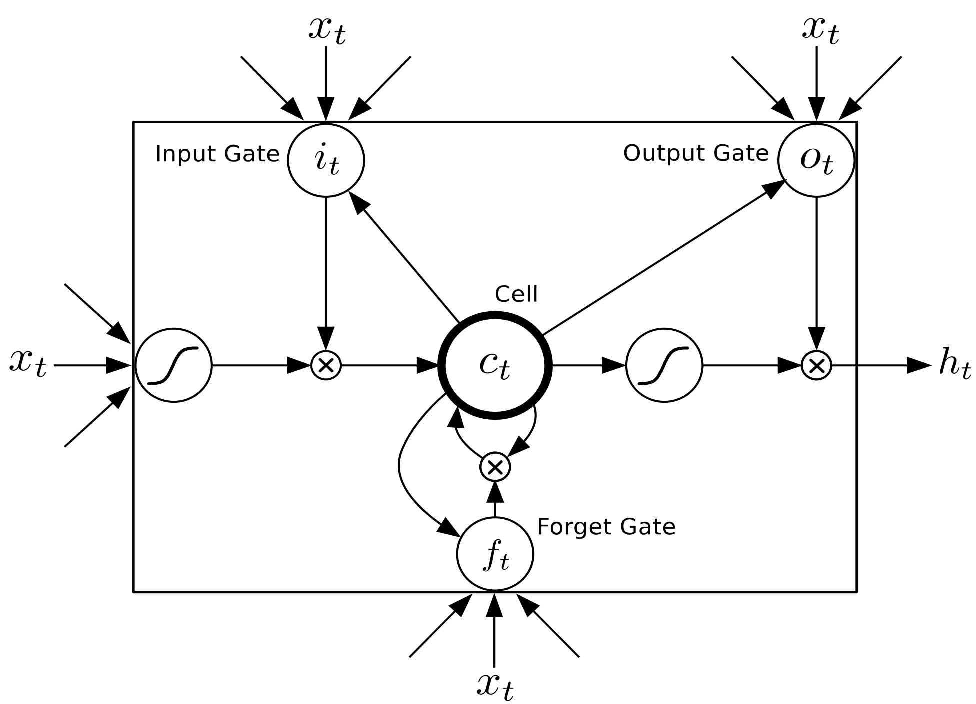
Potentials and difficulties of RNN

- 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]

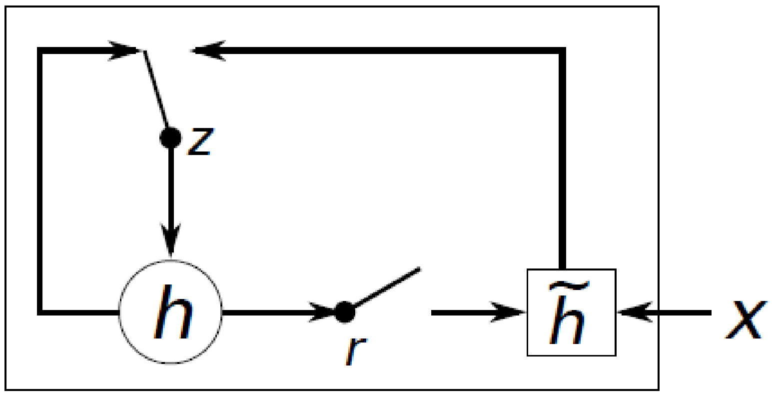


Figure 2: An illustration of the proposed hidden activation function. The update gate z selects whether the hidden state is to be updated with a new hidden state $\tilde{h}$. The reset gate r decides whether the previous hidden state is ignored. See

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

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

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

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

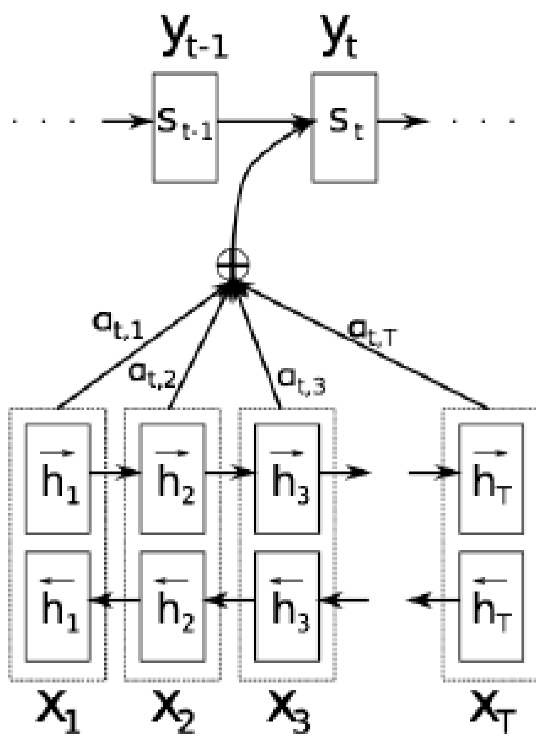
Joint learning to align and translate

- 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**



MSR's Neural Conversation Engine

[Sordoni+ 15; Li+ NAACL16; Li+ ACL16]

Neural model learns the general shape of conversation, and the system output is situationally appropriate and coherent…

H: hi
M: how are you ?
H: not bad
M: what 's wrong ?
H: nothing really
M: wan na talk about it ? i 'm here if you wan na talk x
H: well , i 'm just tired
M: me too , i 'm here if you wan na talk
H: thanks , what are you doing ?
M: you 're welcome and i 'm watching tv and you ?
H: on netflix ?
M: yeah , i 'm watching it on netflix .
H: what are you watching ?
M: a tv show .
H: nice , did you have dinner already ?
M: no , not yet . what about you ?

A more goal-oriented example. No grounding into a real calendar, but the "shape" of the conversation is fluent and plausible…

H: would thursday afternoon work sometime ?
M: yeah , sure . just let me know when you 're free .
H: after lunch is probably the best time
M: okay , sounds good . just let me know when you 're free .
H: would 2 pm work for you ?
M: works for me .
H: well let 's say 2 pm then i 'll see you there
M: sounds good .

- Generating responses vs. retrieving responses
- Easy to incorporate contextual info via embedding
  - User profile – personalized conversation
  - knowledge bases – grounded conversation
- The engine is E2E learned from conversation experience
  - Learning a goal-oriented conversation engine via RL

Neural Response Generation: The Blandness Problem

How was your weekend?

I don't know.

What did you do?

I don't understand what you are talking about.

This is getting boring…

Yes that's what I'm saying.

Blandness problem: cause and remedies

[Li et al., NAACL 2016]

Common ML objective (maximum likelihood)

(whatever the user says) $\xrightarrow{p(\text{target}|\text{source})}$

I don't know.

I don't understand...

That's what I'm saying

Mutual information objective:

(whatever the user says) $\xrightarrow{p(\text{target}|\text{source})}$ I don't know.

(whatever the user says) $\xleftarrow[p(\text{source}|\text{target})]{}$ I don't know.

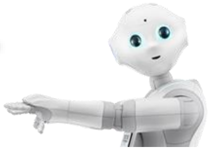
Beyond blandness: Examples

Wow sour starbursts really do make **your mouth water**... mm drool.
Can I have one?

Of course you can! They're **delicious**!

Milan apparently **selling Zlatan** to balance the books... **Where next**, Madrid?

I think he'd be a **good signing**.



'tis a fine **brew** on a day like this! Strong though, **how many** is sensible?

Depends on how much you **drink**!

Well he was on in Bromley a while ago... **still touring**.

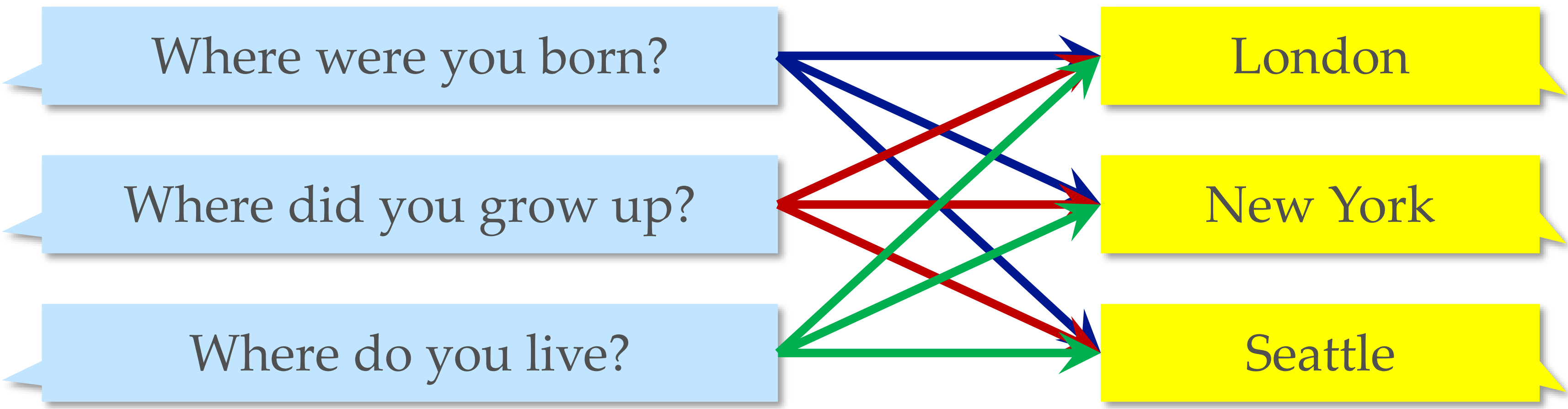
I've never **seen him live**.

A Persona-Based Conversational Model

Why? Motivation is to model:

- personal background
- behavioral and stylistic differences (e.g., introvert vs. extrovert)

Better at "explaining away" conversational data:



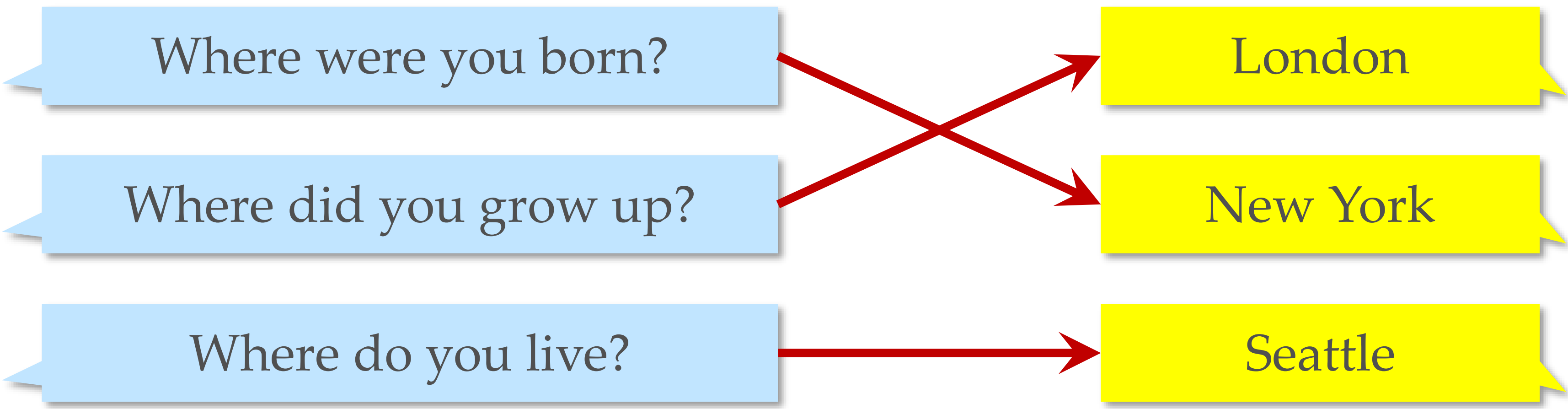
Conversation data is badly entangled! (N-to-1, 1-to-N)

A Persona-Based Conversational Model

Why? Motivation is to model:

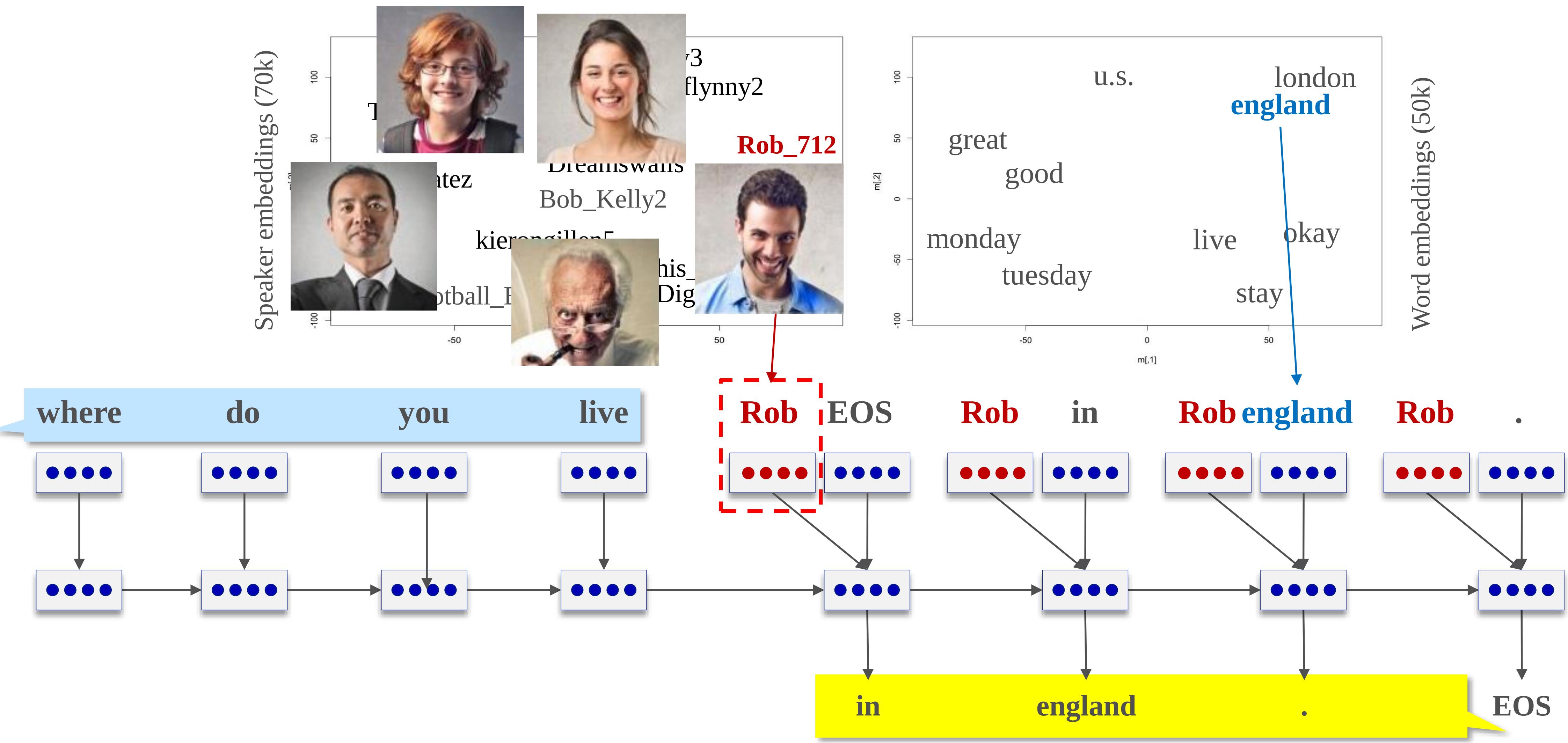
- personal background
- behavioral and stylistic differences (e.g., introvert vs. extrovert)

Better at "explaining away" conversational data:



$p(T|S,$ PERSON_943 $)$

Personalized Response Generation



Sample outputs

User1	
message	Where is your hometown?
response	I'm from England.
message	Where are you from?
response	I'm from England.
message	In which city do you live now?
response	I live in London.
message	In which country do you live now?
response	I live in England.
message	What is your major?
response	Business. You?
message	What did you study in college?
response	I did business studies.
message	How old are you?
response	I'm 18.
message	What is your age?
response	I'm 18.

User2	
message	Where is your hometown?
response	I live in Indonesia.
message	Where are you from?
response	England, you?
message	In which city do you live now?
response	I live in Jakarta.
message	In which country do you live now?
response	I live in Indonesia.
message	What is your major?
response	Business, you?
message	What did you study in college?
response	Psychology, you?
message	How old are you?
response	I'm 18.
message	What is your age?
response	I'm 16.

Sample outputs

User1	
message	Where is your hometown?
response	I'm from England.
message	Where are you from?
response	I'm from England.
message	In which city do you live now?
response	I live in London.
message	In which country do you live now?
response	I live in England.
message	What is your major?
response	Business. You?
message	What did you study in college?
response	I did business studies.
message	How old are you?
response	I'm 18.
message	What is your age?
response	I'm 18.

User2	
message	Where is your hometown?
response	I live in Indonesia.
message	Where are you from?
response	England, you?
message	In which city do you live now?
response	I live in Jakarta.
message	In which country do you live now?
response	I live in Indonesia.
message	What is your major?
response	Business, you?
message	What did you study in college?
response	Psychology, you?
message	How old are you?
response	I'm 18.
message	What is your age?
response	I'm 16.

Sample outputs

User1	
message	Where is your hometown?
response	I'm from England.
message	Where are you from?
response	I'm from England.
message	In which city do you live now?
response	I live in London.
message	In which country do you live now?
response	I live in England.
message	What is your major?
response	Business. You?
message	What did you study in college?
response	I did business studies.
message	How old are you?
response	I'm 18.
message	What is your age?
response	I'm 18.

User2	
message	Where is your hometown?
response	I live in Indonesia.
message	Where are you from?
response	England, you?
message	In which city do you live now?
response	I live in Jakarta.
message	In which country do you live now?
response	I live in Indonesia.
message	What is your major?
response	Business, you?
message	What did you study in college?
response	Psychology, you?
message	How old are you?
response	I'm 18.
message	What is your age?
response	I'm 16.

Interim summary

- Part I: Background
  - A brief history of deep neural networks (DNN)
  - An example of neural models for query classification
  - Different forms of DNN for classification/ranking/generation tasks
- Part II: Deep learning in statistical machine translation and conversation
  - Review of SMT and DNN in SMT
  - Deep semantic translation models
  - Recurrent neural language models
  - Neural network joint models
  - Neural machine translation (Seq2Seq models)
  - Neural conversation models (Seq2Seq models)
- Part III: Learning semantic representations
- Part IV: Natural language understanding
- Part V: Conclusion

Part III

Deep Semantic Model and IR/NL Applications

Deep Semantic Model and IR/NL Applications

- Deep semantic similarity model (DSSM)
- DSSM for Information Retrieval
- DSSM for entity ranking
- Semantic document classification & sentiment analysis

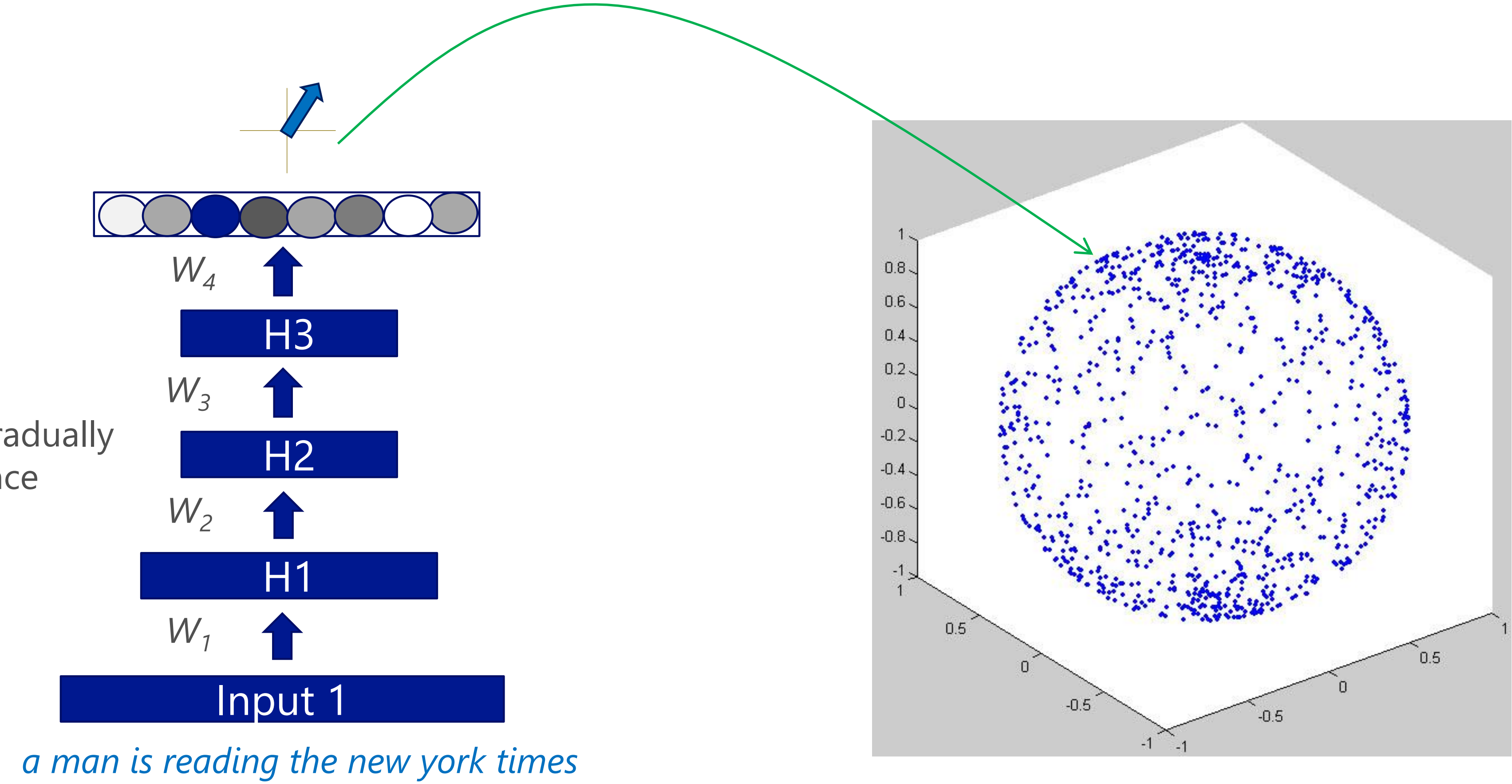
Learning continuous semantic representations for natural language

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

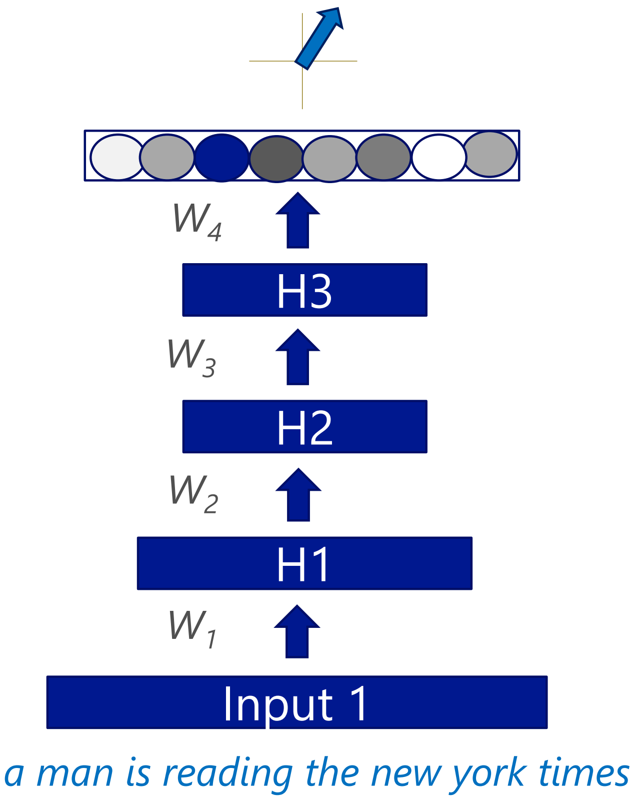
Raw text, e.g., a sequence of words



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*
…		

The supervision problem:



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**)

project the whole sentence to a continuous semantic space – e.g., *Sentence to Vector*.

