Deep Learning for Web Search and Natural Language Processing

Jianfeng Gao

Deep Learning Technology Center (DLTC)

Microsoft Research, Redmond, USA

WSDM 2015, Shanghai, China

*Thank Li Deng and Xiaodong He, with whom we participated in the previous ICASSP2014 and CIKM2014 versions of this tutorial

Mission of Machine (Deep) Learning

"Real" world	Data (collected/labeled)
"Artificial" world	Model (architecture)
Link the two worlds	Training (algorithm)

Outline

- The basics
  - Background of deep learning
  - A query classification problem
  - A single neuron model
  - A deep neural network (DNN) model
  - Potentials and problems of DNN
  - The breakthrough after 2006
- Deep Semantic Similarity Models (DSSM) for text processing
- Recurrent Neural Networks

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. →

The New York Times

Scientists See Promise in Deep-Learning Programs

John Markoff November 23, 2012

Geoff Hinton

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.

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?

Impact of deep learning in speech technology

BloombergBusinessweek
Technology

Acquisitions

The Race to Buy the Human Brains Behind Deep Learning Machines

By Ashlee Vance | January 27, 2014

intelligence projects. “DeepMind is bona fide in terms of its research capabilities and depth,” says Peter Lee, who heads Microsoft Research.

According to Lee, Microsoft, Facebook (FB), and Google find themselves in a battle for deep learning talent. Microsoft has gone from four full-time deep learning experts to 70 in the past three years. “We would have more if the talent was there to be had,” he says. “Last year, the cost of a top, world-class deep learning expert was about the same as a top NFL quarterback prospect. The cost of that talent is pretty remarkable.”

A query classification problem

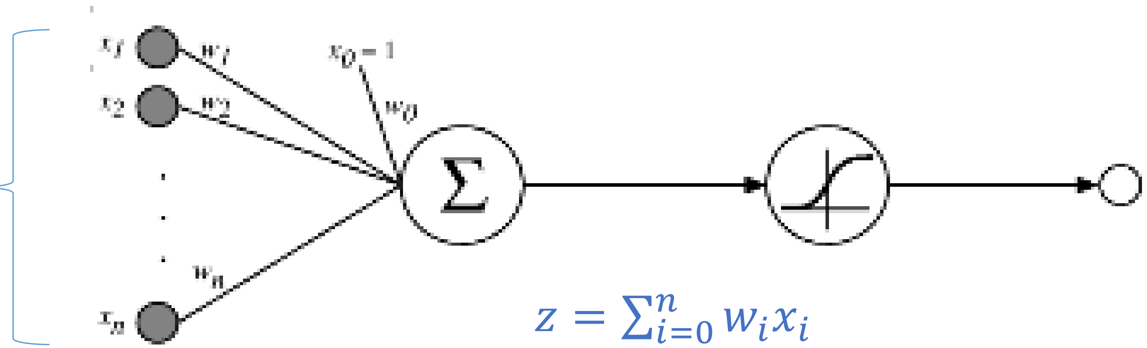
- Given a search query q, e.g., “denver sushi downtown”
- Identify its domain c e.g.,
  - Restaurant
  - Hotel
  - Nightlife
  - Flight
  - etc.
- So that a search engine can tailor the interface and result to provide a richer personalized user experience

A single neuron model

- For each domain c, build a binary classifier
  - Input: represent a query q as a vector of features $x = [x_1, \dots x_n]^T$
  - Output: $y = P(c|q)$
  - q is labeled c is $P(c|q) > 0.5$
- 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

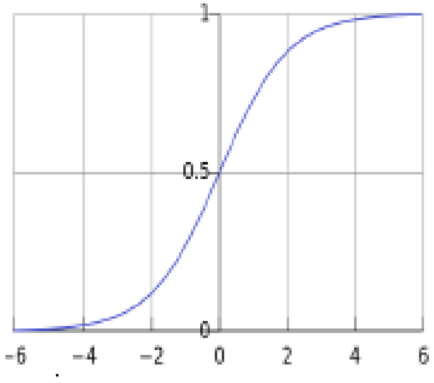


$z = \sum_{i=0}^{n} w_i x_i$

Output: $P(c|q)$

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

- w: weight vector to be learned
- z: weighted sum of input features
- σ: the logistic function
  - Turn a score to a probability
  - A sigmoid non-linearlity (activation function), essential in multi-layer/deep neural network models