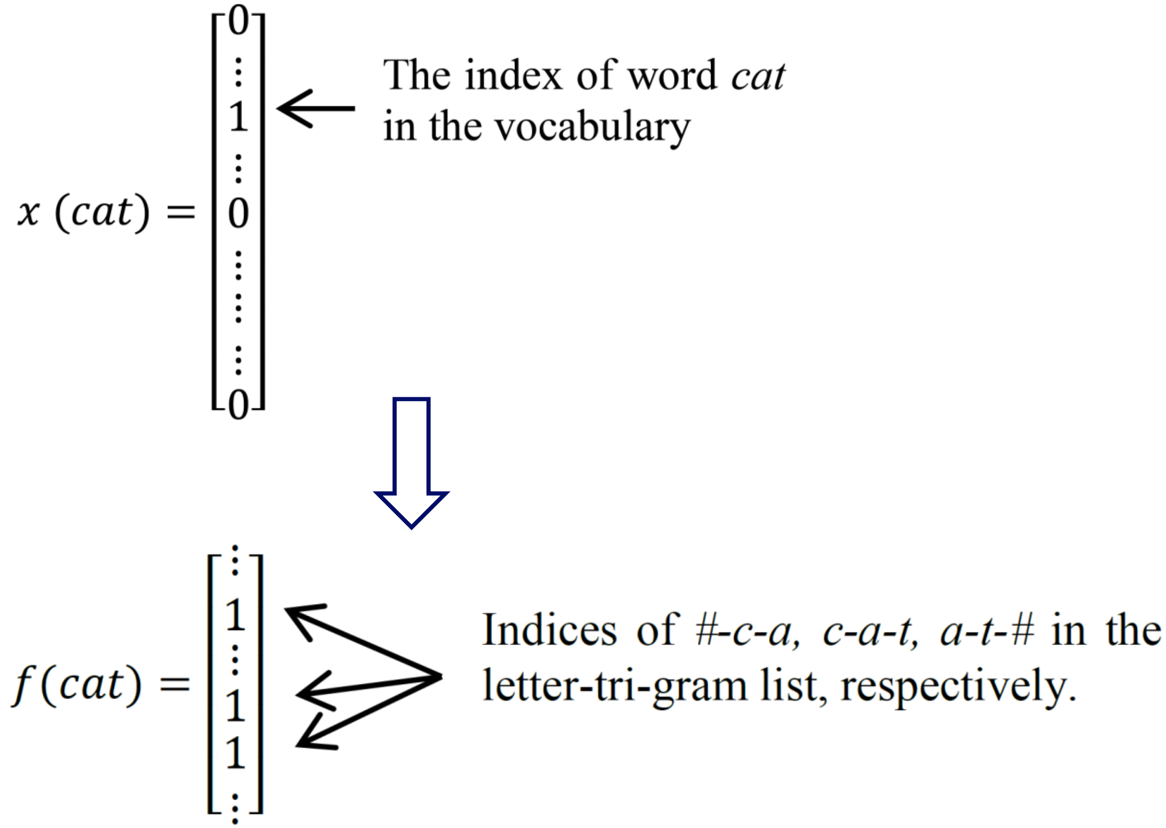
The DSSM is built upon **characters** (rather than words) for scalability and generalizability

The DSSM is trained by optimizing an **similarity-driven** objective

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

Character-level coding (a.k.a. word hashing)

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



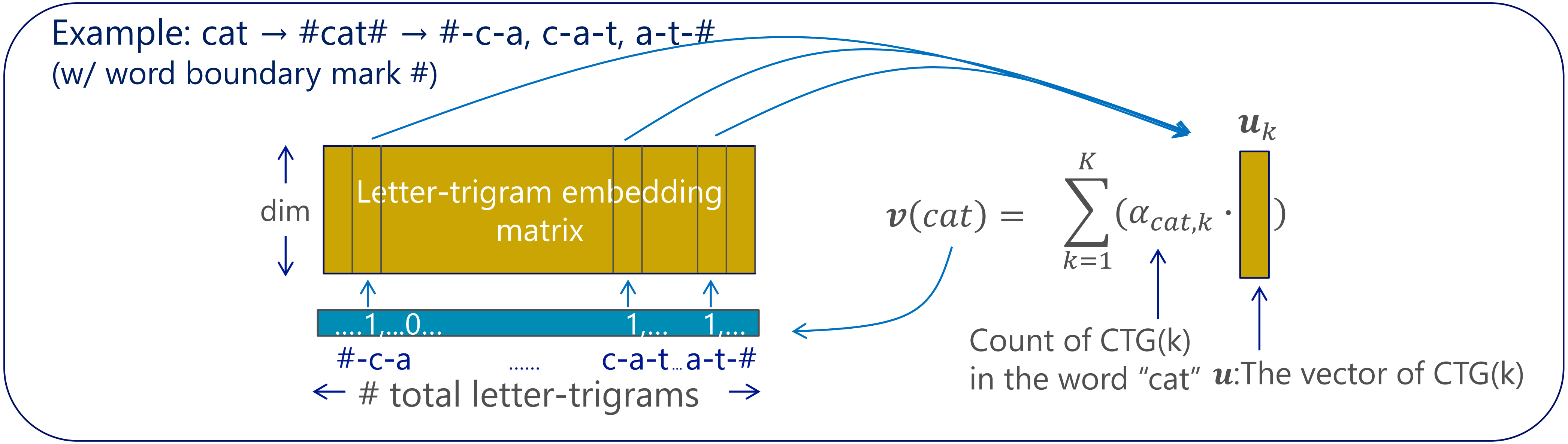
What if different words have the same word hashing code (collision)?

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

Learning character-trigram embedding vectors

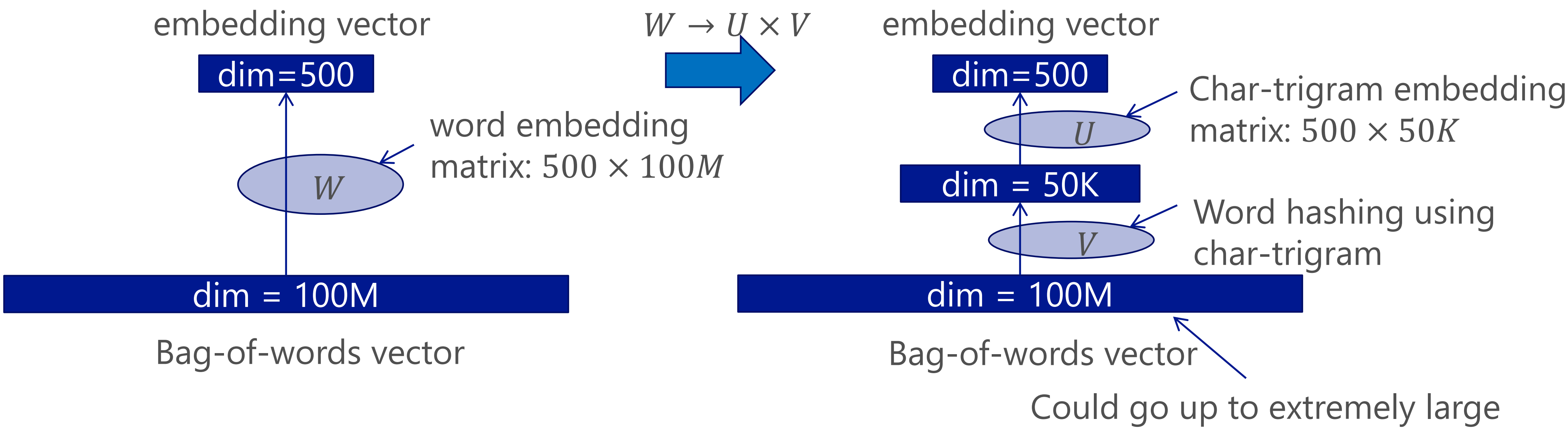
Learn **one vector per character-trigram** (CTG), the encoding matrix is a fixed matrix

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



DSSM: built at the character-level

Decompose *any* word into set of context-dependent characters



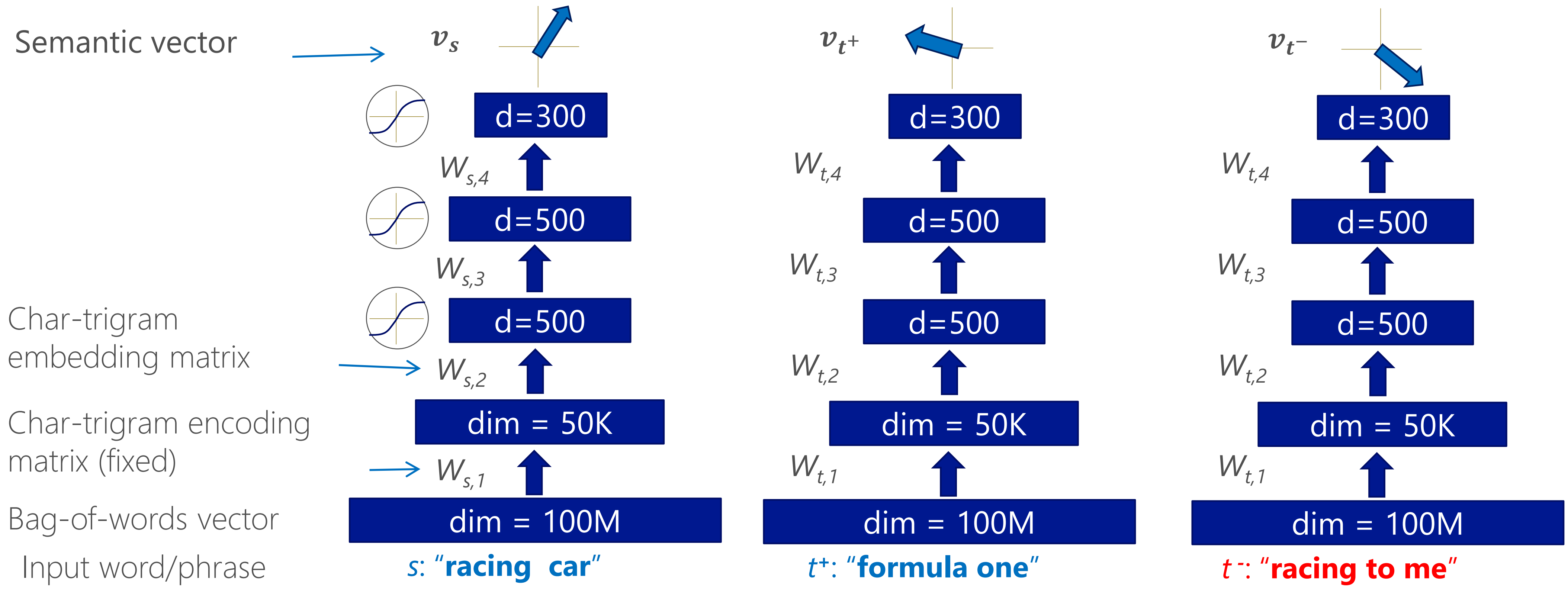
Preferable for large scale NL tasks

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

DSSM: a similarity-driven Sent2Vec model

Initialization:

Neural networks are initialized with random weights

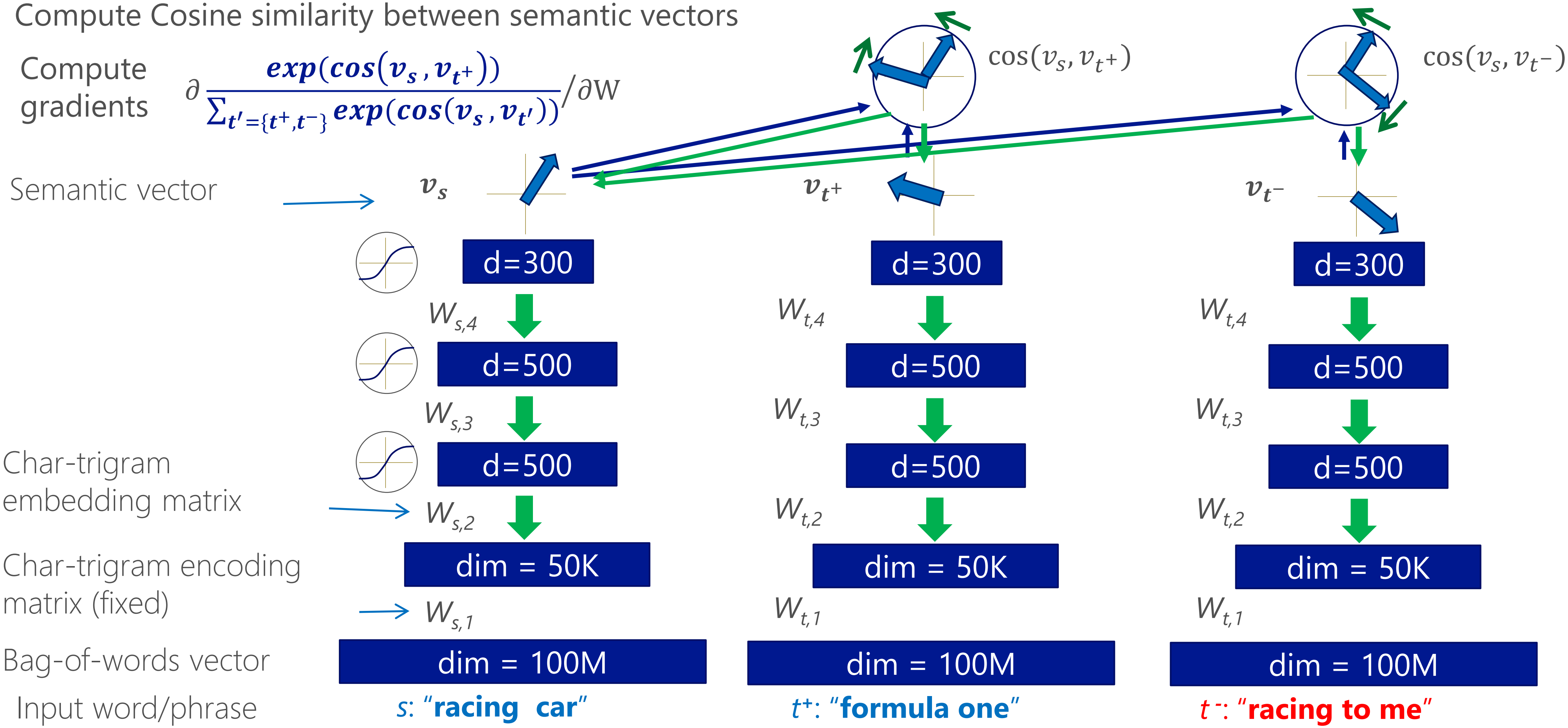


DSSM: a similarity-driven Sent2Vec model

Training:

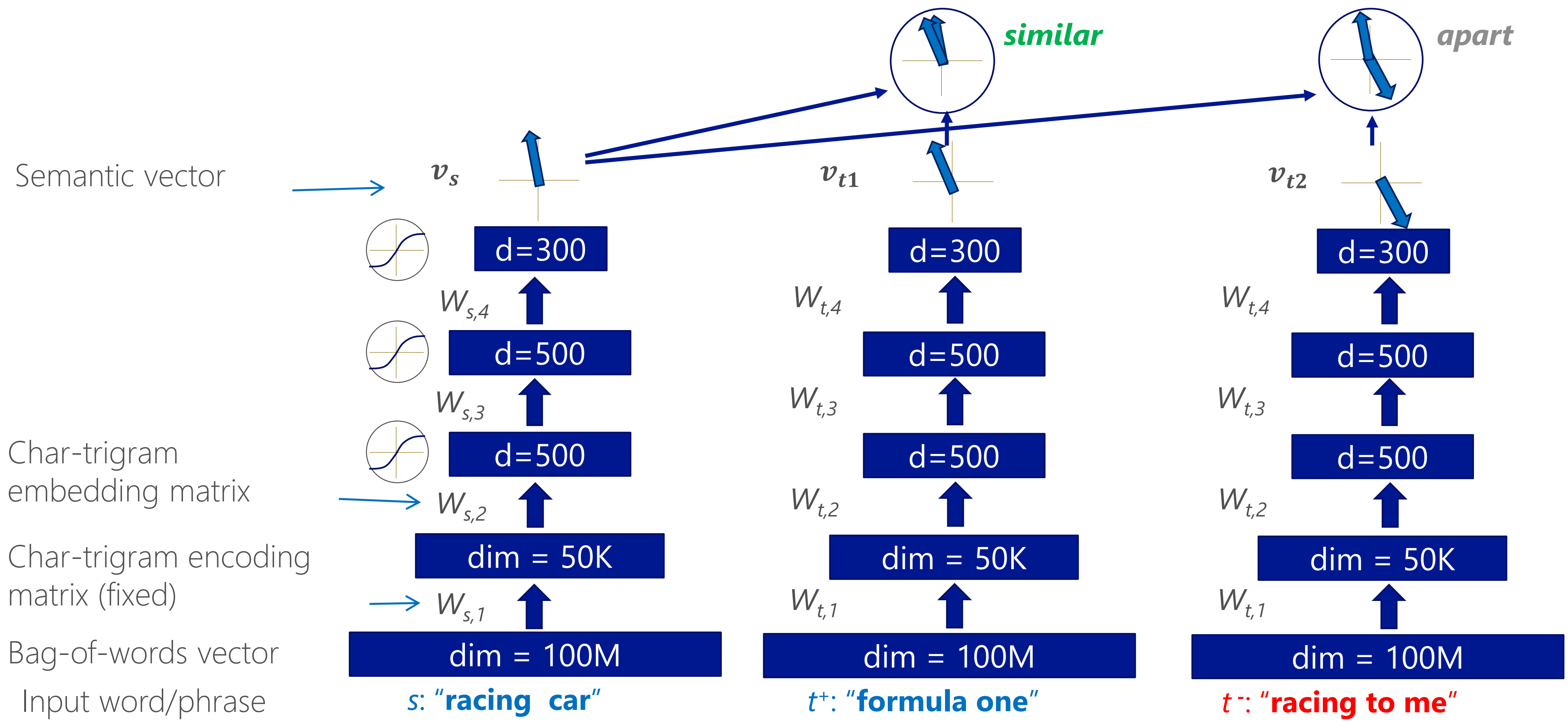
Compute Cosine similarity between semantic vectors

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



DSSM: a similarity-driven Sent2Vec model

Runtime:



Training objectives

Objective: cosine similarity based loss

Using web search as an example:

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

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

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

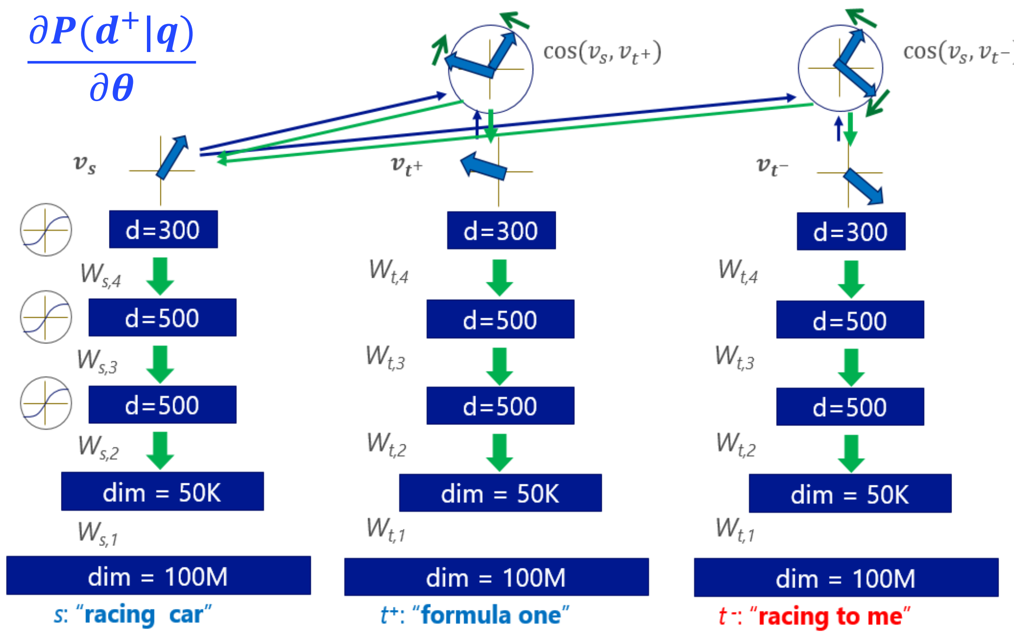
$$v_{\boldsymbol{\theta}}(d) = \sigma(W_{t,4} \times \sigma(W_{t,3} \times \sigma(W_{t,2} \times ltg(d)))$$

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]

Using Convolutional Neural Net in DSSM

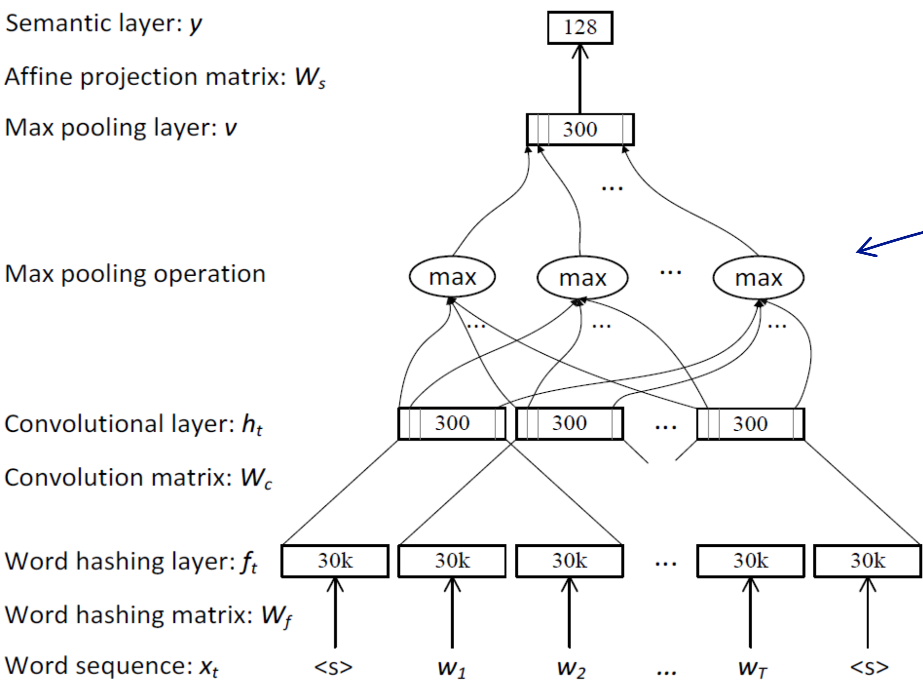


Figure 1: Illustration of the C-DSSM. A convolutional layer with the window size of three is illustrated.

Model local context at the convolutional layer

Model global context at the pooling layer

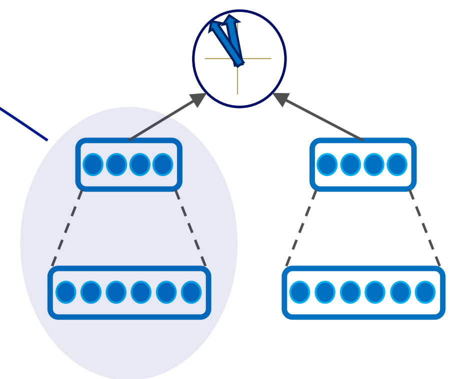


Figure credit [Shen, He, Gao, Deng, Mesnil, WWW2014]

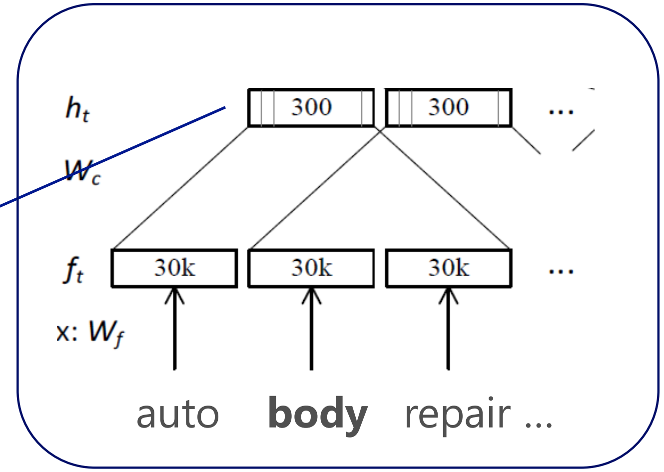
Strong performance on many NLP tasks

Information Retrieval: [Shen, He, Gao, Deng, Mesnil, WWW2014 & CIKM2014], **Entity Ranking**: [Gao, Pantel, Gamon, He, Deng, Shen, EMNLP2014], **Question answering**: [Yih, He, Meek, ACL2014; Yih, Chang, He, Gao, ACL2015], **Recommendation** [Elkahky, Song, He, WWW2015], **Spoken language understanding** [Chen, Hakkani-Tür, He, ICASSP2016]...

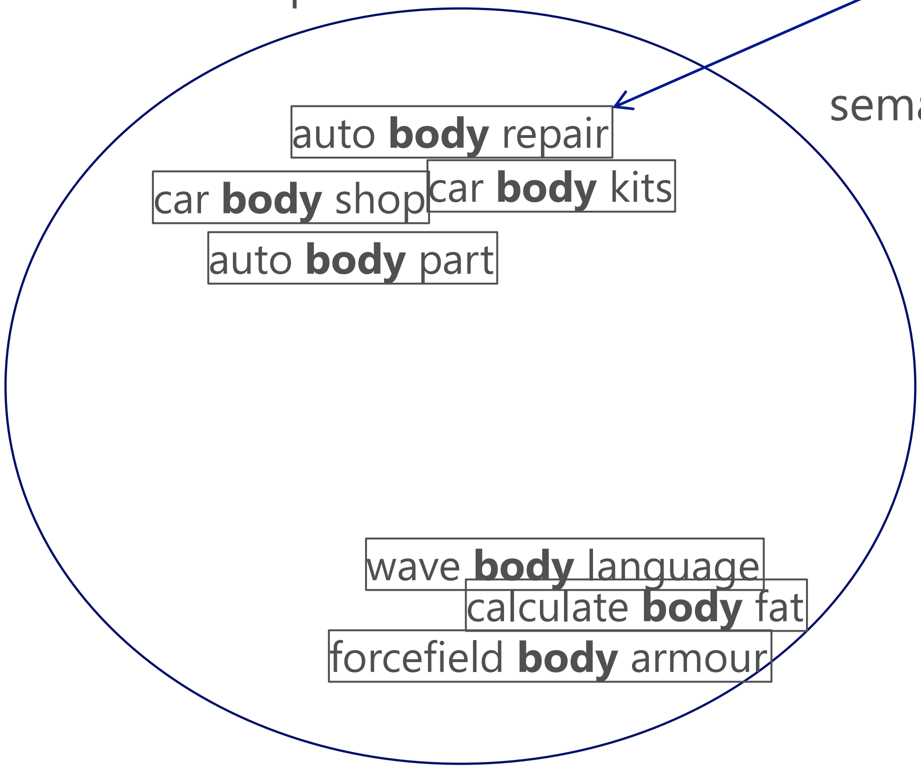
– What does the model learn at the convolutional layer?

Capture the local context dependent word sense

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



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



semantic space

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

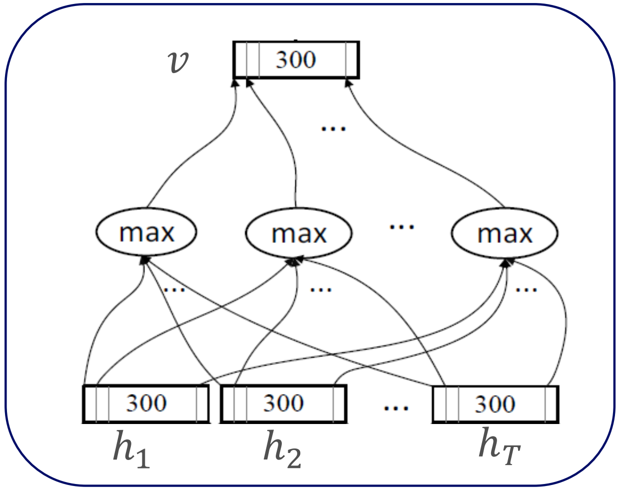
car **body** shop	**cosine similarity**
car **body** kits	0.698
auto **body** repair	0.578
auto **body** parts	0.555
wave **body** language	0.301
calculate **body** fat	0.220
forcefield **body** armour	0.165

high similarity

low similarity

CDSSM: What happens at the max-pooling layer?

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



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

where $i = 1, \ldots, 300$

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
  - Mine semantically-similar text pairs from Search Logs, e.g., 30 Million (Query, Document) Click Pairs

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

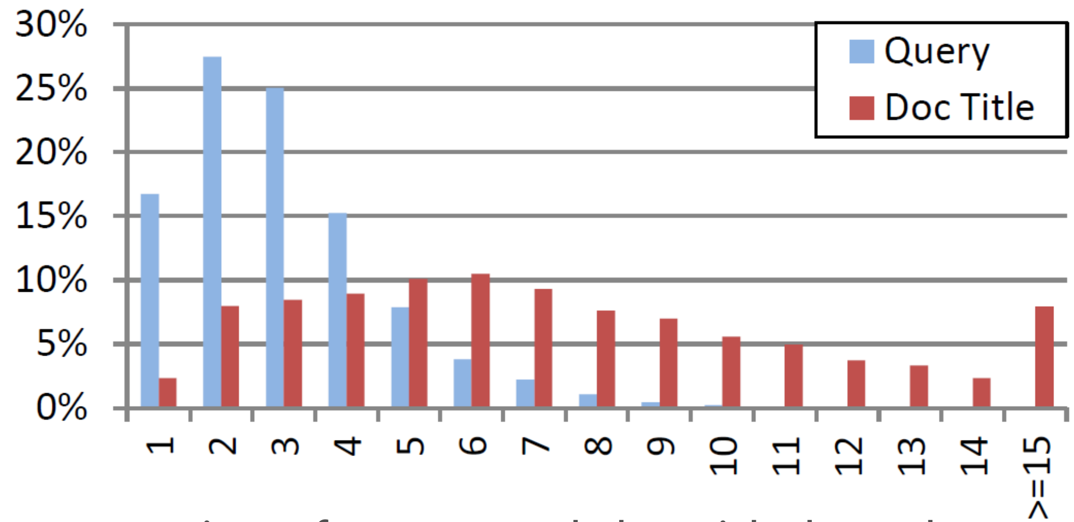
[Gao, He, Nie, CIKM2010]

Experimental Setting

- 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

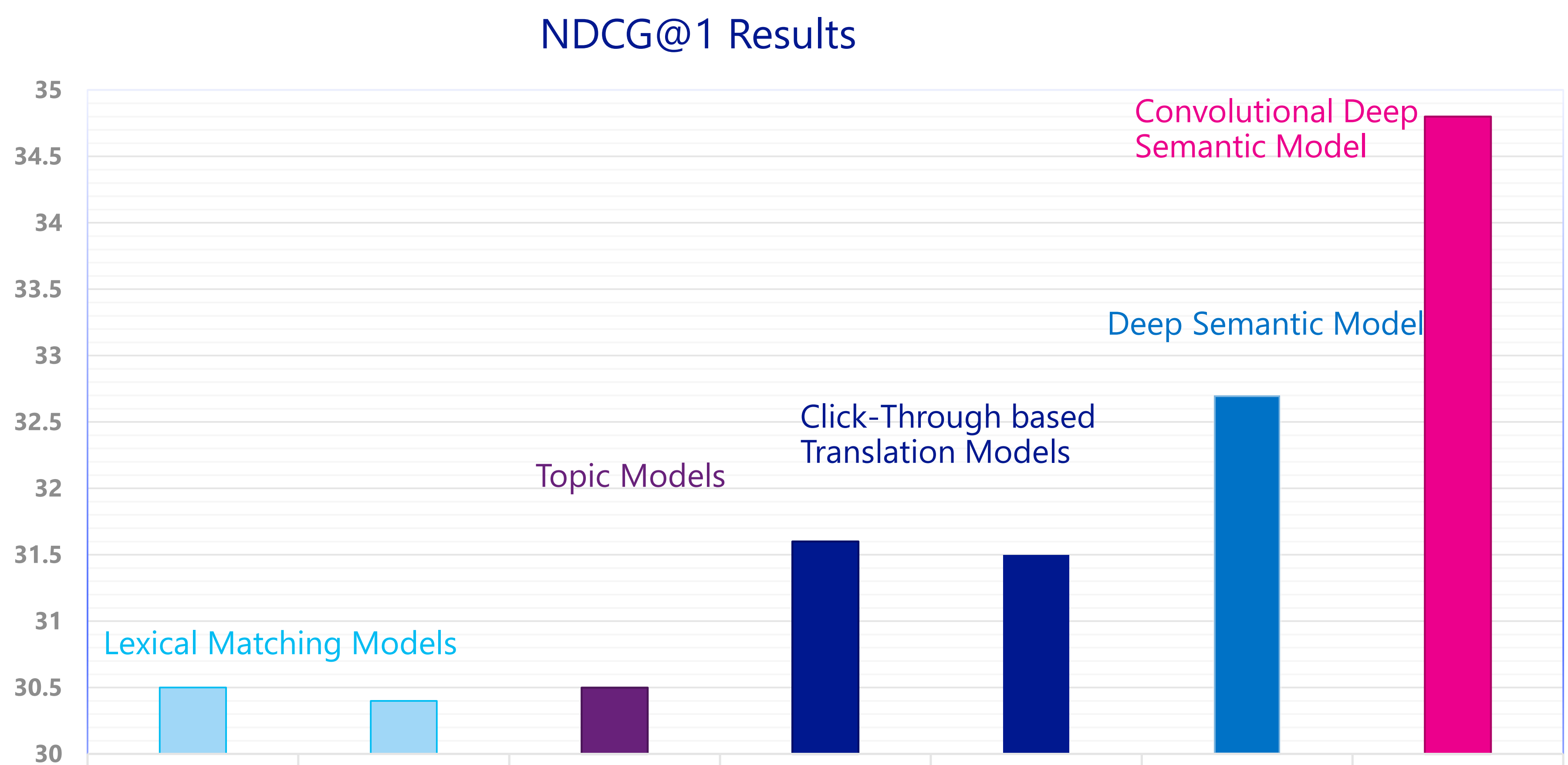
- Using NVidia GPU K40 for training



Dist. of query and doc title length

Results

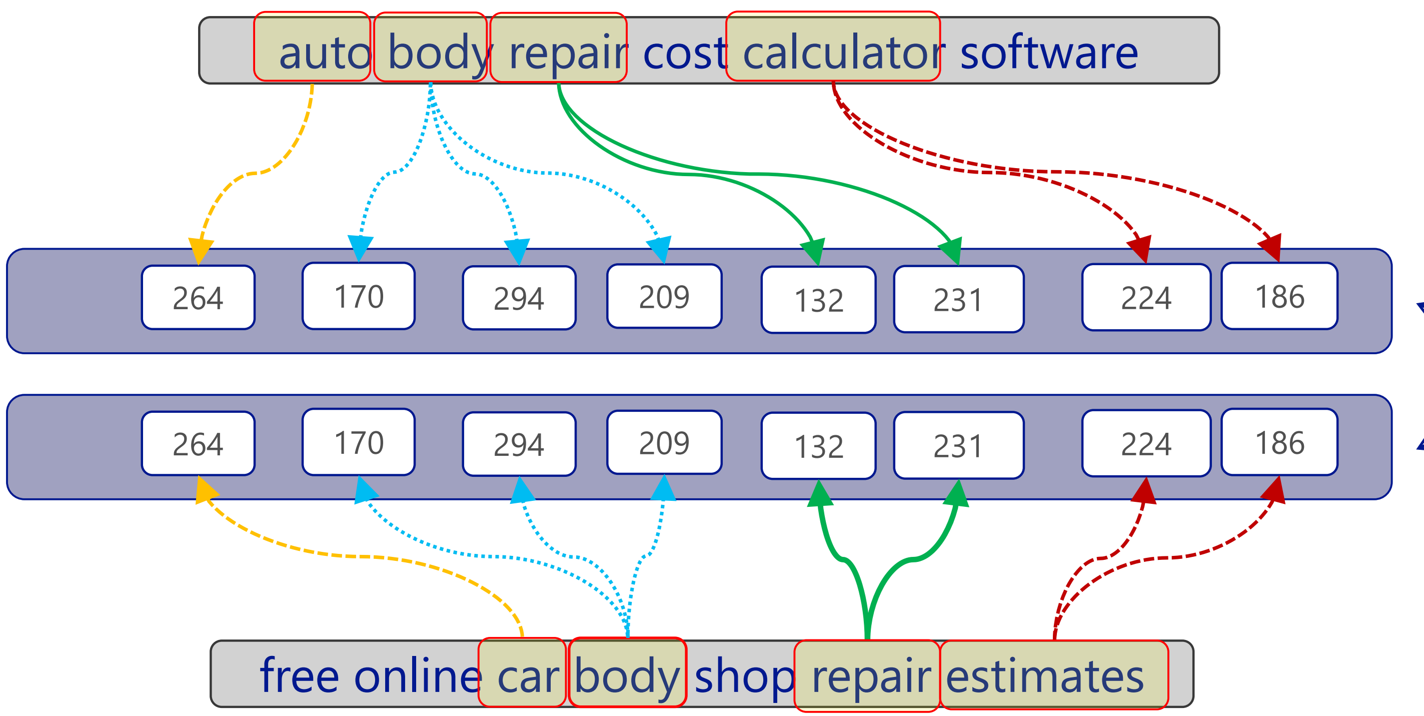
[Shen et al. CIKM2014]



	1	2	3	4	5	6	7
■ NDCG@1	30.5	30.4	30.5	31.6	31.5	32.7	34.8

Example: semantic matching

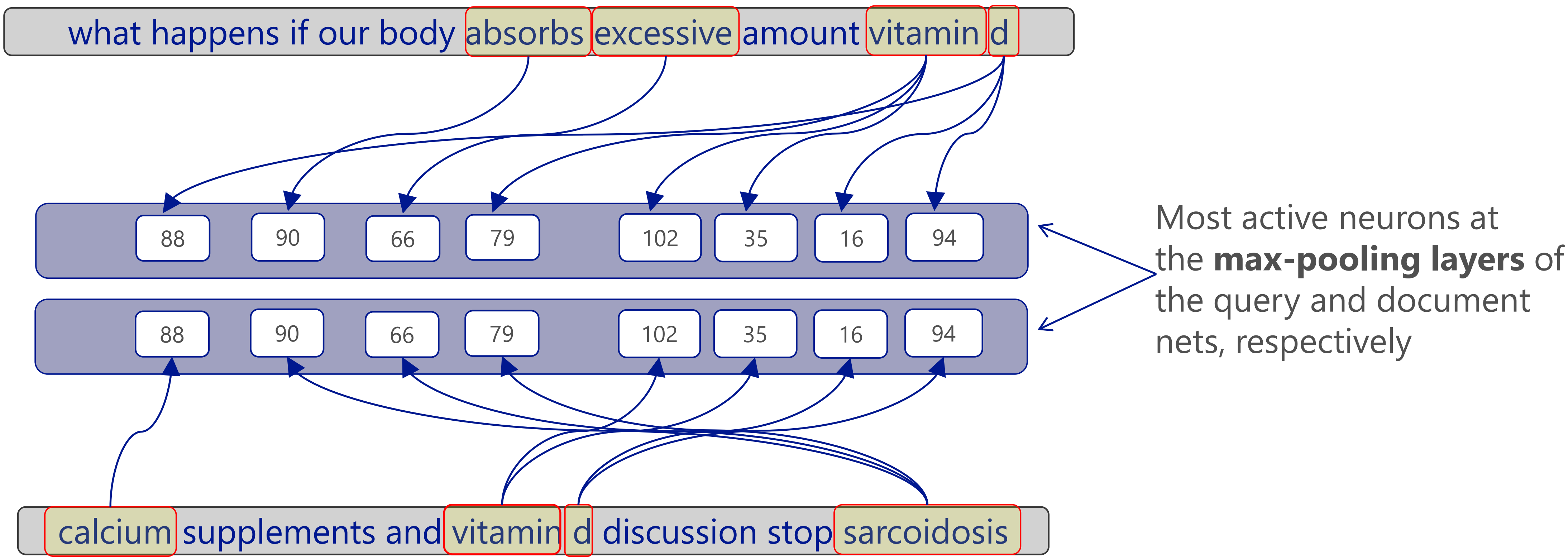
- Semantic matching of query and document



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

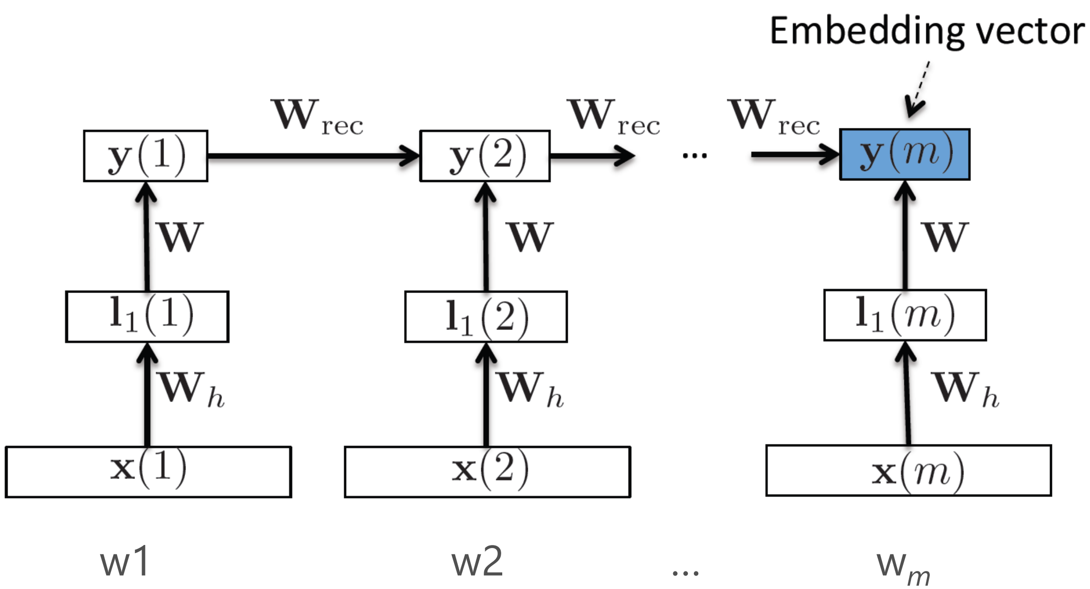
More complex semantic matching example

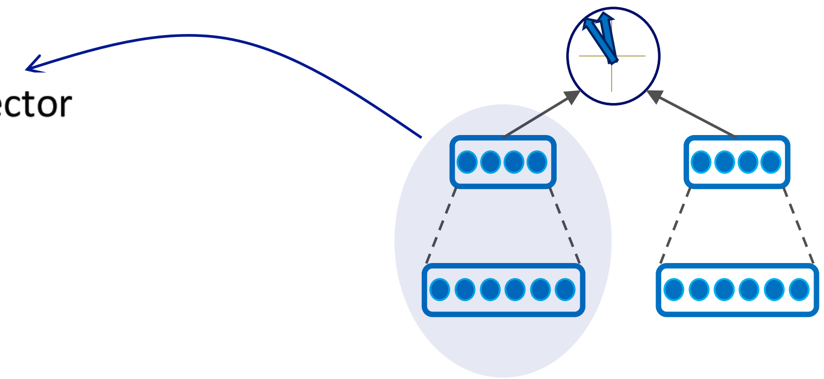
sarcoidosis is a disease, a symptom is excessive amount of calcium in one's urine and blood. So medicines that increase the absorbing of calcium should be avoid. While **Vitamin d** is closely associated to **calcium absorbing**.
We observed that "sarcoidosis" in the document title and "absorbs" "excessive" and "vitamin (d)" in the query have high activations at neurons 90, 66, 79, indicating that the model knows that **"sarcoidosis" share similar semantic meaning with "absorbs" "excessive" "vitamin (d)", collectively**.



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



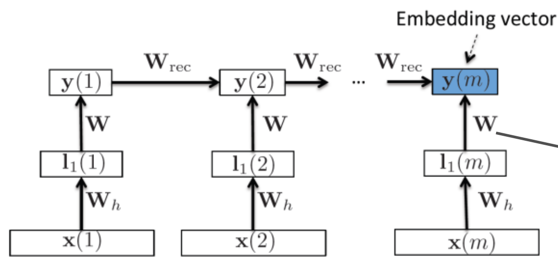


[Palangi, Deng, Shen, Gao, He, Chen, Song, Ward, 2015]

Using LSTM cells

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

[Hochreiter and J. Schmidhuber, 1997]



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

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

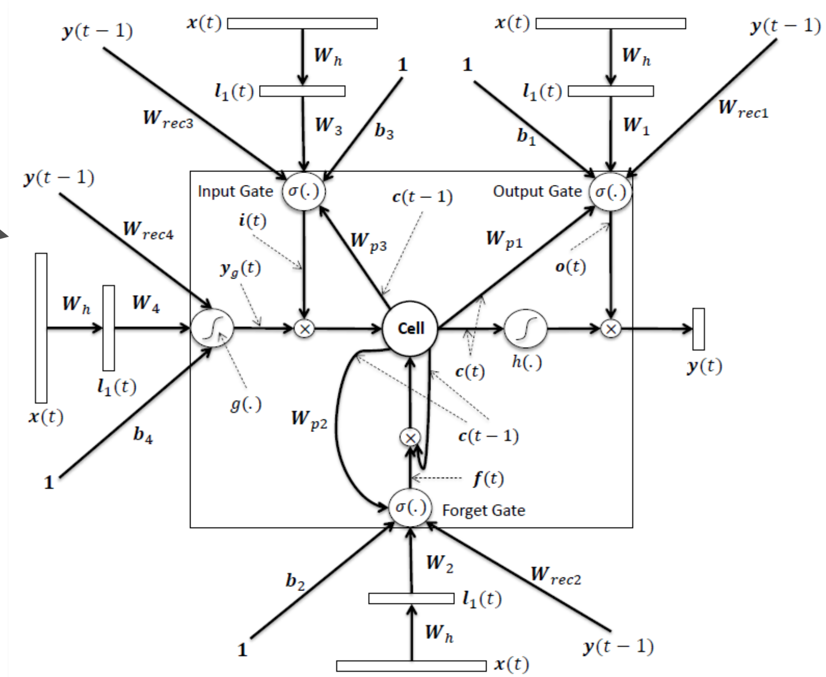


Figure 2. The basic LSTM architecture used for sentence embedding

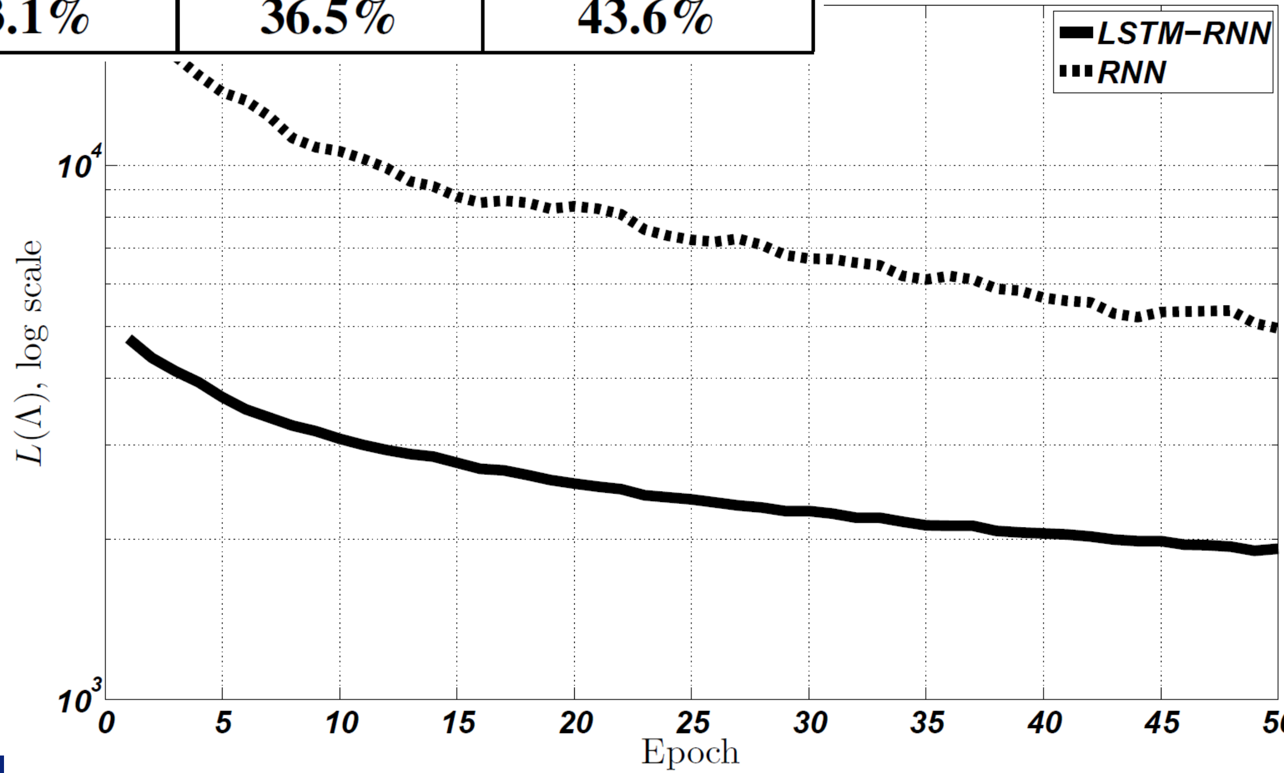
[Palangi, Deng, Shen, Gao, He, Chen, Song, Ward, Deep Sentence Embedding Using the LSTM network: Analysis and Application to IR, IEEE TASL, 2016]

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%**

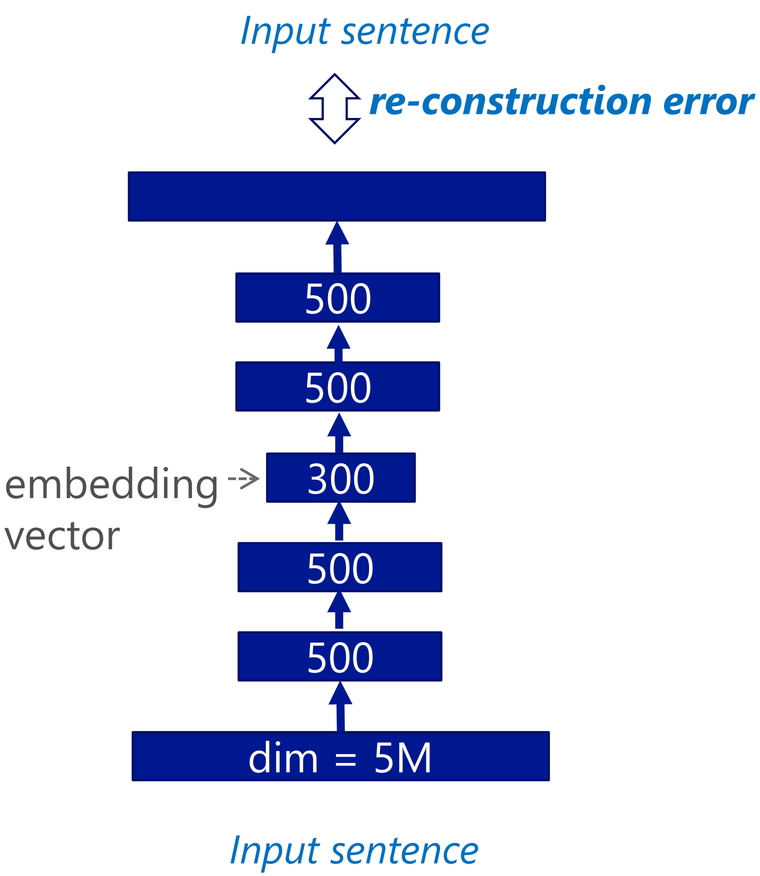
LSTM learns much faster than regular RNN