Model training: how to assign w

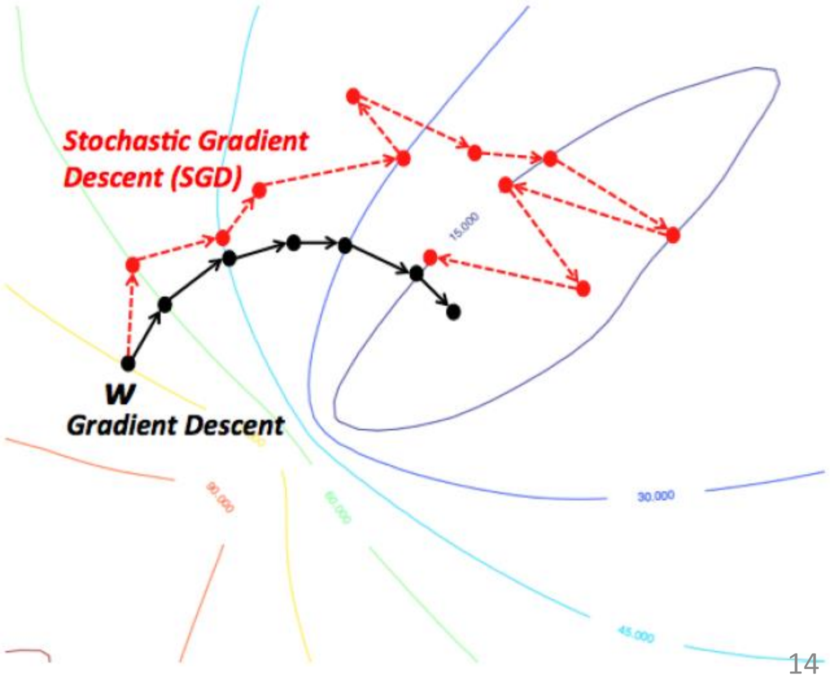
- Training data: a set of $\left(x^{(m)}, y^{(m)}\right)_{m=\{1,2,\ldots,M\}}$ pairs
  - Input $x^{(m)} \in R^n$
  - Output $y^{(m)} = \{0,1\}$
- Goal: learn function $f: x \rightarrow y$ to predict correctly on new input x
  - Step 1: choose a function family, e.g.,
    - neural networks, logistic regression, support vector machine, in our case
    - $f(x) = \sigma(\sum_{i=0}^{n} w_i x_i) = \sigma(w^T x)$
  - Step 2: optimize parameters w on training data, e.g.,
    - minimize a loss function (mean square error loss)
    - $\min_{w} \sum_{m=1}^{M} L^m$
    - where $L^{(m)} = \frac{1}{2}\left(f_w\left(x^{(m)}\right) - y^{(m)}\right)^2$