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



Reflection: from Auto-encoder to DSSM

Auto-encoder

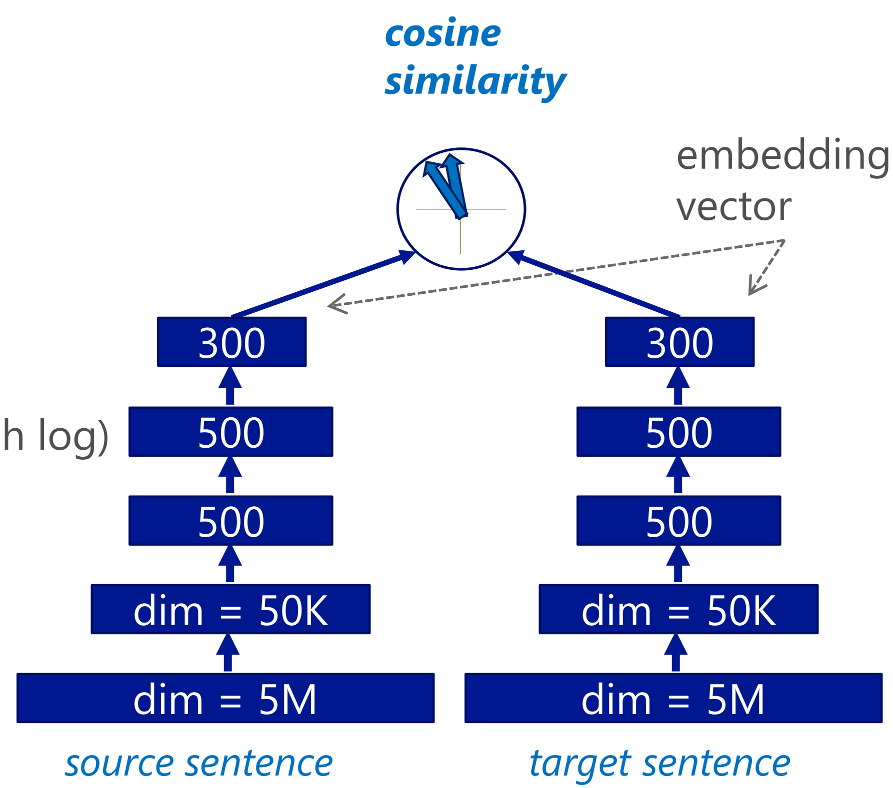


Training loss func.:
AE: reconstruction error
DSSM: distance between embedding vectors

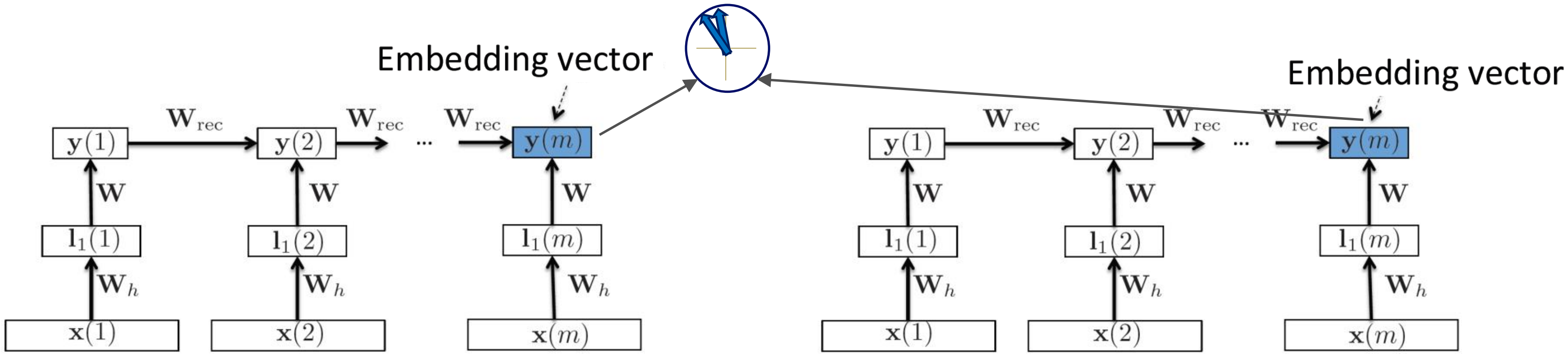
Training data:
AE: unsupervised (e.g., doc<->doc)
DSSM: weakly supervised (e.g., query<->doc search log)

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

DSSM



More comparison: DSSM vs. Seq2Seq

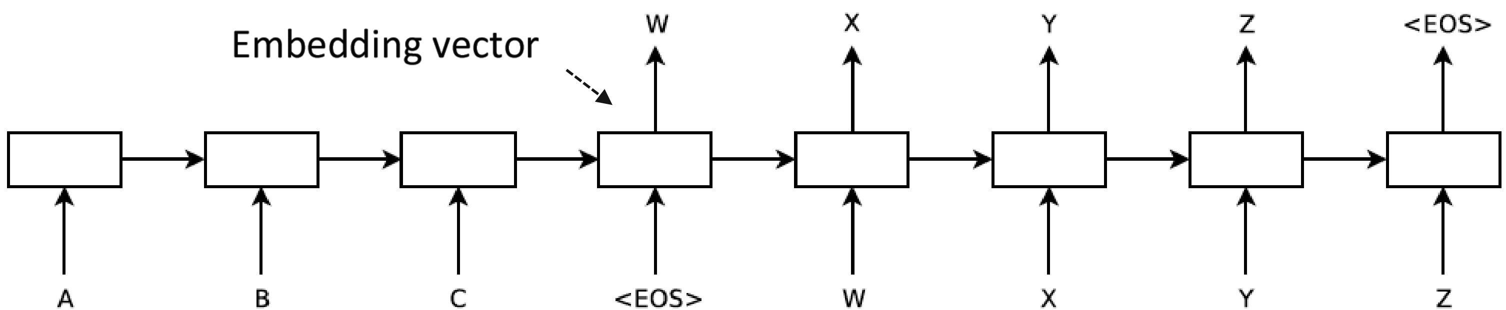


Source sentence

Target sentence

DSSM optimizes *sentence-level* semantic similarity

vs.



Source sentence

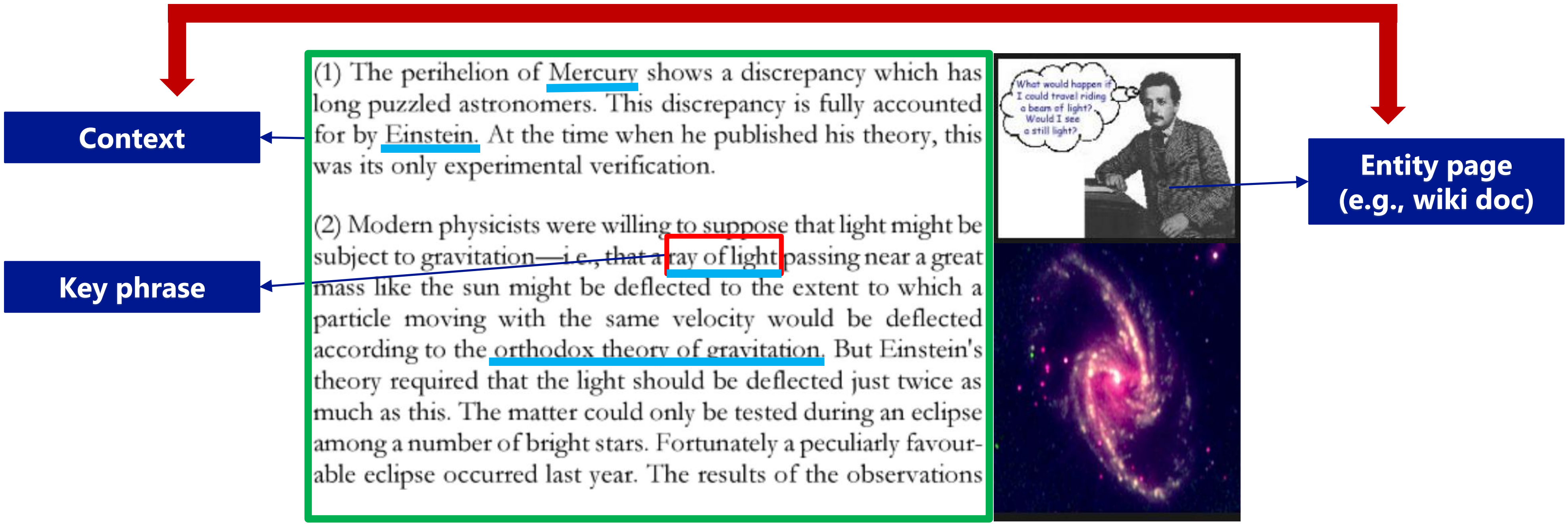
Target sentence

Seq2Seq optimizes *word-level* cross-entropy

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

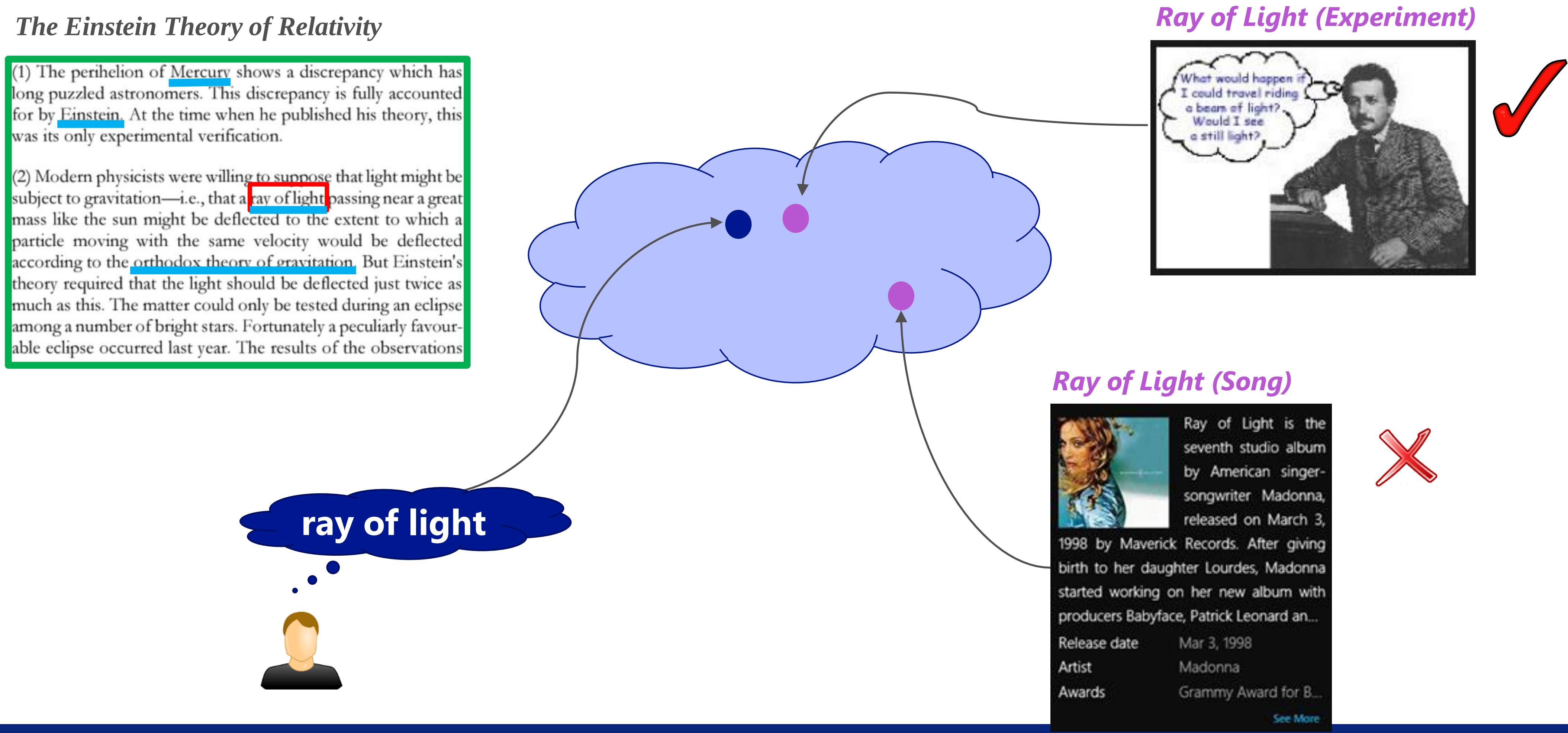
Contextual Entity Ranking

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



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

Learning DSSM for contextual entity ranking



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.
…

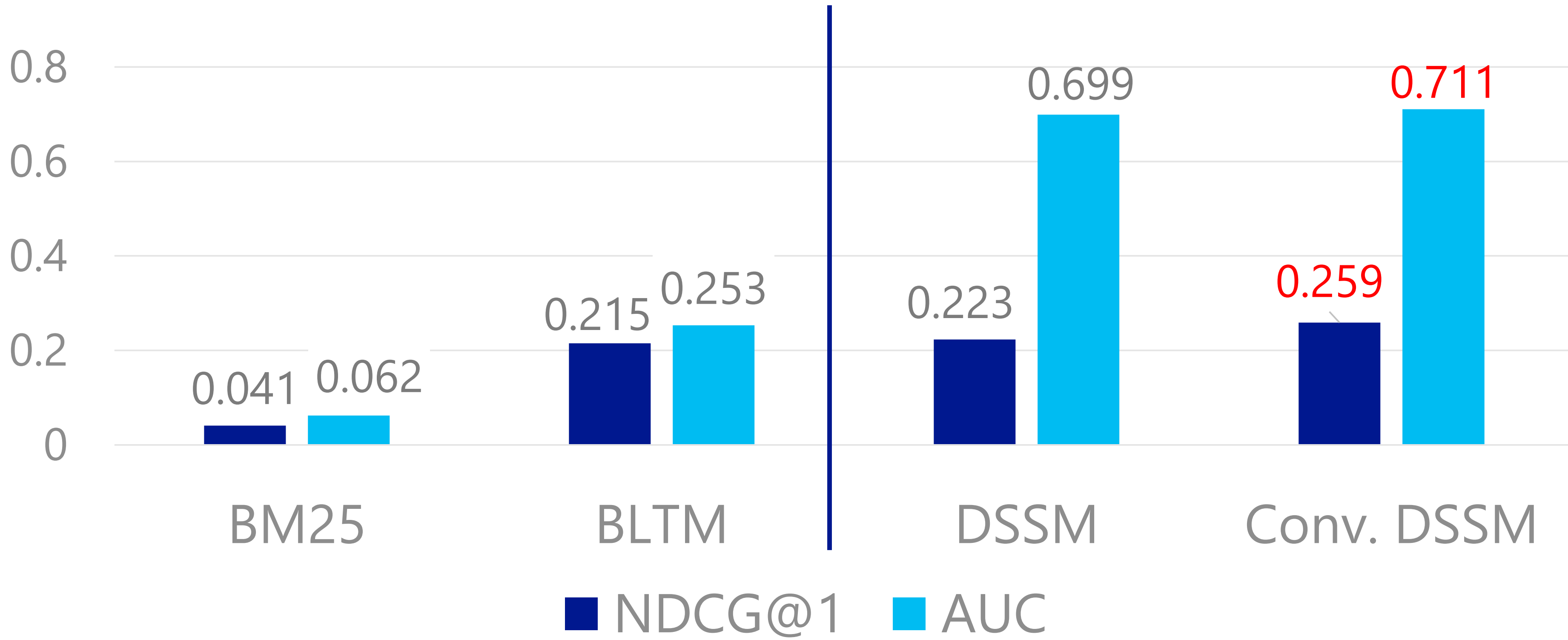
http://en.wikipedia.org/wiki/Bush_(band)

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

Contextual Entity Search: Experimental 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: DSSM



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

Some related work

Deep CNN for text input
Mainly classification tasks in the paper

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

Sequence to sequence learning

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

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.]

Interim summary

Learn Sent2Vec by the DSSM (Open Source: http://aka.ms/sent2vec/)

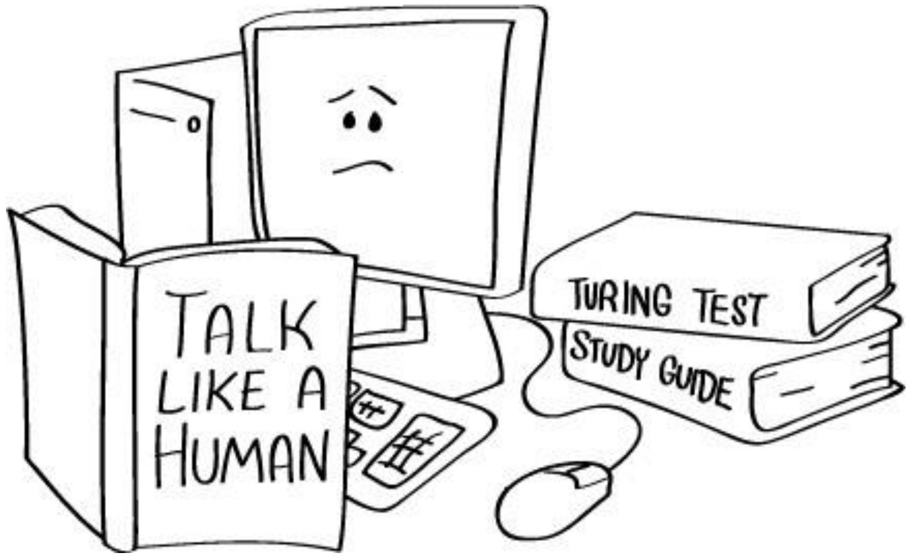
- The DSSM projects the whole-sentence to a continuous space
- The DSSM is built on the character level
- The DSSM directly optimizes semantic similarity objective functions

Part IV

Natural Language Understanding

Natural Language Understanding

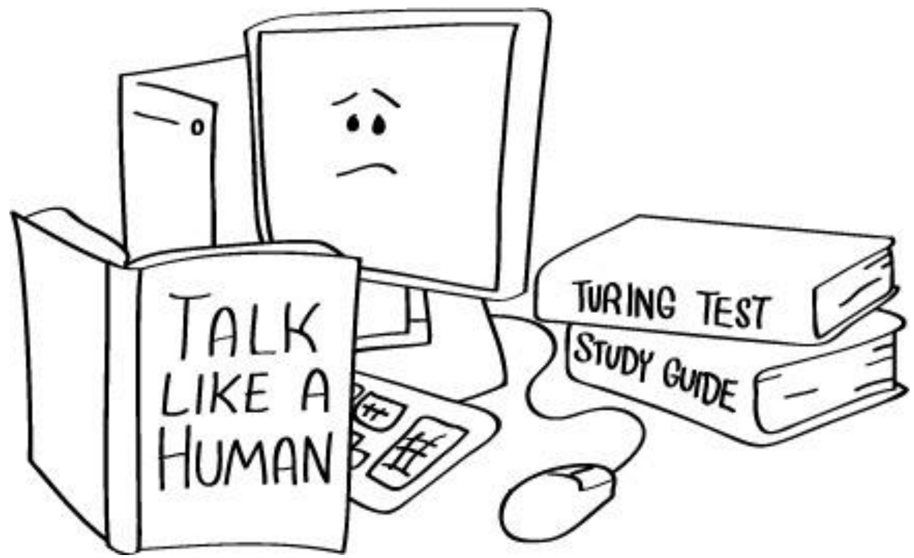
- Build an intelligent system that can interact with human using natural language
- Research challenge
  - Meaning representation of text
  - Support useful inferential tasks



http://csunplugged.org/turing-test

Natural Language Understanding

- Continuous Word Representations
  - Language is compositional
  - Word is the basic semantic unit
- Knowledge Base Embedding
- KB-based Question Answering & Machine Comprehension



http://csunplugged.org/turing-test

Continuous Word Representations

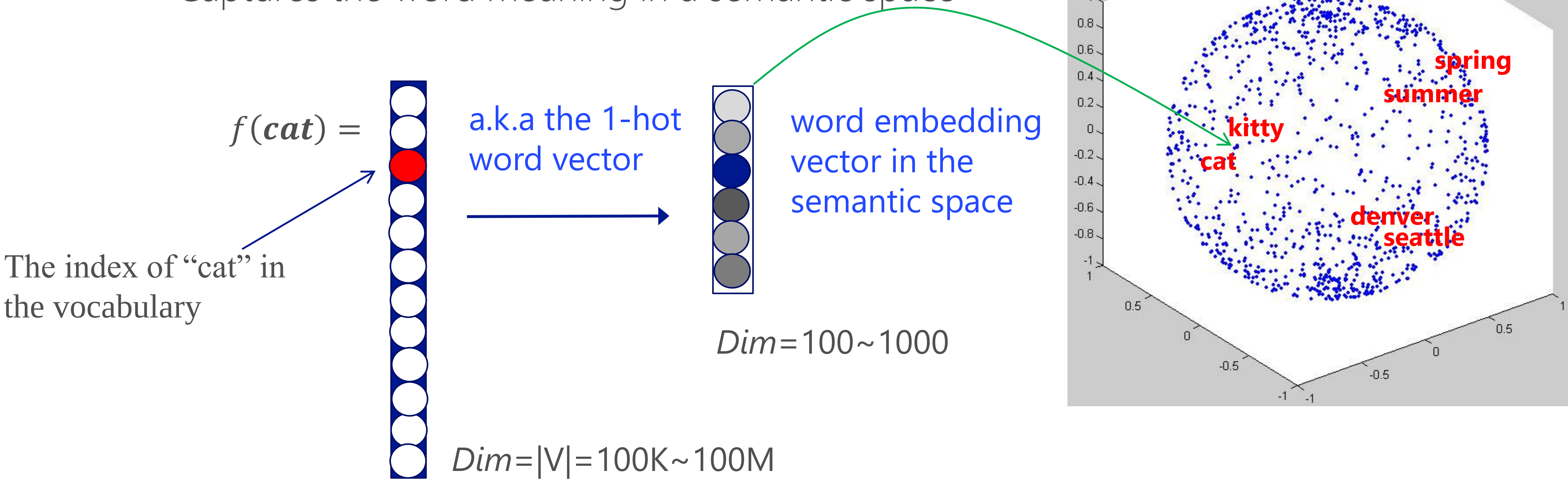
- A lot of popular methods for creating word vectors!
  - Vector Space Model [Salton & McGill 83]
  - Latent Semantic Analysis [Deerwester+ 90]
  - Brown Clustering [Brown+ 92]
  - Latent Dirichlet Allocation [Blei+ 01]
  - Deep Neural Networks [Collobert & Weston 08]
  - Word2Vec [Mikolov+ 13]
  - GloVe [Pennington+ 14]
- Encode term co-occurrence information
- Measure semantic similarity well

Semantic Embedding

Project raw text into a continuous semantic space

e.g., word embedding

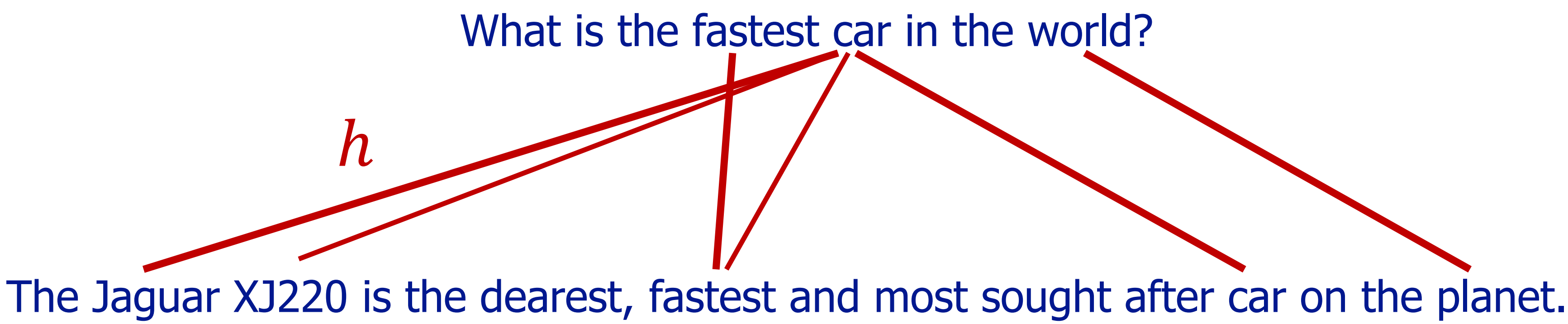
Captures the word meaning in a semantic space



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

Why is Word Embedding Useful?

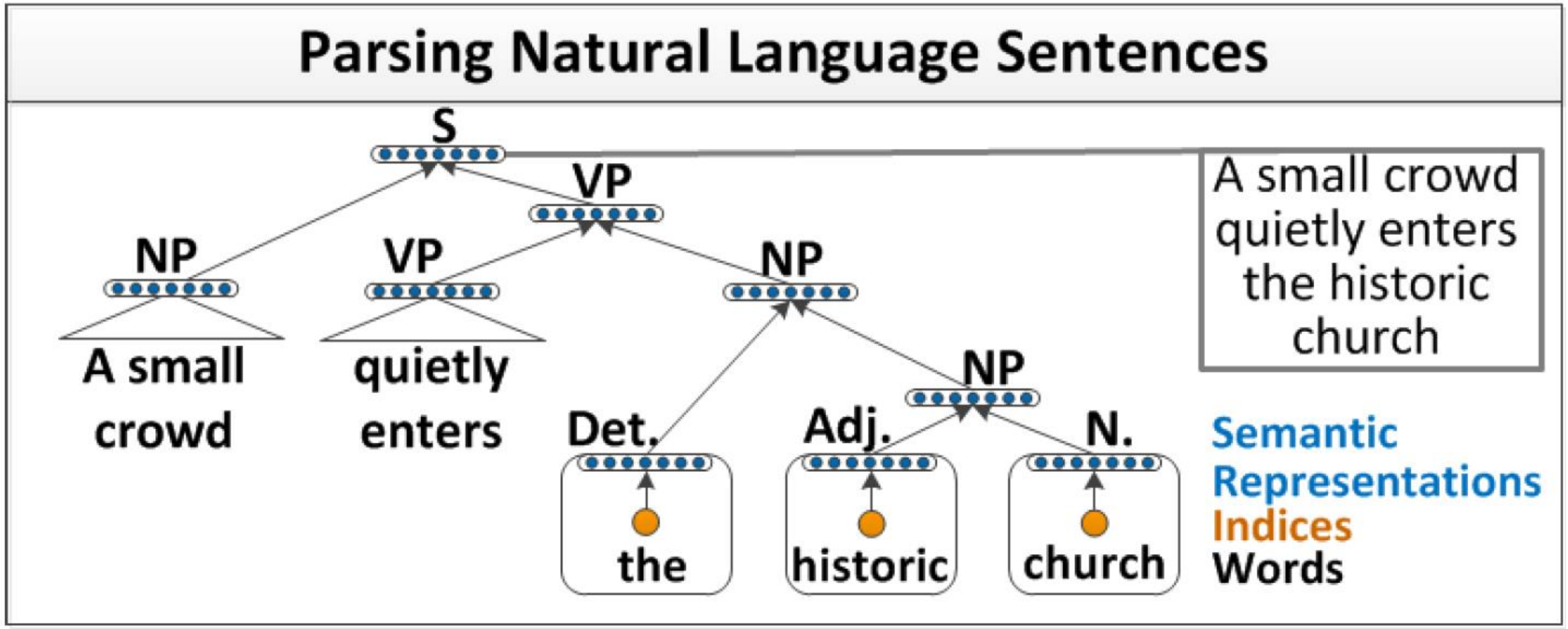
- Lexical semantics – semantic word similarity
  - Used as features in many NLP applications
  - e.g., Question/Sentence matching [Yih+ ACL-13; Jansen+ ACL-14]



- 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)

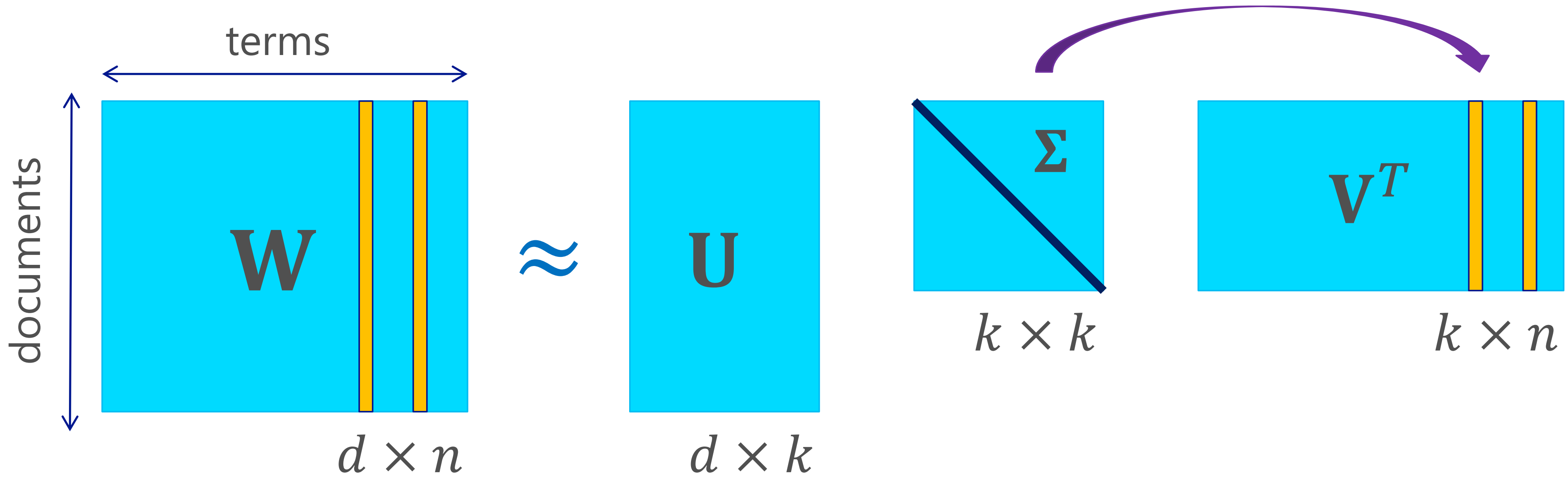
- "Pre-training" of a neural-network model
  - Take word vectors trained on a general corpus as input
  - e.g., Recursive NN for parsing [Socher+ ICML-11]



Roadmap – Continuous Word Representations

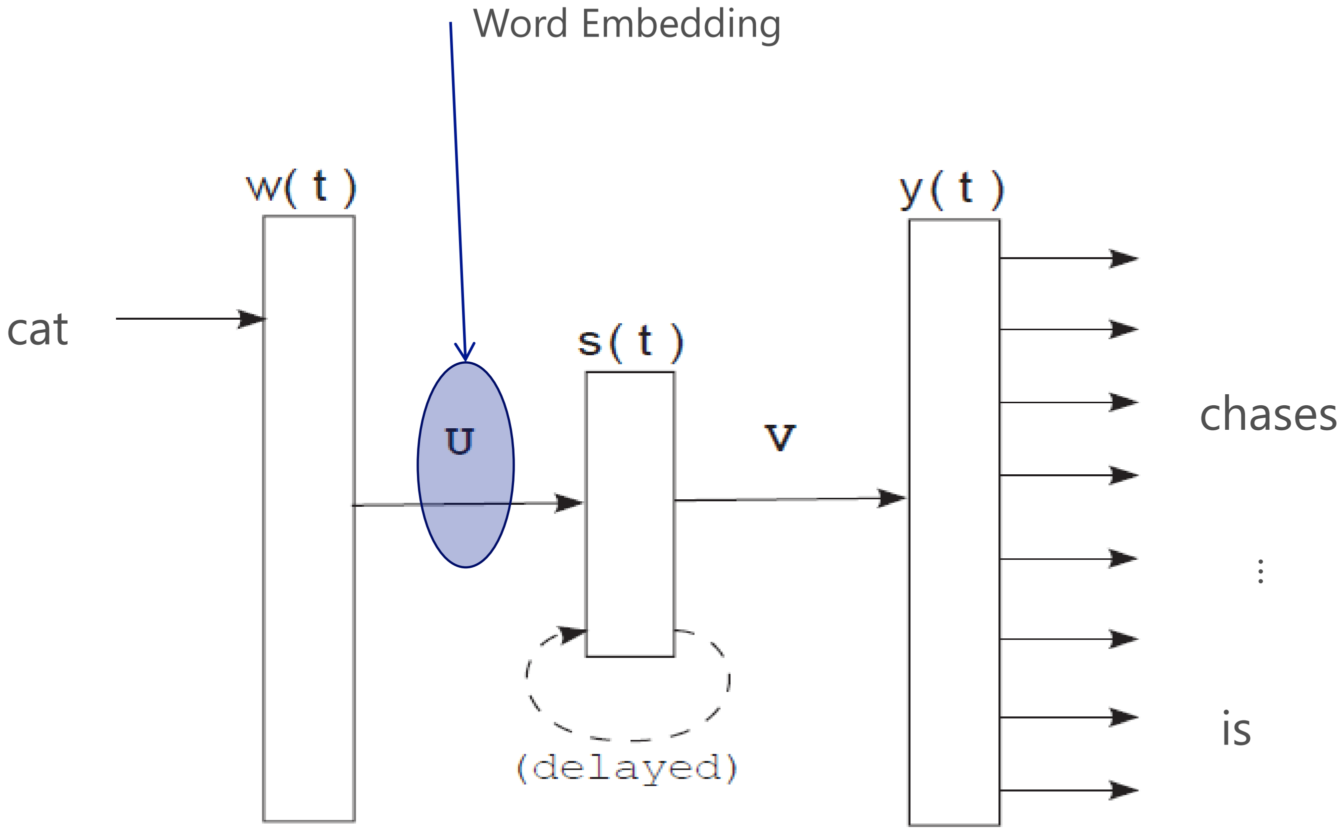
- Samples of word embedding models
  - Latent Semantic Analysis (LSA), Recurrent Neural Networks
  - SENNA, CBOW/Skip-gram, DSSM, GloVe
- Evaluation
  - Semantic word similarity
  - Relational similarity (word analogy)
- Related work
  - Model different word relations
  - Other word embedding models

Latent Semantic Analysis



- SVD generalizes the original data
- Uncovers relationships not explicit in the thesaurus
- Term vectors projected to 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

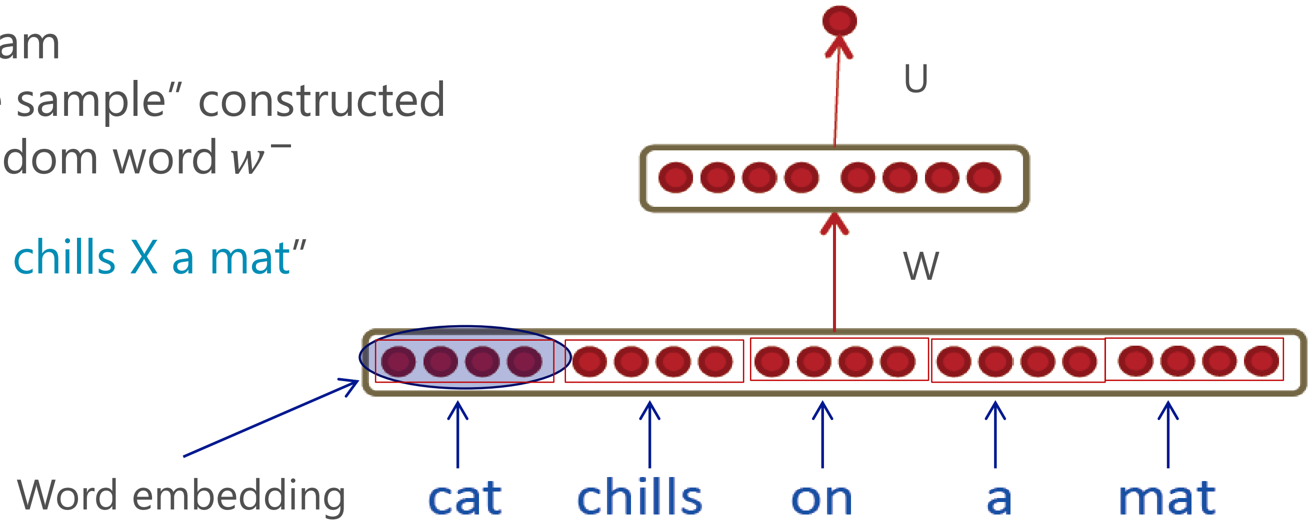
$$S^+ = Score(w_1, w_2, w_3, w_4, w_5)$$
$$S^- = Score(w_1, w_2, w^-, w_4, w_5)$$

And

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

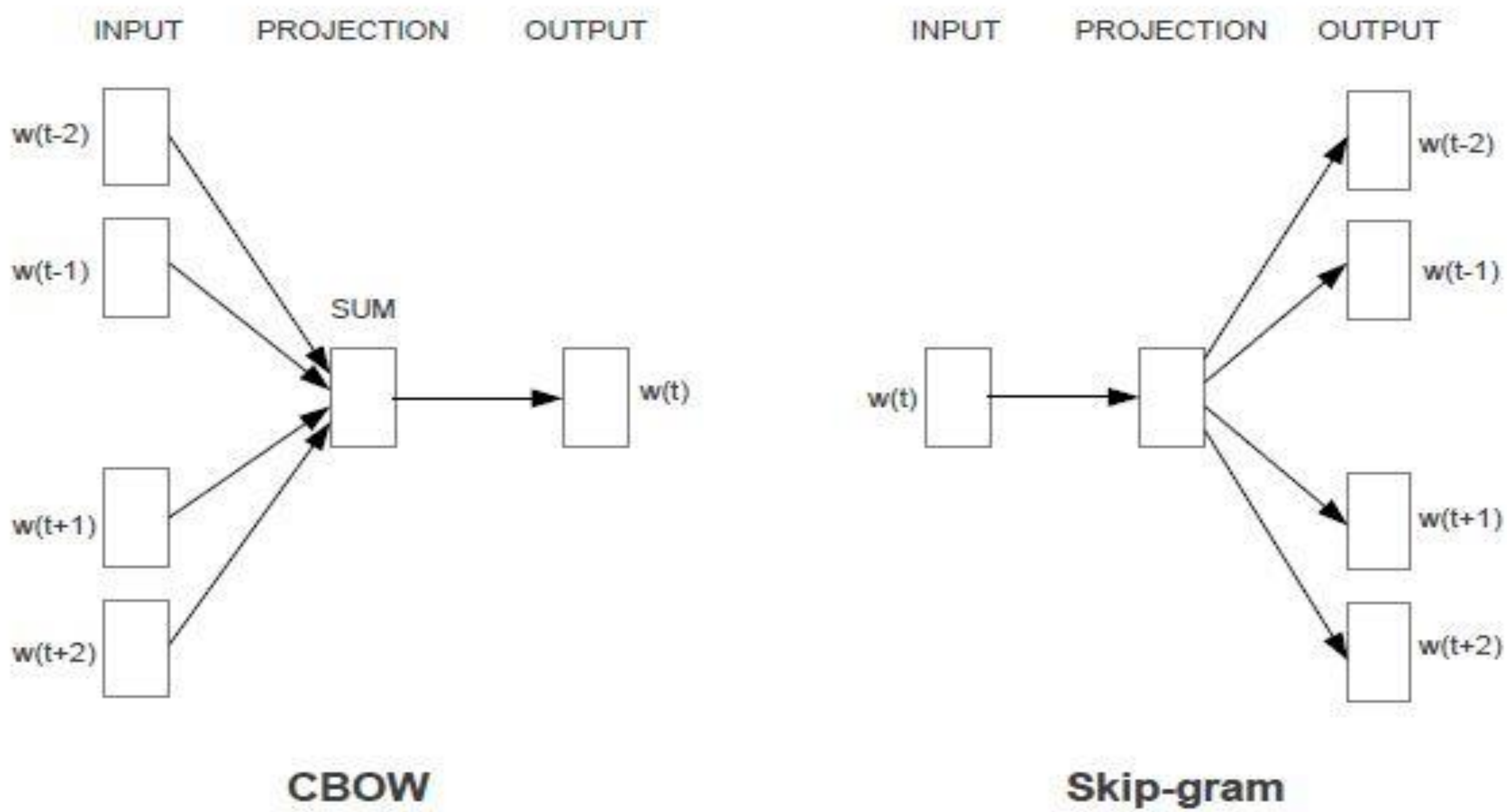
$< w_1, w_2, w^-, w_4, w_5 >$ is a "negative sample" constructed by replacing the word w_3 with a random word w^-

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

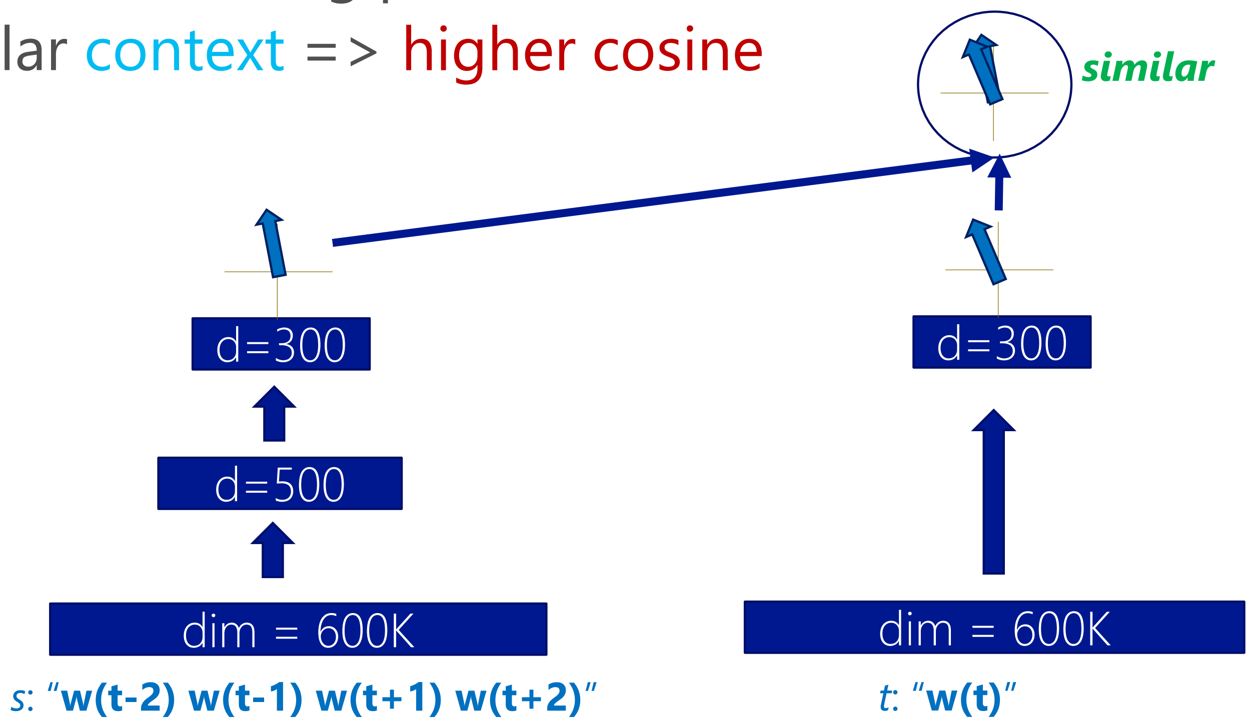


Continuous Bag-of-Words

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)

[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 $\stackrel{?}{=}$ man : woman

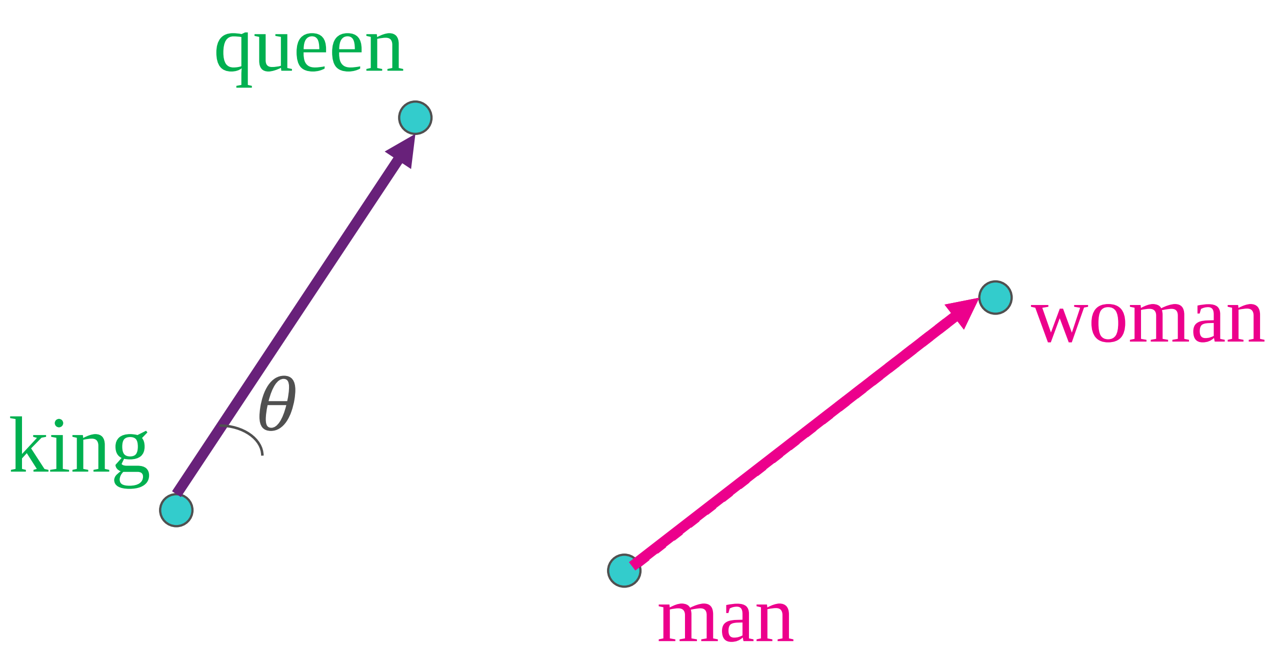
- Determine whether two pairs of words have the same relation (the "analogy" problem) [Bejar et al. '91]
  - (silverware : fork) vs. (clothing : shirt) [singular collective]
  - (coast : ocean) vs. (sidewalk : road) [contiguity]
  - (psychology : mind) vs. (astronomy : stars) [knowledge]

- 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

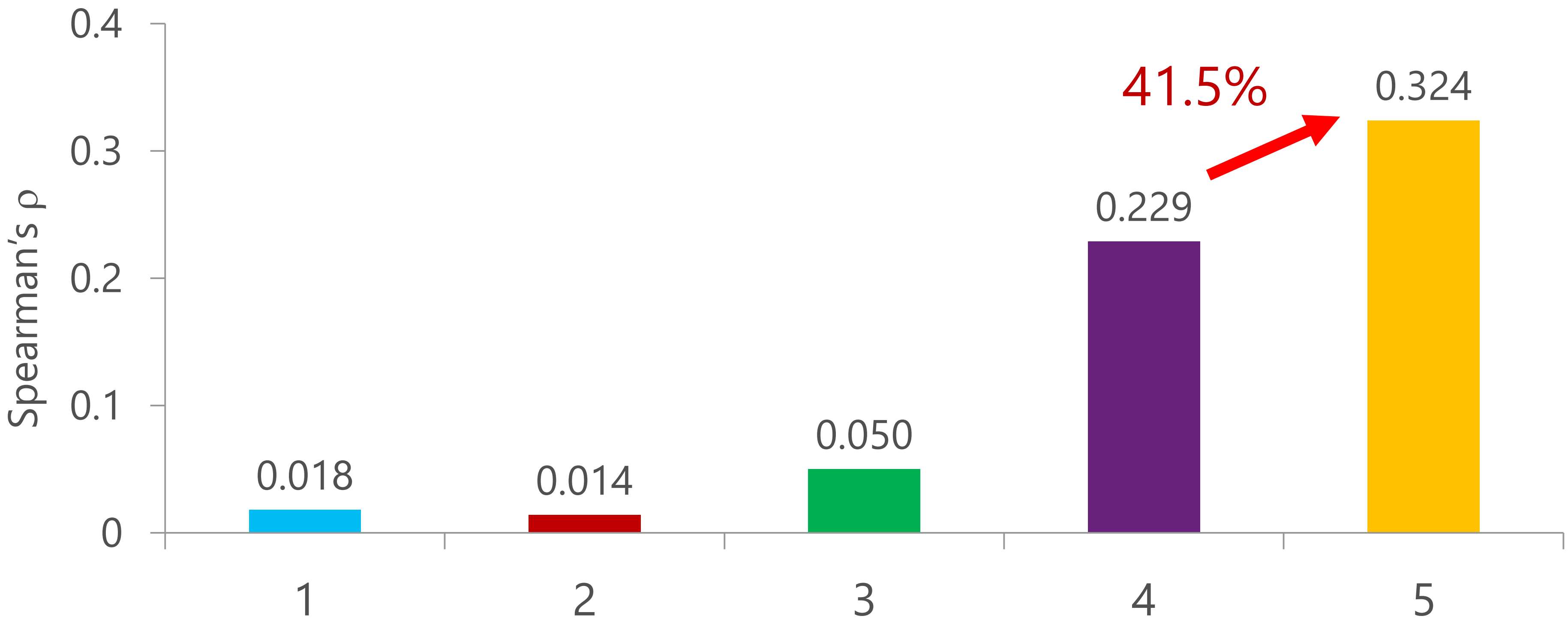
- Word embedding taken from recurrent neural network language model (RNN-LM) [Mikolov+ 2011]



- 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 from (Mnih et al., 2013); skip-gram (SG) and CBOW from (Mikolov et al., 2013a,b); GloVe from (Pennington+, 2014)

Discussion

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

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

Related Work – Model Different Word Relations

Tomorrow will be rainy.