Training the single neuron model, w

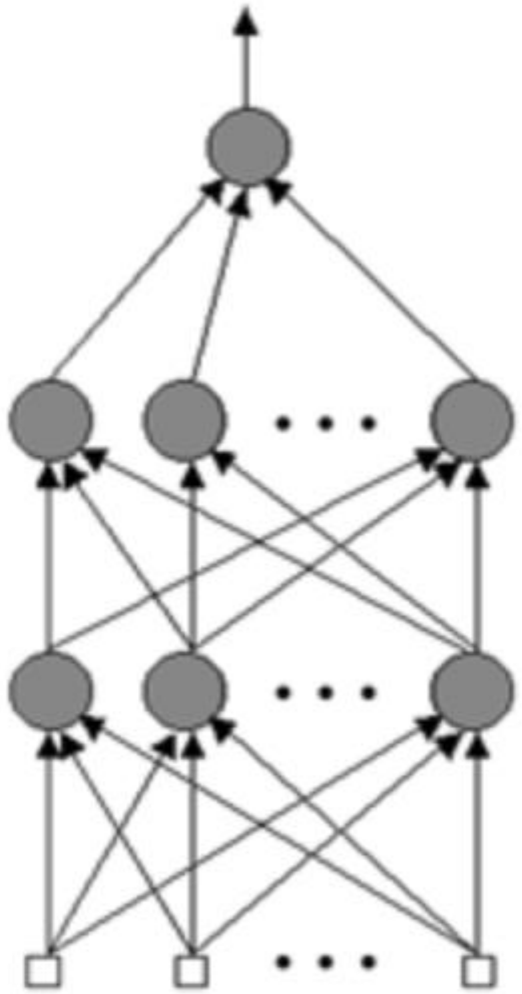
- Stochastic gradient descent (SGD) algorithm
  - Initialize w randomly
  - Update for each training sample until convergence: $w^{new} = w^{old} - \eta \frac{\partial L}{\partial w}$
- Mean square error loss: $L = \frac{1}{2}(\sigma(w^T x) - y)^2$
- Gradient: $\frac{\partial L}{\partial w} = \delta \sigma'(z) x$
  - $z = w^T x$
  - Error: $\delta = \sigma(z) - y$
  - Derivative of sigmoid $\sigma'(z) = \sigma(z)\big(1 - \sigma(z)\big)$

SGD vs. gradient descent

- Gradient descent is a batch training algorithm
  - update w per batch of training samples
  - goes in steepest descent direction
- SGD is noisy descent (but faster per iteration)
- Loss function contour plot (Duh 2014)
  - $\sum_{m=1}^{M} \frac{1}{2}(\sigma(w^T x) - y)^2 + \|w\|$