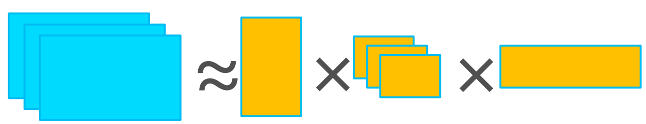
Tomorrow will be sunny.

$similar$(r , sunny)?

$antonym$(rainy, sunny)?

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

$f_{rel}(\bullet, \bullet)$

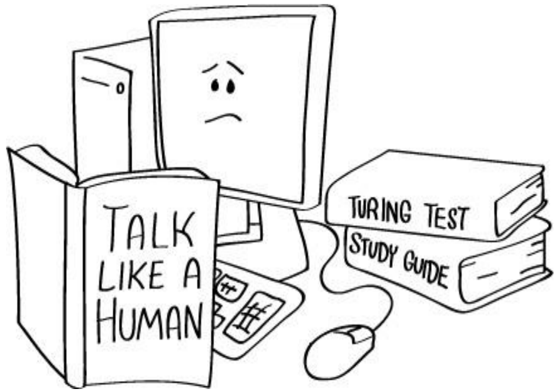


Related Work – Word Embedding Models

- Other word embedding models
  - [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]
- Theoretical justification and unification
  - Word Embeddings as Metric Recovery in Semantic Spaces [Hashimoto+ TACL-16]
- New Evaluation: RelEval@ACL-16 – Evaluating Vector Space Representations for NLP

Natural Language Understanding

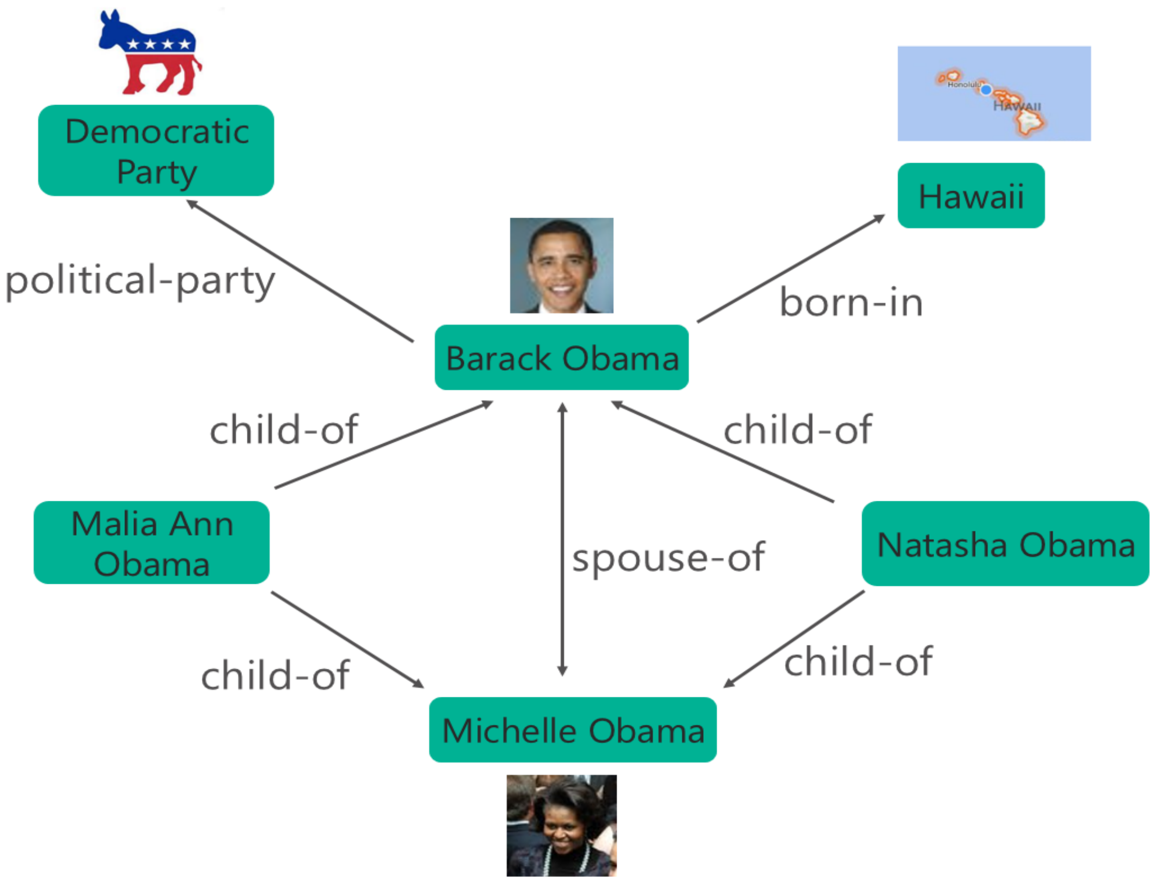
- Continuous Word Representations & Lexical Semantics
- **Knowledge Base Embedding**
  - Nickel et al., "A Review of Relational Machine Learning for Knowledge Graphs"
- KB-based Question Answering & Machine Comprehension

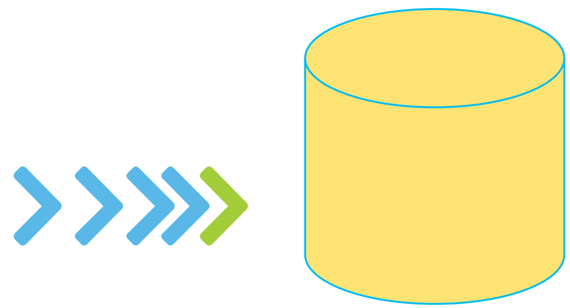


http://csunplugged.org/turing-test

Knowledge Base

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





Freebase
DBpedia
YAGO
NELL
OpenIE/ReVerb

Current KB Applications in NLP & IR

- Question Answering

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

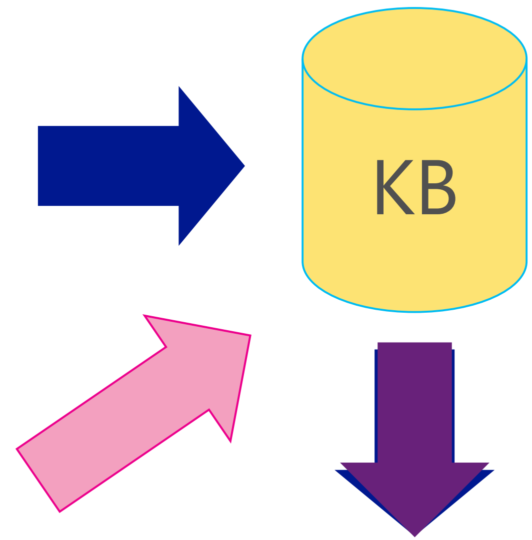
  $\lambda x.parent(Obama, x) \wedge gender(x, Female)$

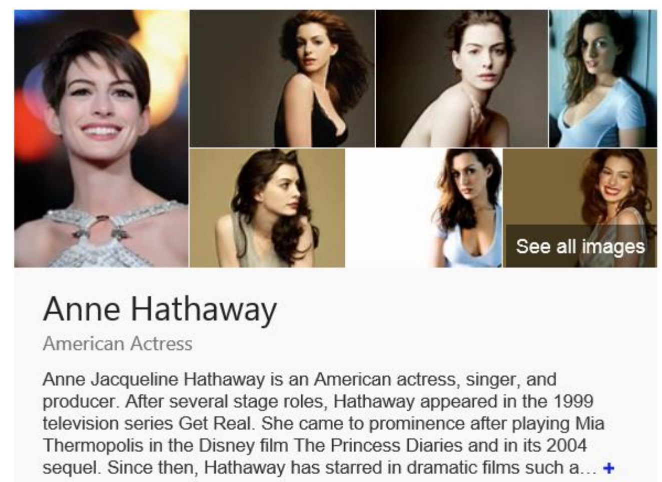
- Information Extraction
  - *"Hathaway was born in Brooklyn, New York."*

  $bornIn(Hathaway, Brooklyn)$

  $contains(New\ York, Brooklyn)$

- Web Search
  - Identify entities and relationships in queries





Reasoning with Knowledge Base

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

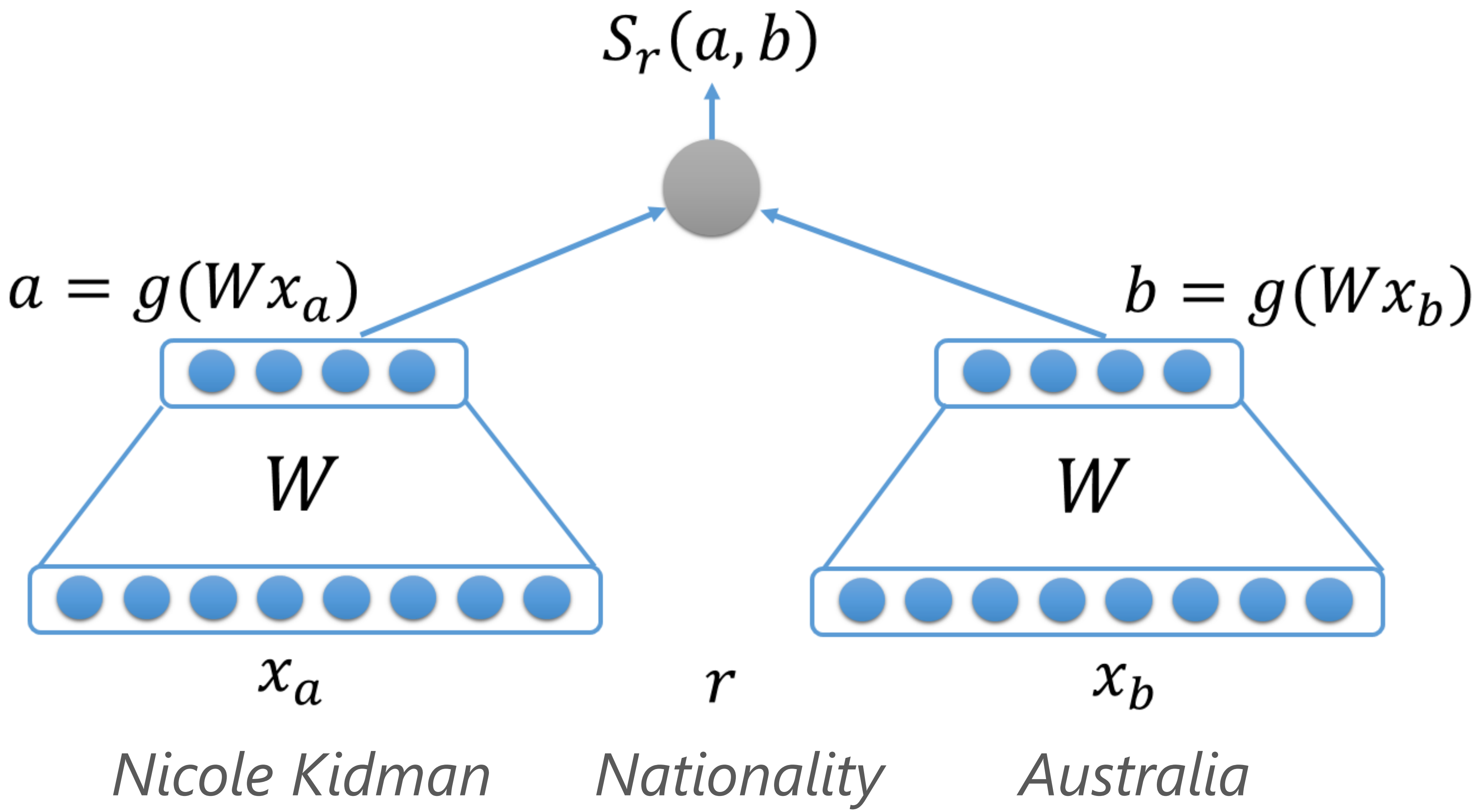
- Modeling multi-relational data
  - Statistical relational learning [Getoor & Taskar, 2007]
  - Path ranking methods (e.g., random walk) [e.g., Lao+ 2011]
  - Knowledge base embedding
    - Very efficient
    - Better prediction accuracy

Knowledge Base Embedding

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

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

Neural Knowledge Base Embedding



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

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

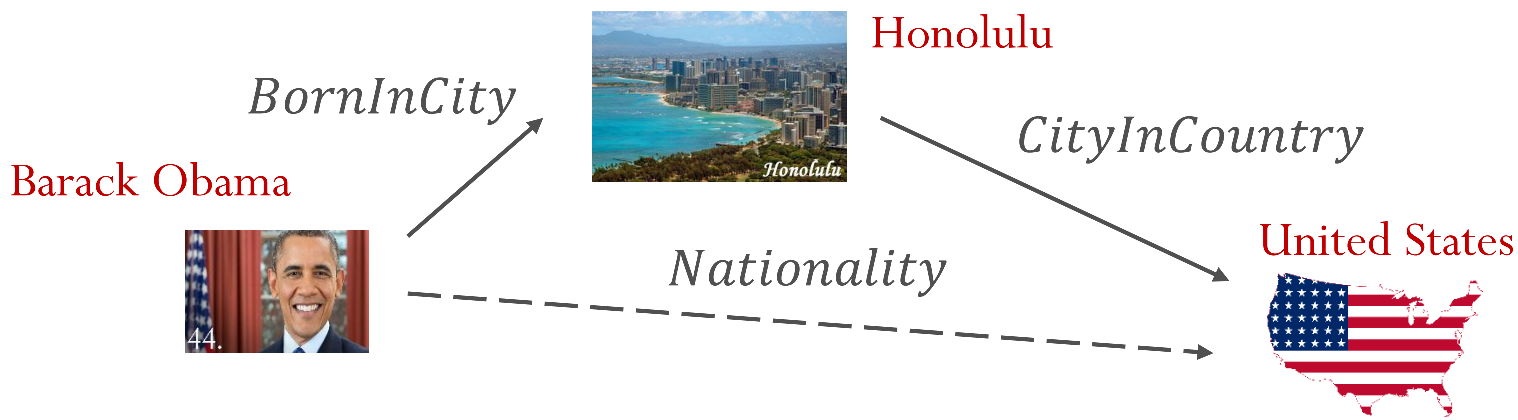
Empirical Comparisons of NN-based KB Embedding Methods [Yang+ ICLR-2015]

- 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 [Yang+ ICLR-2015]

- Can relation embedding capture relation composition?

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



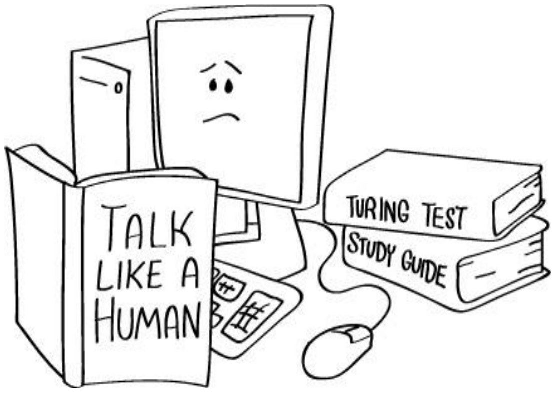
- Embedding-based Horn-clause rule extraction
  - For each relation r, find a chain of relations $r_1 \cdots r_n$, such that:

$$dist(M_r, M_1 \circ M_2 \circ \cdots \circ M_n) < \theta$$

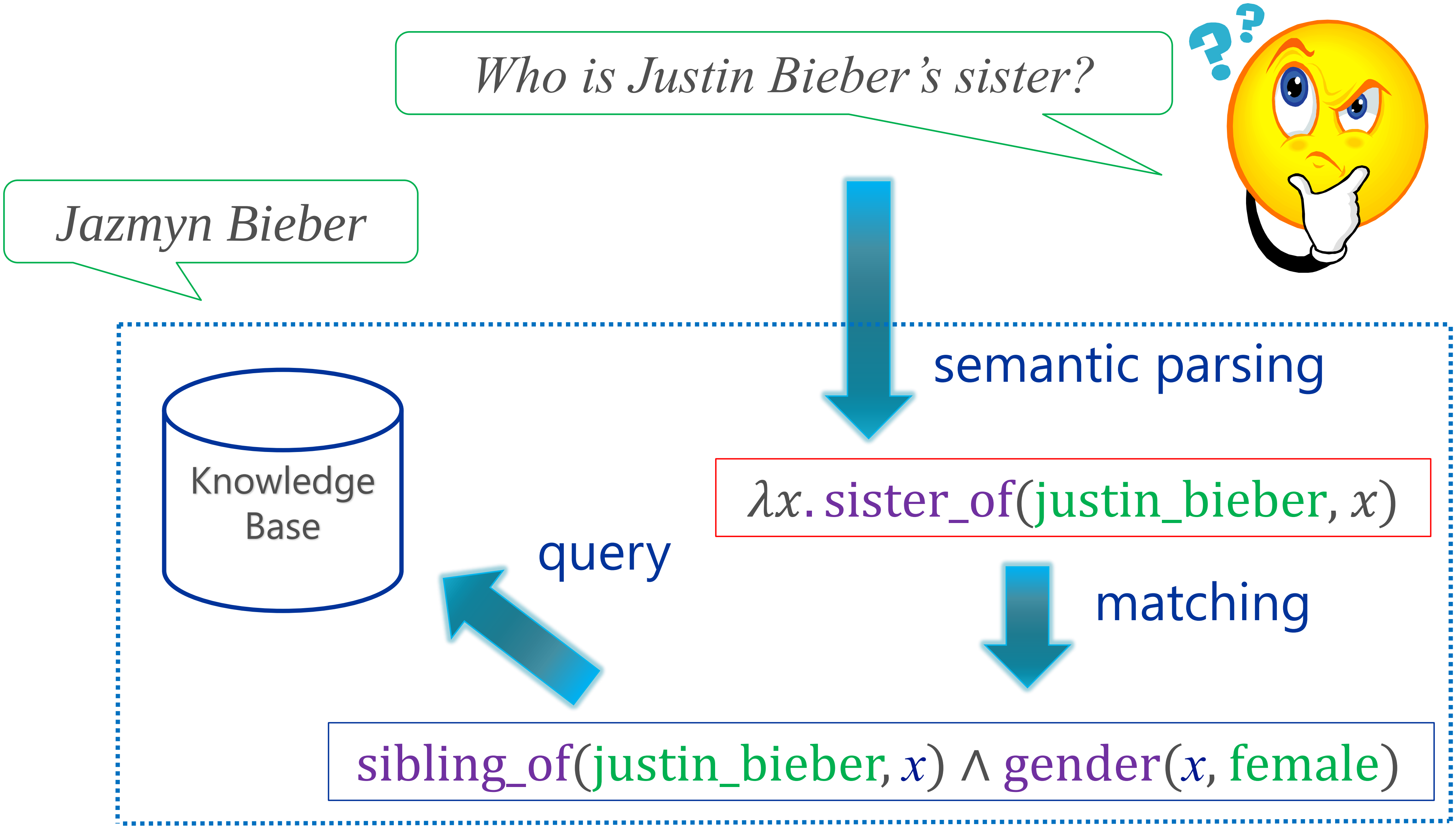
  - $r_1(e_1, e_2) \land r_2(e_2, e_3) \cdots \land r_n(e_n, e_{n+1}) \rightarrow r(e_1, e_{n+1})$

Natural Language Understanding

- Continuous Word Representations & Lexical Semantics
- Knowledge Base Embedding
- KB-based Question Answering & Machine Comprehension



http://csunplugged.org/turing-test


Who is Justin Bieber's sister?
Jazmyn Bieber
semantic parsing
Knowledge Base
$\lambda x.\,\text{sister\_of}(\text{justin\_bieber}, x)$
query
matching
$\text{sibling\_of}(\text{justin\_bieber}, x) \wedge \text{gender}(x, \text{female})$

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

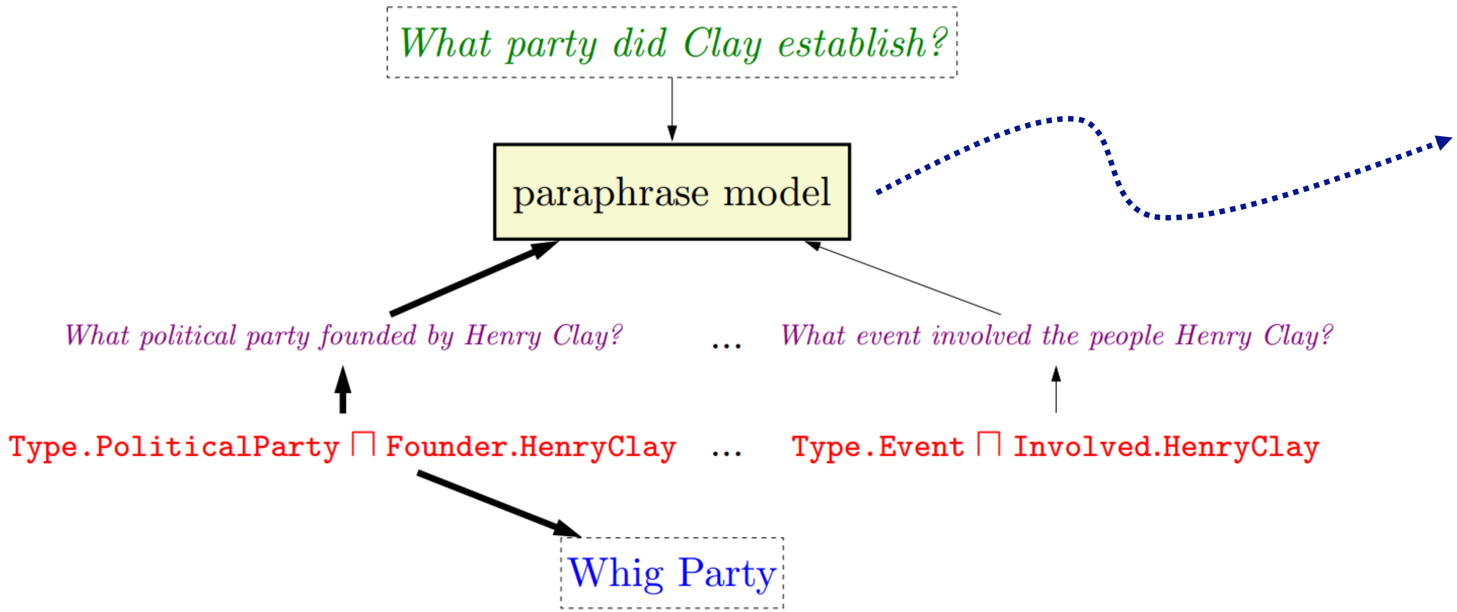
Matching Question and Relation

- Similar text can be mapped 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]



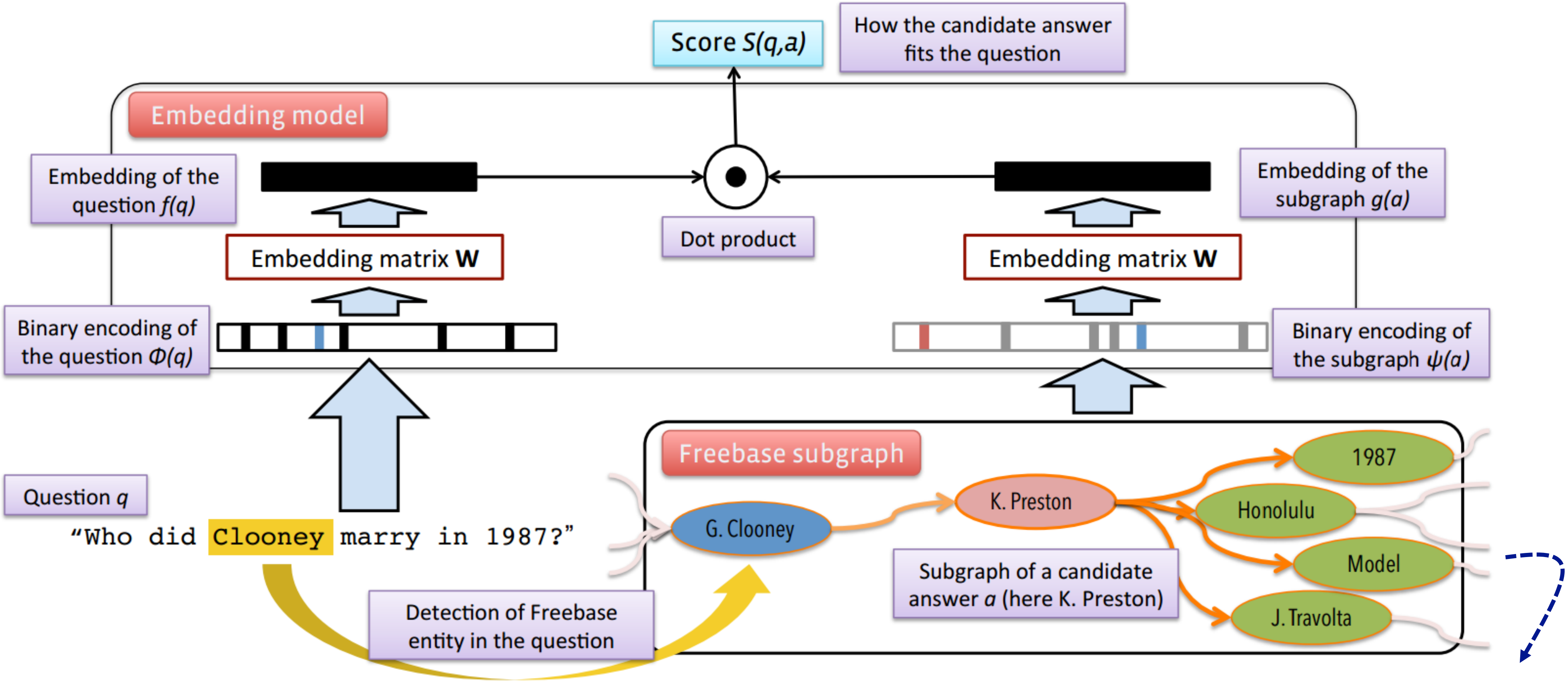
Learned using question paraphrases in Paralex [Fader+ ACL-13]