Multi-layer (deep) neural networks



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

Vector w

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

Projection matrix $\mathbf{W}_2$

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

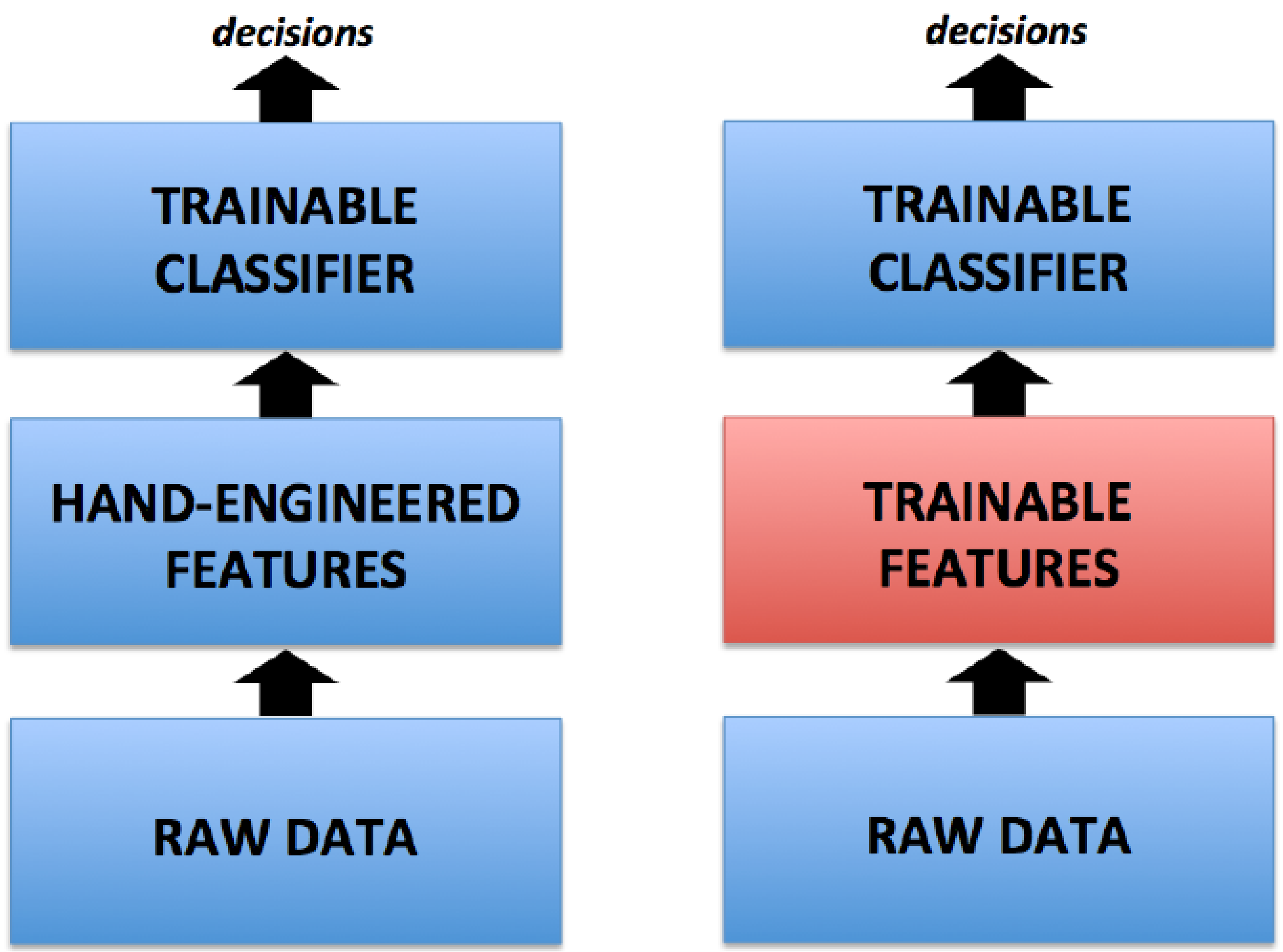
Projection matrix $\mathbf{W}_1$

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).

Standard Machine Learning Process | **Deep Learning**



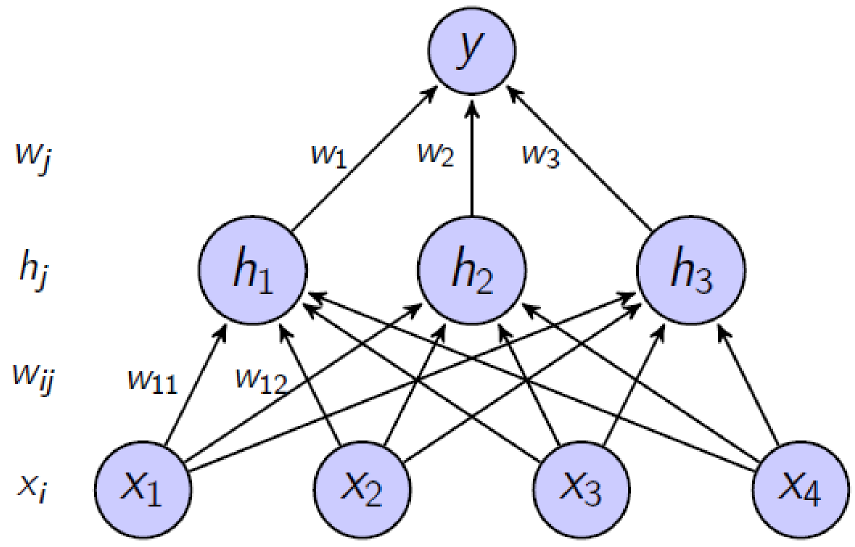
Adapted from [Duh 2014]

Revisit the activation function: σ

- Assuming a L-layer neural network
  - $y = \mathbf{W}_L \sigma\left(\ldots \sigma\left(\mathbf{W}_2 \sigma(\mathbf{W}_1 x)\right)\right)$, where y is the output vector
- If σ is a linear function, then L-layer neural network is compiled down into a single linear transform
- σ: map scores to probabilities
  - Useful in prediction as it transforms the neuron weighted sum into the interval [0..1]
  - Unnecessary for model training except in the Boltzman machine or graphical models

Training a two-layer neural net

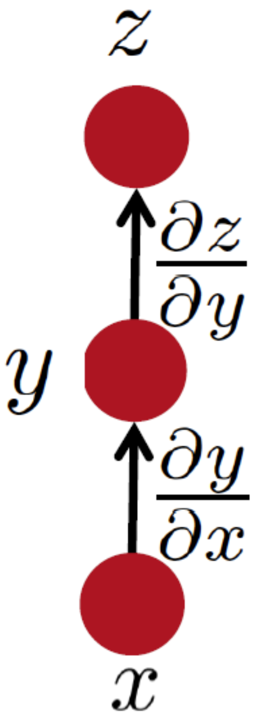
- Training data: a set of $\left(x^{(m)}, y^{(m)}\right)_{m=\{1,2,\ldots,M\}}$ pairs
  - Input $x^{(m)} \in R^n$
  - Output $y^{(m)} = \{0,1\}$
- Goal: learn function $f: x \rightarrow y$ to predict correctly on new input x
  - $f(x) = \sigma\left(\sum_j w_j \cdot \sigma\left(\sum_i w_{ij} x_i\right)\right)$
  - Optimize parameters w on training data via
    - minimize a loss function: $\min_w \sum_{m=1}^{M} L^m$
    - where $L^{(m)} = \frac{1}{2}\left(f_w\left(x^{(m)}\right) - y^{(m)}\right)^2$



Training neural nets: back-propagation

- Stochastic gradient descent (SGD) algorithm
  - $w^{new} = w^{old} - \eta \frac{\partial L}{\partial w}$
- $\frac{\partial L}{\partial w}$: sample-wise loss w.r.t. parameters
- Need to apply the derivative chain rule correctly
  - $z = f(y)$
  - $y = g(x)$
  - $\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} \frac{\partial y}{\partial x}$
- A detailed discussion in [Socher & Manning 2013]

Simple chain rule



$$\Delta z = \frac{\partial z}{\partial y} \Delta y$$

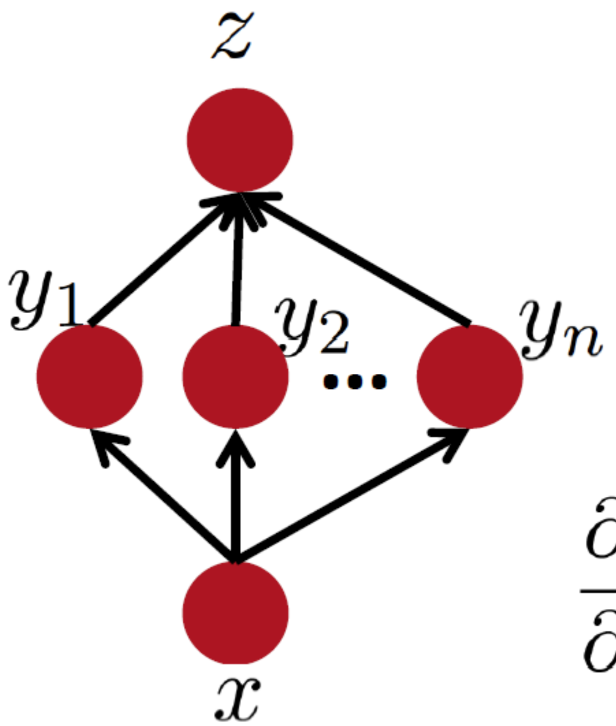
$$\Delta y = \frac{\partial y}{\partial x} \Delta x$$

$$\Delta z = \frac{\partial z}{\partial y} \frac{\partial y}{\partial x} \Delta x$$

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} \frac{\partial y}{\partial x}$$

[Socher & Manning 2013]