- 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)

Semantic Parsing

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

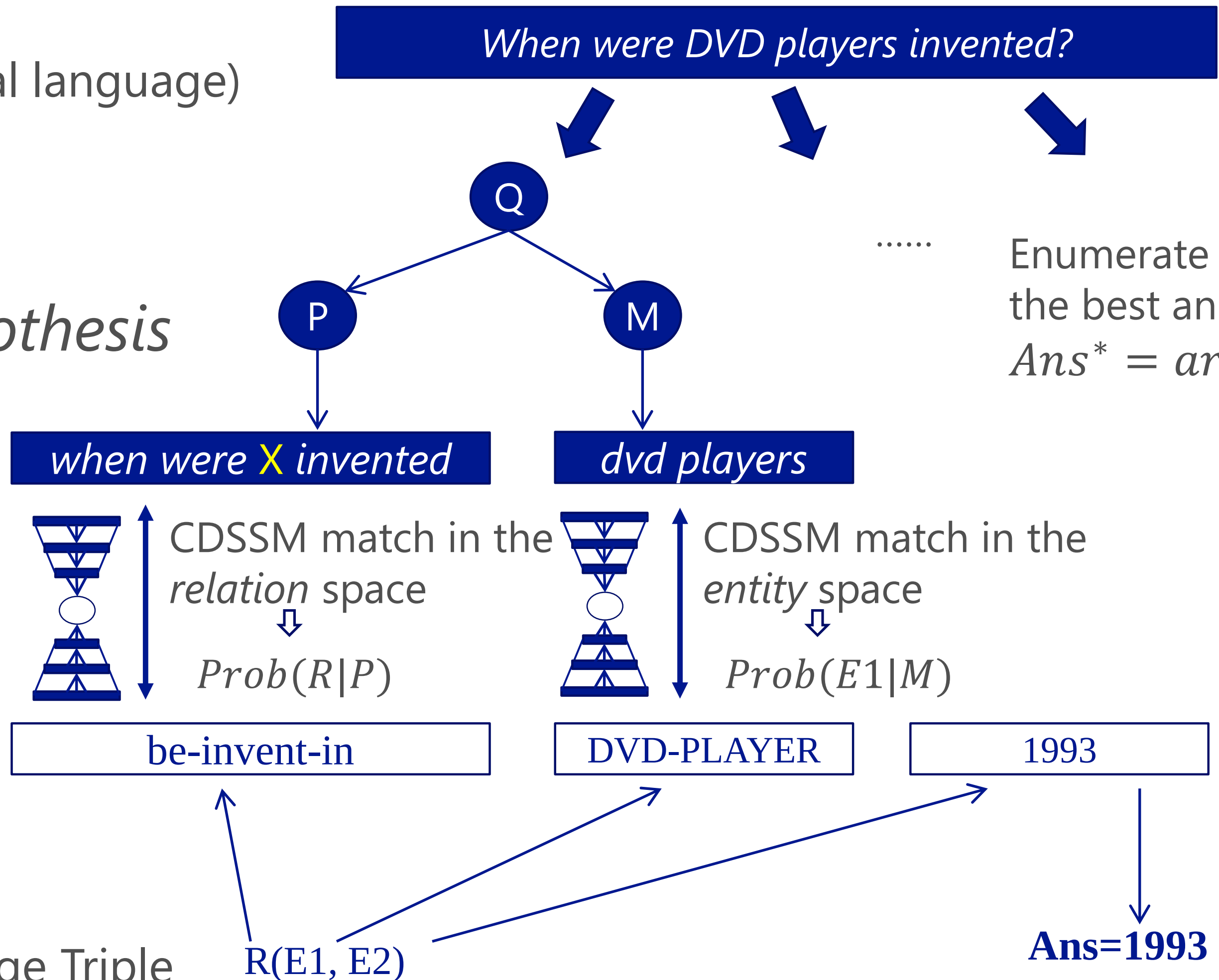
$$Q \rightarrow P \wedge M$$
$$P \rightarrow when\ were\ \mathrm{X}\ invented$$
$$M \rightarrow DVD\ players$$
$$when\ were\ X\ invented \rightarrow \mathrm{be{-}invent{-}in}_2$$
$$DVD\ players \rightarrow \mathrm{dvd{-}player}$$

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

A joint decoding process

Question
(in natural language)

A hypothesis



Knowledge Triple
(in logical form)

Ans=1993

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

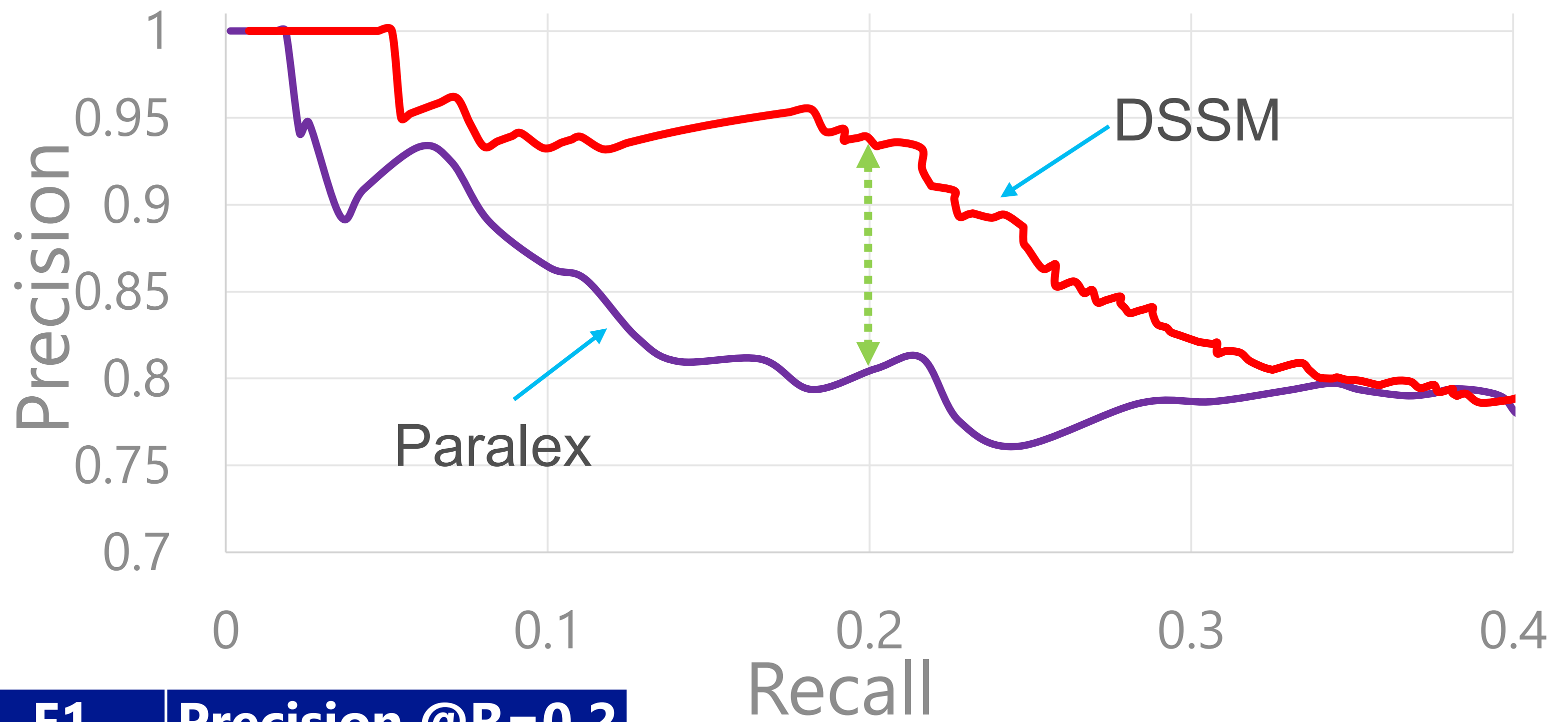
Joint decoding for:
- entity linking
- semantic parsing
- inferring answer

Enumerate all hypotheses to search for the best answer:
$Ans^* = argmax_{Ans} P(Ans|KB,Q)$

$$P(Ans|KB,Q) = \sum_{SP} P(Ans,SP|KB,Q)$$
$$\approx \max_{SP,Triple} P(Ans|SP,KB,Q)P(SP|Q)$$
$$\approx \max_{SP,Triple} Prob(R|P) \times Prob(E1|M)$$

[Yih, He, Meek, ACL 2014]

Experiments: Results

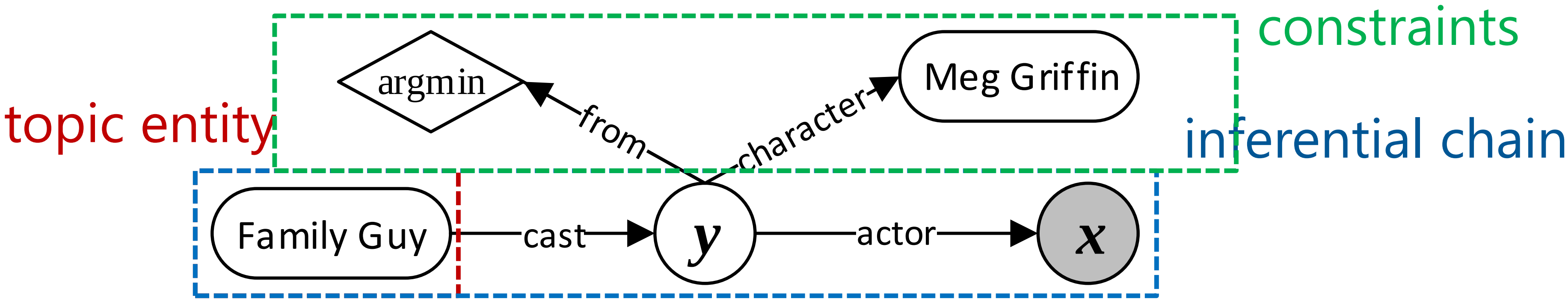


On paralex dataset (the UW benchmark)	F1	Precision @R=0.2
ParaLex (baseline)	54%	80.6%
SPQA based on DSSM	61%	93.4%

From [Yih, He, Meek, ACL 2014]

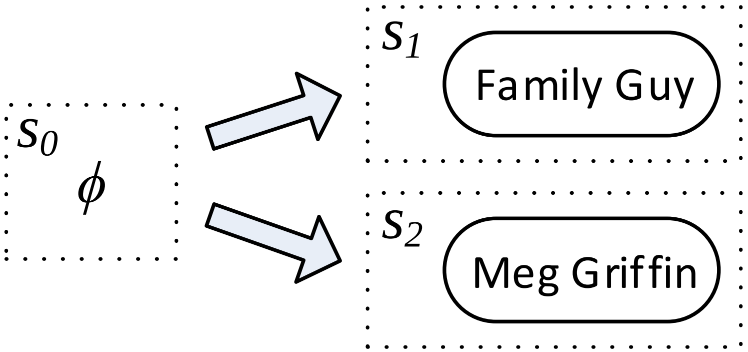
Staged Query Graph Generation [Yih, Chang, He, Gao, ACL'15]

- 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?
- $\lambda x. \exists y. \text{cast}(\text{FamilyGuy}, y) \wedge \text{actor}(y, x) \wedge \text{character}(y, \text{MegGriffin})$



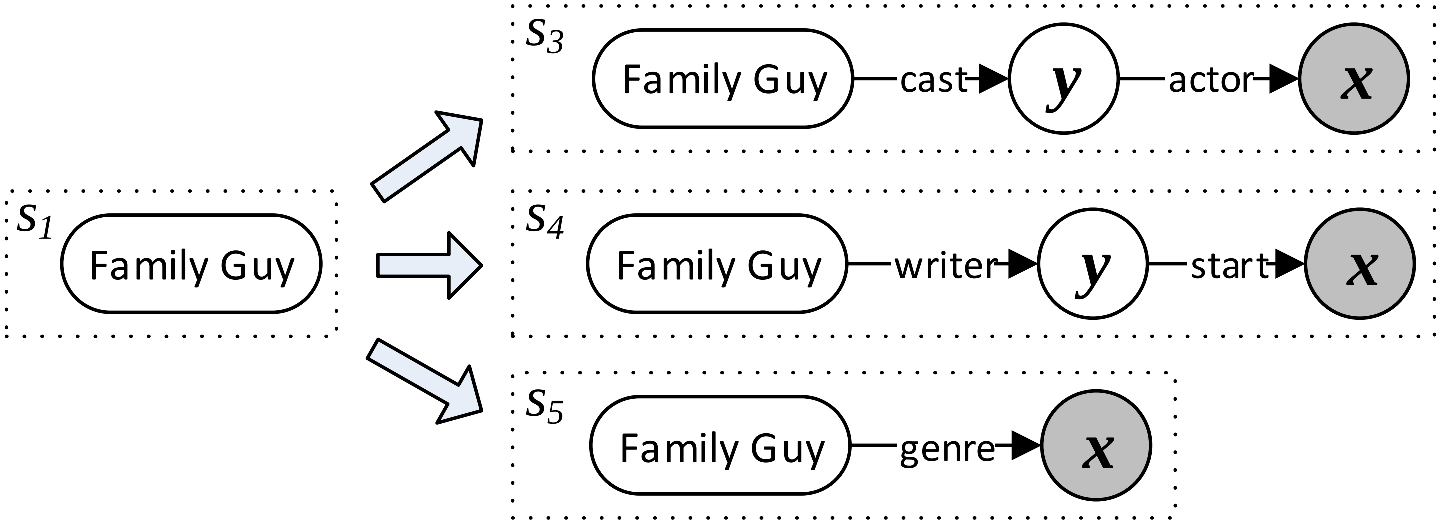
Graph Generation Stages

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



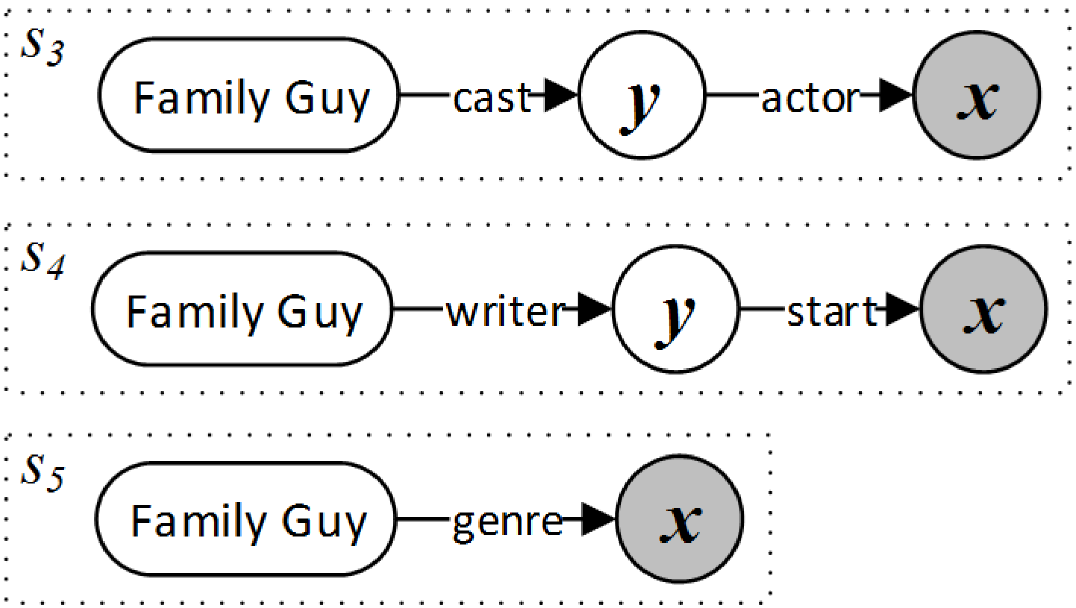
1. Topic Entity Linking [Yang&Chang ACL-15]

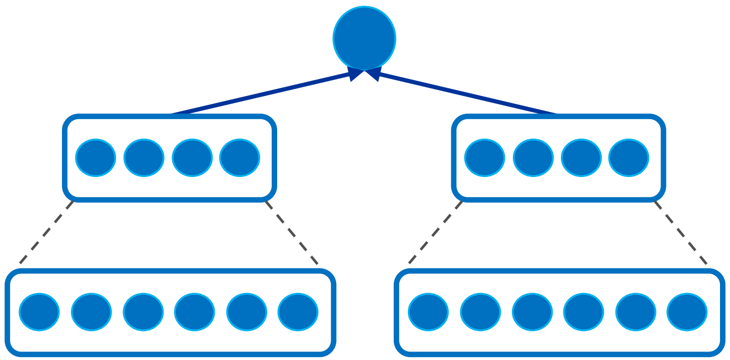
2. Identify the core inferential chain



Identify Inferential Chain using DSSM

- Who first voiced Meg on Family Guy?



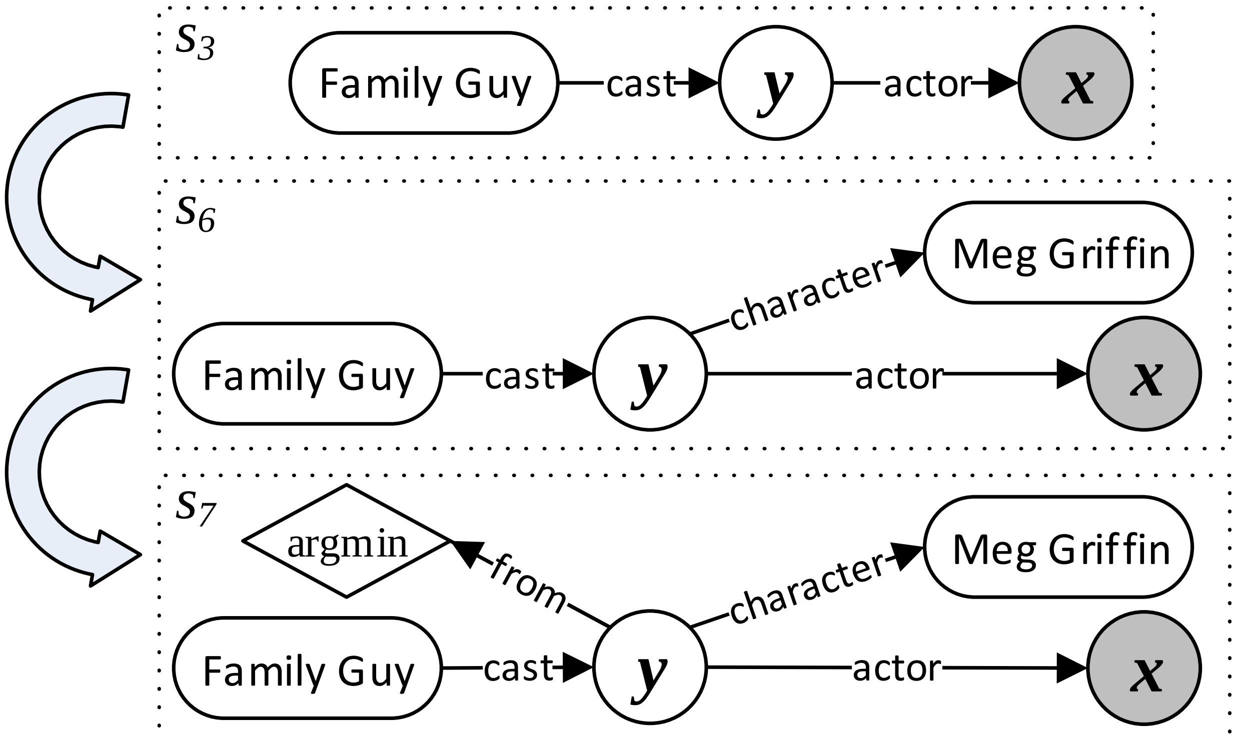


- Semantic match (**"Who first voiced Meg on ⟨*e*⟩"**, "cast-actor")
- Single pattern/relation matching model: 49.6% F_1 (vs. 52.5% F_1 Full)

Graph Generation Stages (cont'd)

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

3. Augment constraints



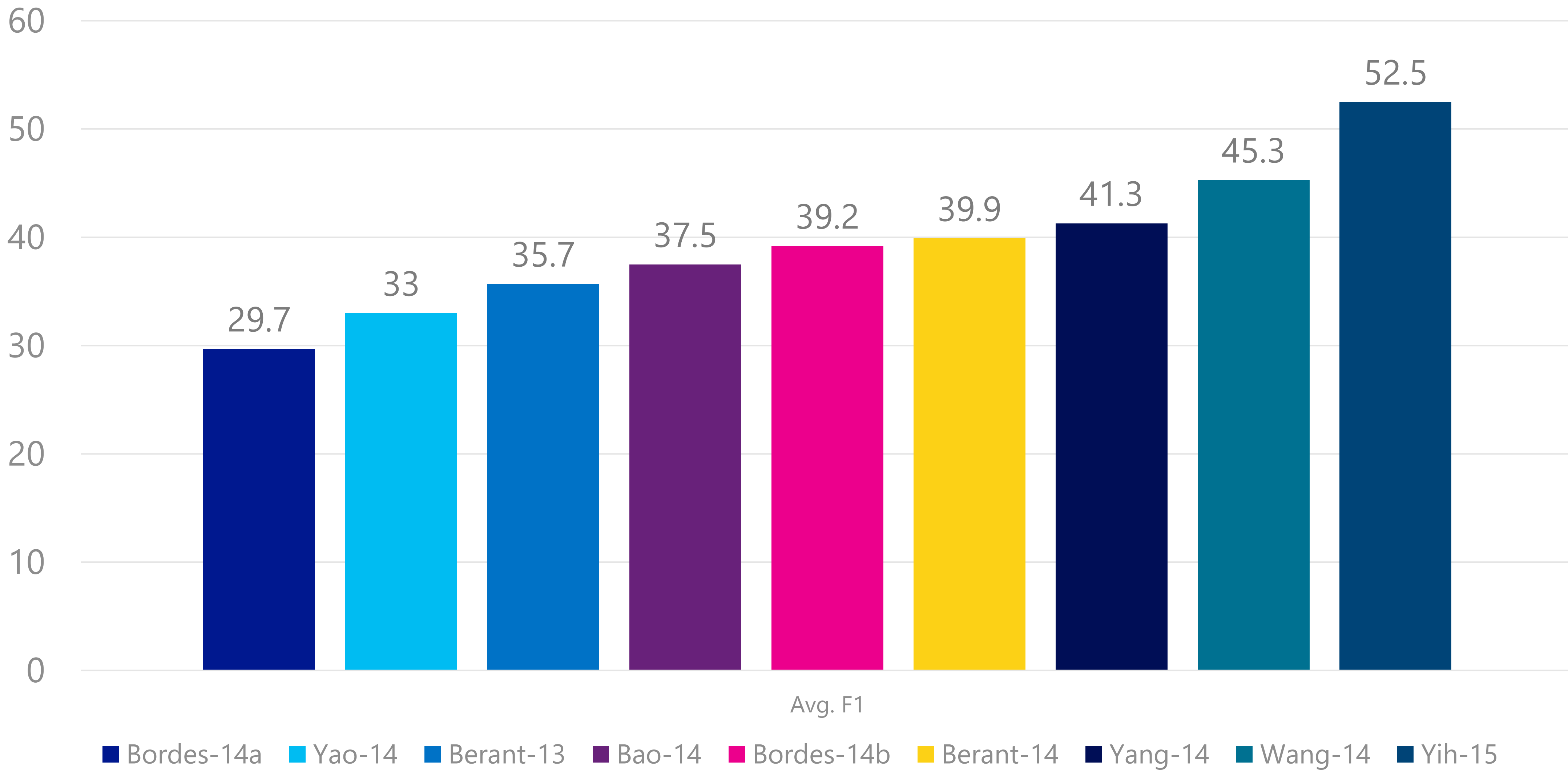
WebQuestions Dataset [Berant+ EMNLP-2013]

- *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

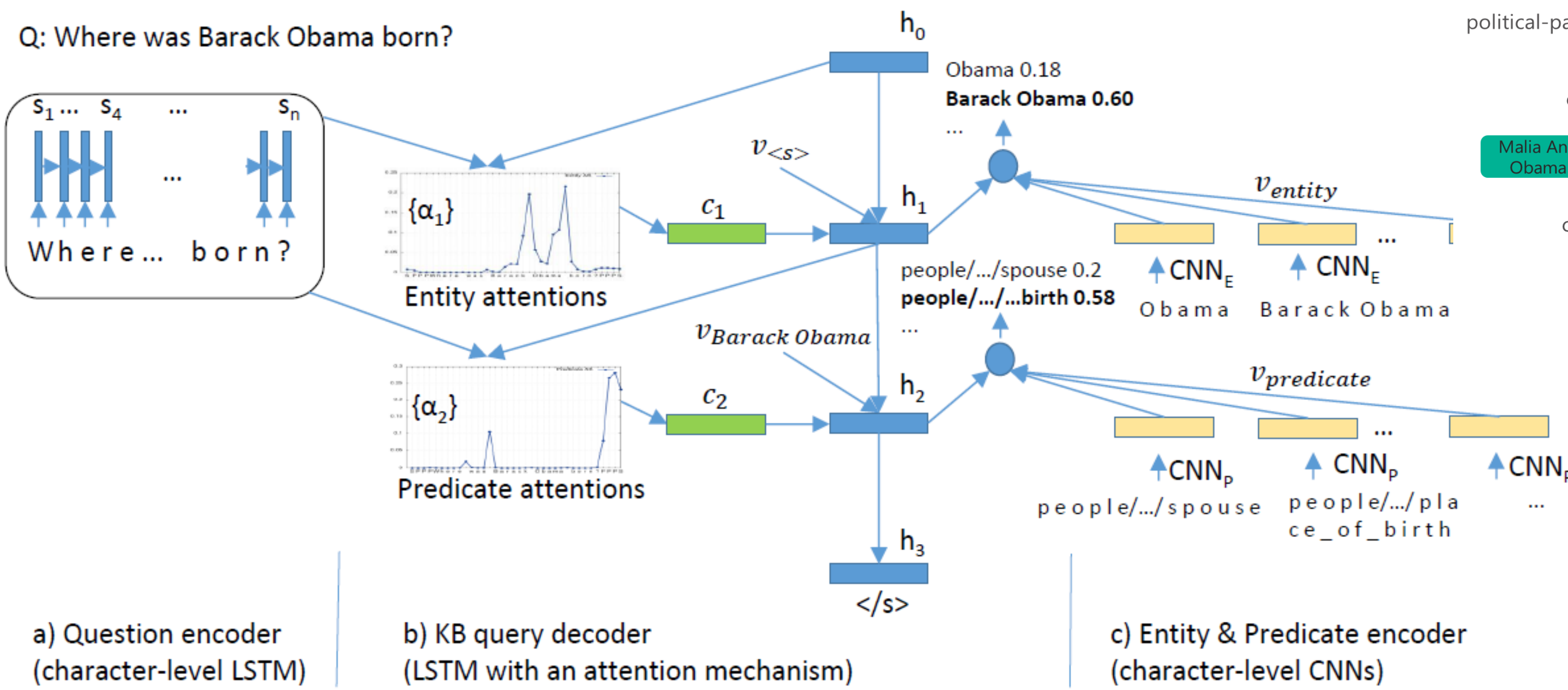
[Examples from Berant]

- 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



More recent progress: Character-level Question Answering with Attention



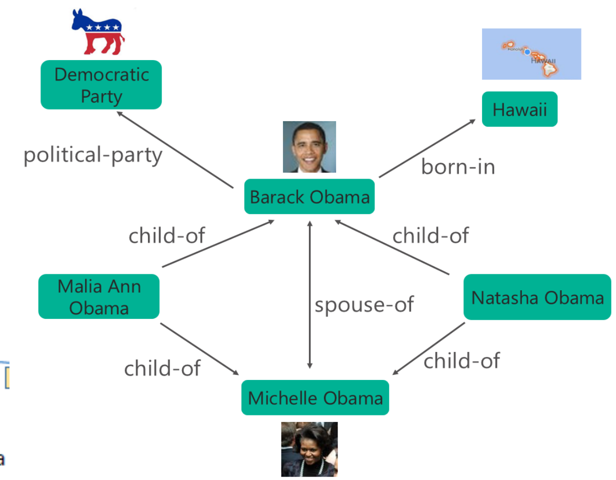
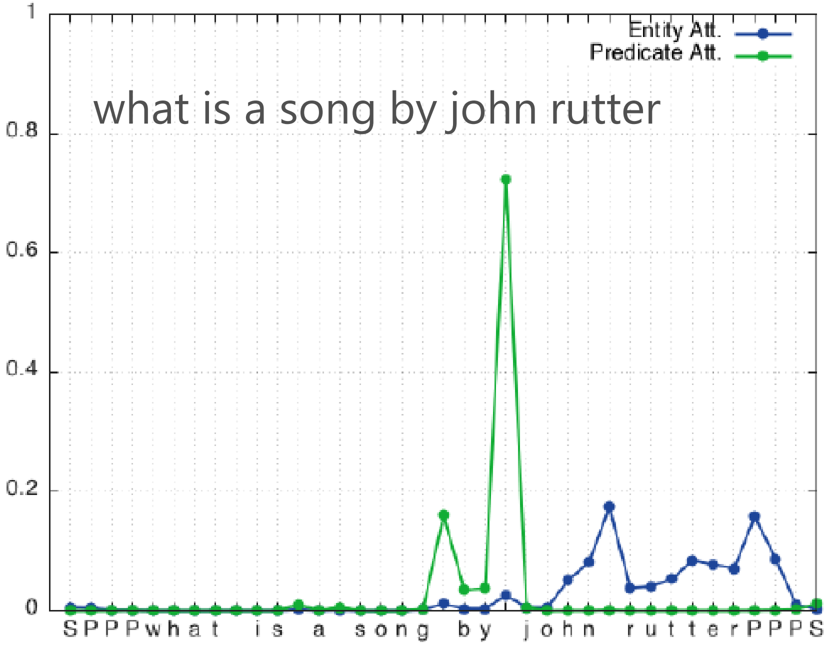
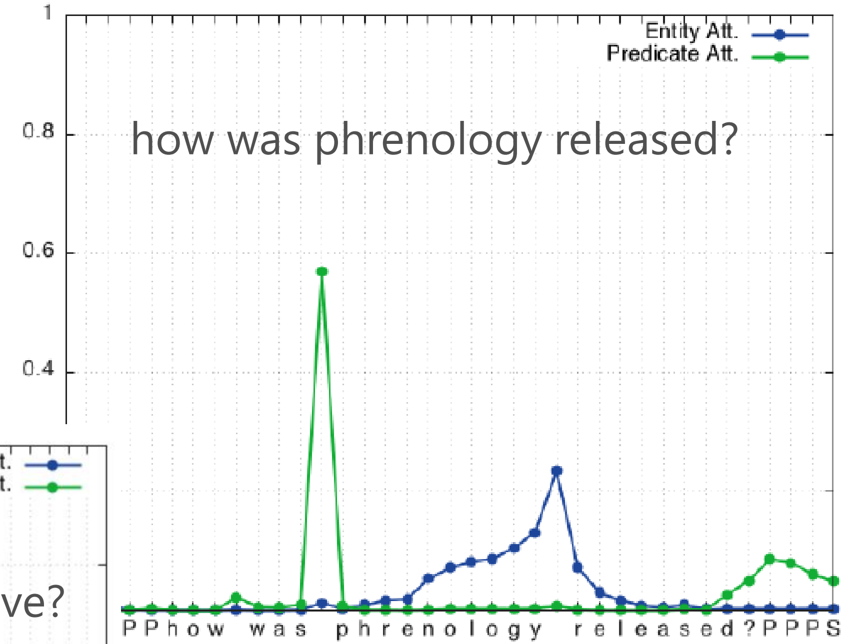


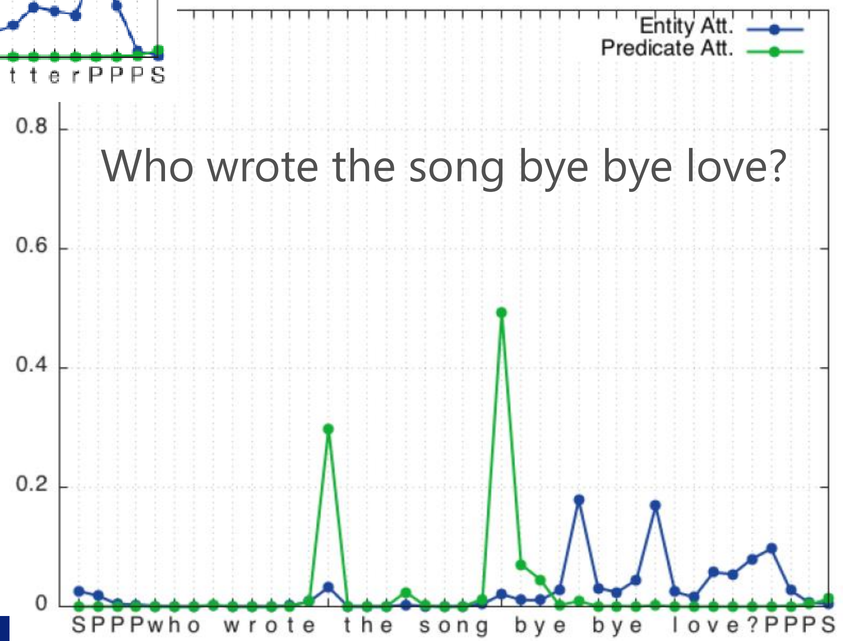
Figure 1: Our encoder-decoder architecture that generates a query against a structured knowledge base. We encode our question via a long short-term memory (LSTM) network and an attention mechanism to produce our context vector. During decoding, at each time step, we feed the current context vector and an embedding of the English alias of the previously generated knowledge base entry into an attention-based decoding LSTM to generate the new candidate entity or predicate.

[David Golub and Xiaodong He, 2016]

Capture semantic meaning at the character level







Compared to Memory Neural Net

Results on SimpleQuestions dataset									
KB	Train sources WQ	SIQ	PRP	Autogen. Questions	Embed Type	Model	Ensemble	SQ Accuracy	# Train Examples
FB2M	no	yes	no	no	Char	Ours	1 model	**70.9**	76K
FB2M	no	yes	no	no	Word	Ours	1 model	53.9	76K
FB2M	yes	yes	yes	yes	Word	MemNN	1 model	62.7	26M
FB5M	no	yes	no	no	Char	Ours	1 model	**70.3**	76K
FB5M	no	yes	no	no	Word	Ours	1 model	53.1	76K
FB5M	yes	yes	yes	yes	Word	MemNN	5 models	63.9	27M
FB5M	yes	yes	yes	yes	Word	MemNN	Subgraph	62.9	27M
FB5M	yes	yes	yes	yes	Word	MemNN	1 model	62.2	27M

Emerging Neural Net models in language understanding:

- RNN (recurrent network) / LSTM (long short-term memory network) / Seq2Seq / Encoder-Decoder
- Attention Model / Memory Neural Network / Neural Turing Machine
- Reinforcement learning
- Scenarios: QA, dialog, lang. generation, parsing, Text-based Games, Real-time Thread recommendation …

DeepMind Q&A Dataset [Hermann et al., NIPS-15]

- High-level dataset creation process
  - Pick a large corpus (e.g., news articles, stories)
  - Develop an (almost) automatic way to generate (fill-in-the-blank) questions

- 93k CNN & 220k Daily Mail articles
- Bullet points (summary / paraphrases) → Cloze questions
  - Replacing one entity with a placeholder
  - ~4 questions per document
  - ~1M document / query / answer triples

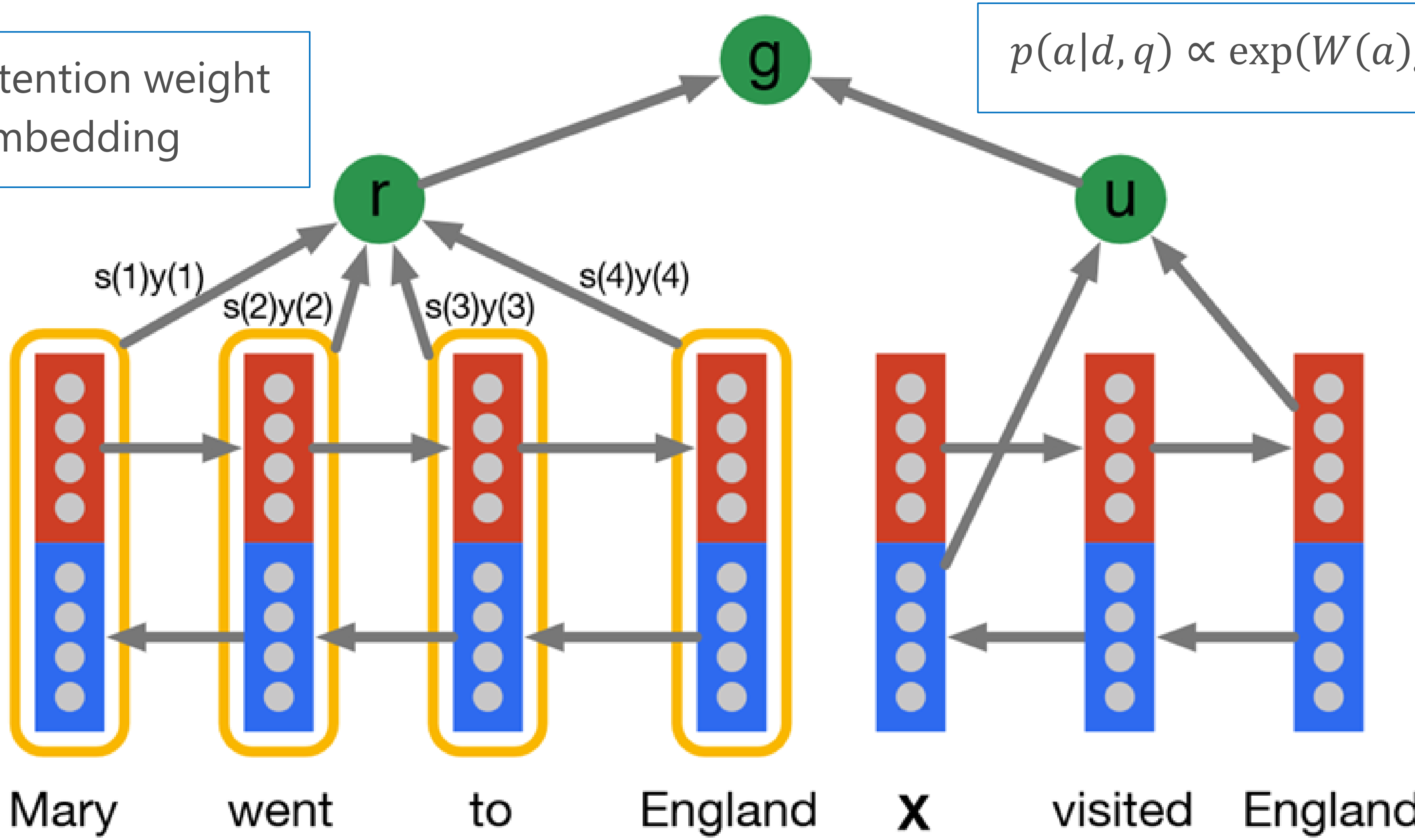
Example [Hermann et al., NIPS-15. Table 3]

Original Version	Anonymised Version
Context	
The BBC producer allegedly struck by Jeremy Clarkson will not press charges against the “Top Gear” host, his lawyer said Friday. Clarkson, who hosted one of the most-watched television shows in the world, was dropped by the BBC Wednesday after an internal investigation by the British broadcaster found he had subjected producer Oisin Tymon “to an unprovoked physical and verbal attack.” …	the *ent381* producer allegedly struck by *ent212* will not press charges against the “ *ent153* ” host , his lawyer said friday . *ent212* , who hosted one of the most - watched television shows in the world , was dropped by the *ent381* wednesday after an internal investigation by the *ent180* broadcaster found he had subjected producer *ent193* “ to an unprovoked physical and verbal attack . ” …
Query	
Producer **X** will not press charges against Jeremy Clarkson, his lawyer says.	producer **X** will not press charges against *ent212* , his lawyer says .
Answer	
Oisin Tymon	*ent193*

Neural Network Models – Attentive Reader

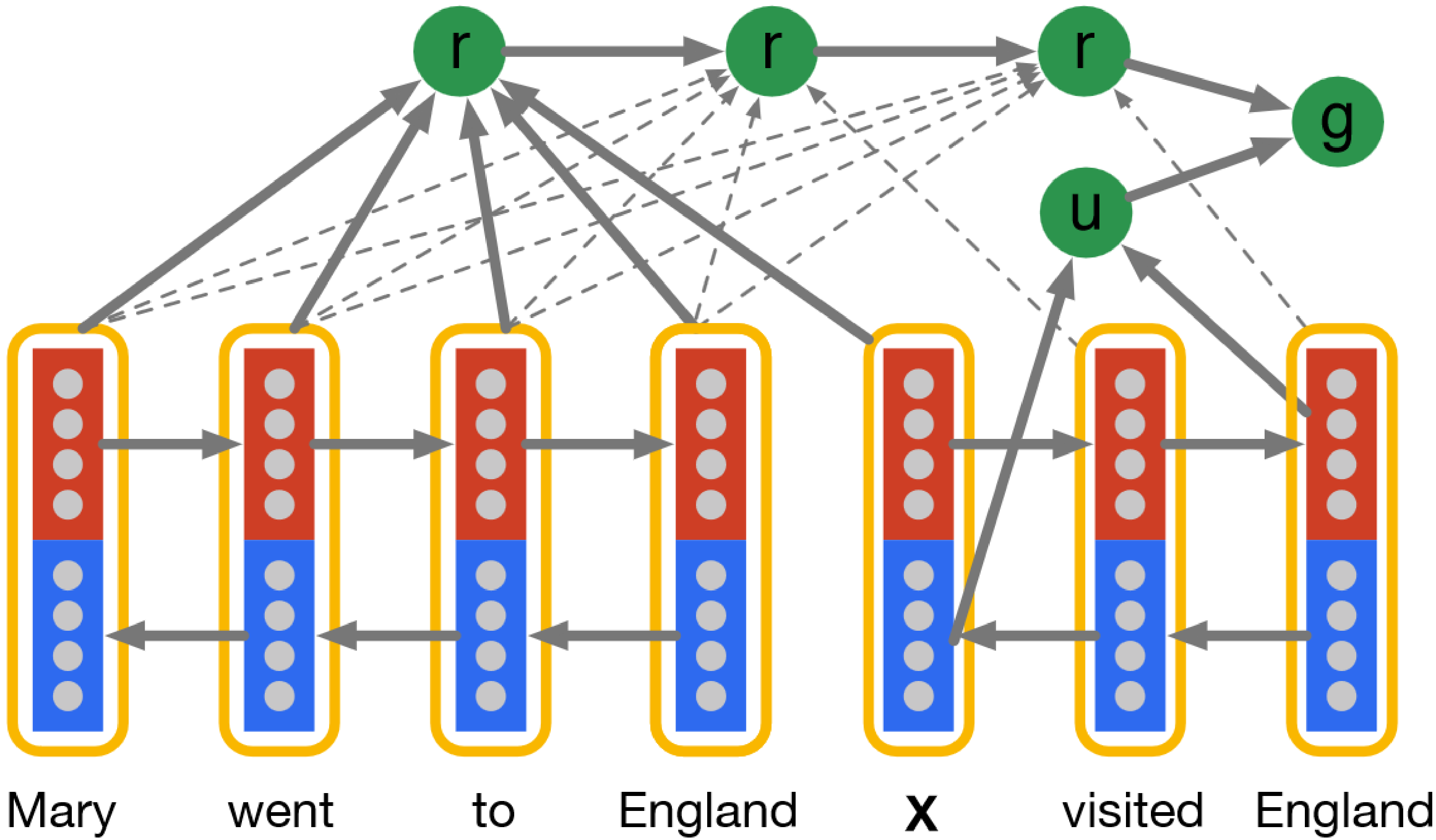
s: attention weight
y: embedding

$p(a|d,q) \propto \exp(W(a)g(d,q))$

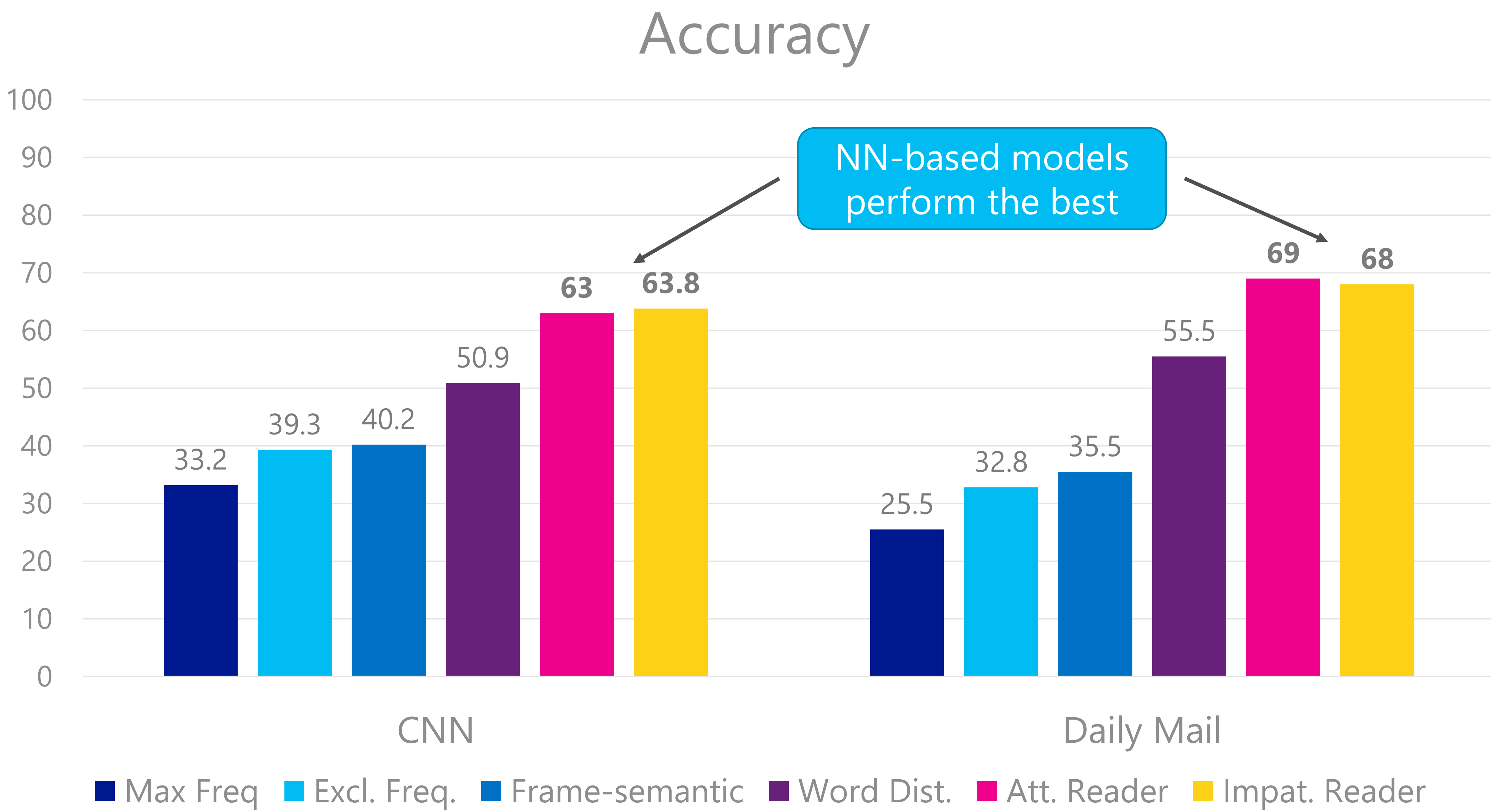


[Hermann et al., NIPS-15. Fig 1a]

Neural Network Models – Impatient Reader



[Hermann et al., NIPS-15. Fig 1b]


Accuracy
100
90
80
70
60
50
40
30
20
10
0
NN-based models perform the best
33.2
39.3
40.2
50.9
63
63.8
25.5
32.8
35.5
55.5
69
68
CNN
Daily Mail
Max Freq
Excl. Freq.
Frame-semantic
Word Dist.
Att. Reader
Impat. Reader

A Thorough Examination… [Chen et al. ACL-16]

- Challenges & Questions
  - A clever way of creating large supervised data, but an artificial task
  - Unclear what level of reading comprehension needed
- Good News – The task is not really difficult!
  - An entity-centric classifier with simple features works comparably
  - A variant of the Attentive Reader model achieves the new best result
- Bad News – The task is not really difficult!
  - Not much “comprehension” is needed
  - Probably reached the ceiling (25% questions unanswerable)

Interim summary

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

- Continuous Word Representations & Lexical Semantics
- Knowledge Base Embedding
- KB-based Question Answering & Machine Comprehension

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
- DeepMind Q&A dataset http://cs.nyu.edu/~kcho/DMQA/
- Stanford Q&A dataset https://stanford-qa.com/