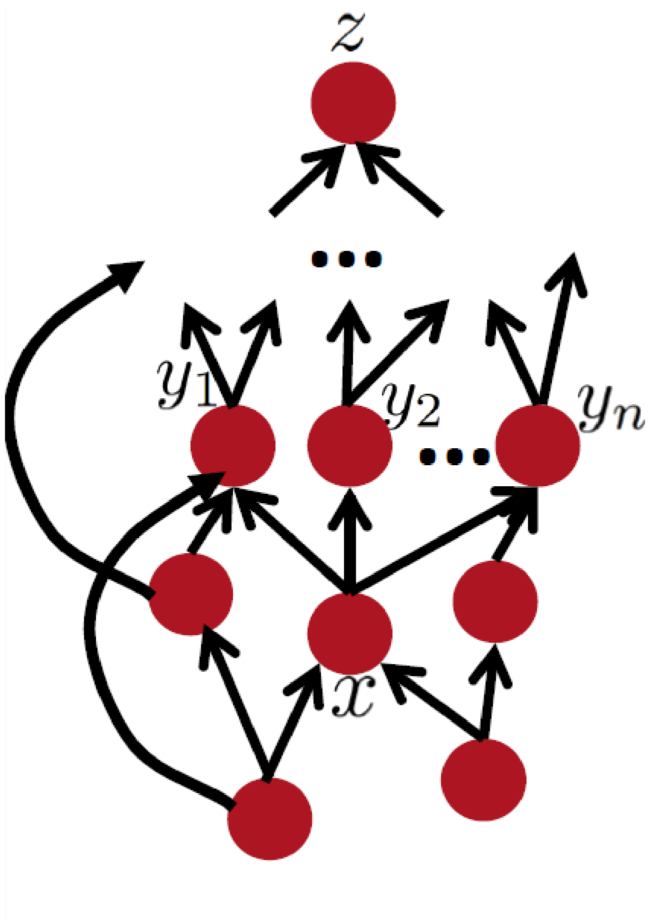
Multiple paths chain rule



$$\frac{\partial z}{\partial x} = \sum_{i=1}^{n} \frac{\partial z}{\partial y_i} \frac{\partial y_i}{\partial x}$$

[Socher & Manning 2013]

Chain rule in flow graph



Flow graph: any directed acyclic graph

node = computation result

arc = computation dependency

$\{y_1,\, y_2,\, \ldots\, y_n\}$ = successors of x

$$\frac{\partial z}{\partial x} = \sum_{i=1}^{n} \frac{\partial z}{\partial y_i} \frac{\partial y_i}{\partial x}$$

[Socher & Manning 2013]

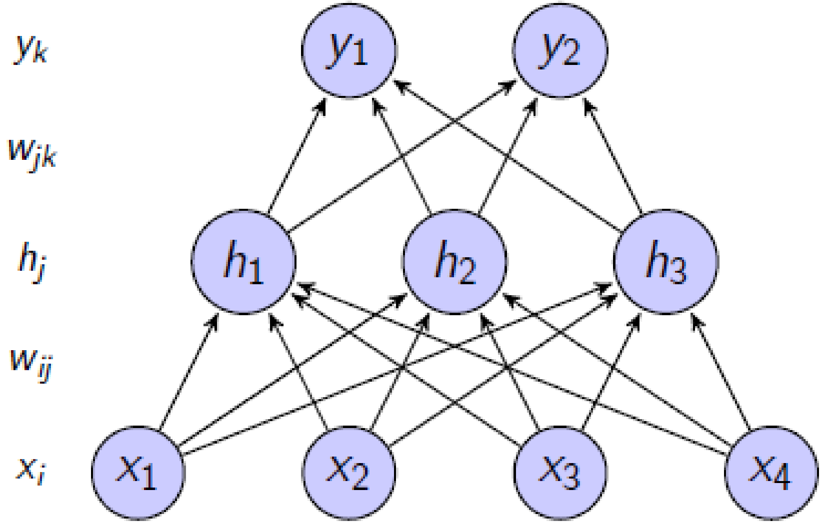
Training neural nets: back-propagation

Assume two outputs (y_1, y_2) per input x, and

Loss per sample: $L = \sum_k \frac{1}{2}(\sigma(z_k) - y_k)^2$

Forward pass:

$$y_k = \sigma(z_k),\ z_k = \sum_j w_{jk} h_j$$
$$h_j = \sigma(z_j),\ z_j = \sum_i w_{ij} x_i$$



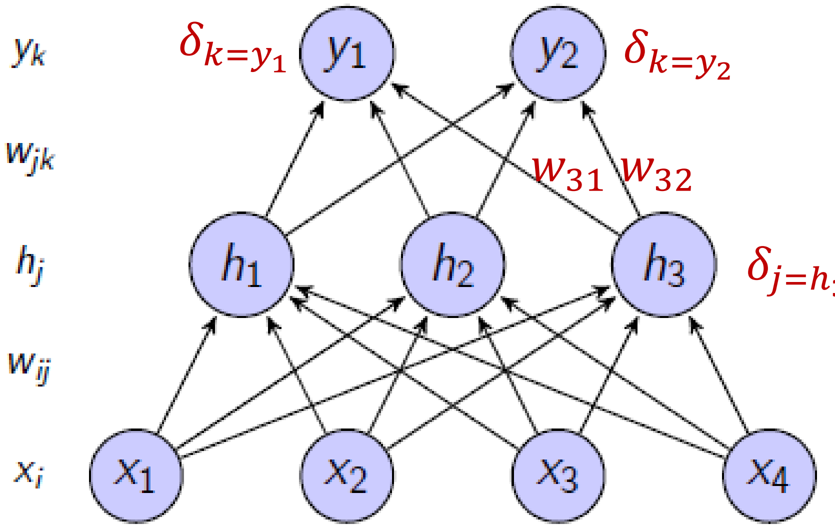
Derivatives of the weights

$$\frac{\partial L}{\partial w_{jk}} = \frac{\partial L}{\partial z_k}\frac{\partial z_k}{\partial w_{jk}} = \delta_k \frac{\partial(\sum_j w_{jk} h_j)}{\partial w_{jk}} = \delta_k h_j$$
$$\frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial z_j}\frac{\partial z_j}{\partial w_{ij}} = \delta_j \frac{\partial(\sum_i w_{ij} x_i)}{\partial w_{ij}} = \delta_j x_i$$
$$\delta_k = \frac{\partial L}{\partial z_k} = (\sigma(z_k) - y_k)\sigma'(z_k)$$
$$\delta_j = \sum_k \frac{\partial L}{\partial z_k}\frac{\partial z_k}{\partial z_j} = \sum_k \delta_k \frac{\partial}{\partial z_j}\left(\sum_j w_{jk}\sigma(z_j)\right) = \left(\sum_k \delta_k w_{jk}\right)\sigma'(z_j)$$

Adapted from [Duh 2014]

Training neural nets: back-propagation

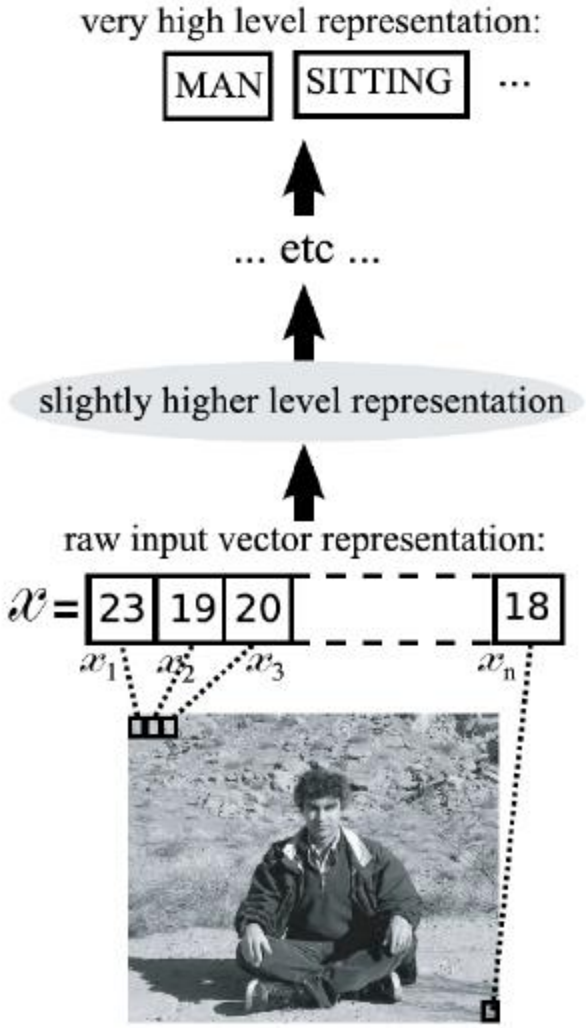
- All updates involve some scaled error from output × input feature:
  - $\frac{\partial L}{\partial w_{jk}} = \delta_k h_j$ where $\delta_k = (\sigma(z_k) - y_k)\sigma'(z_k)$
  - $\frac{\partial L}{\partial w_{ij}} = \delta_j x_i$ where $\delta_j = \left(\sum_k \delta_k w_{jk}\right)\sigma'(z_j)$
- First compute δ_k from output layer, then δ_j for other layers and iterate.



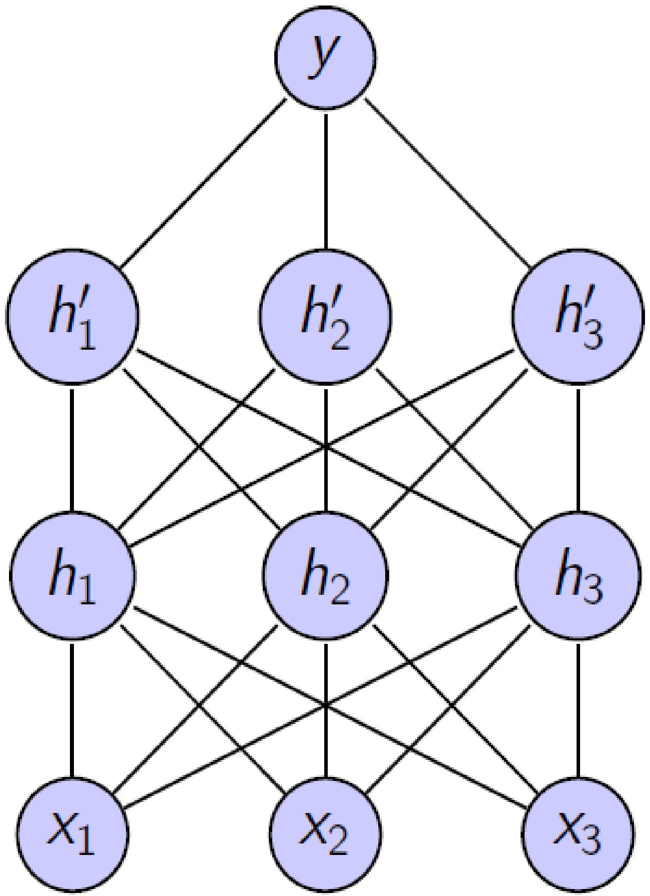
$$\delta_{j=h_3} = \left(\delta_{k=y_1} w_{31} + \delta_{k=y_2} w_{32}\right)\sigma'(z_{j=h_3})$$

Adapted from (Duh 2014)

Potential of DNN



[Bengio, 2009]

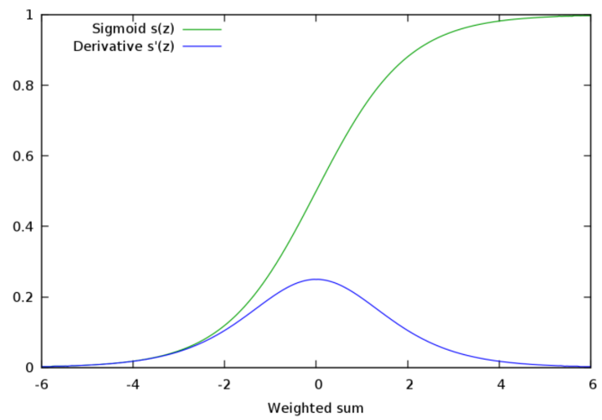


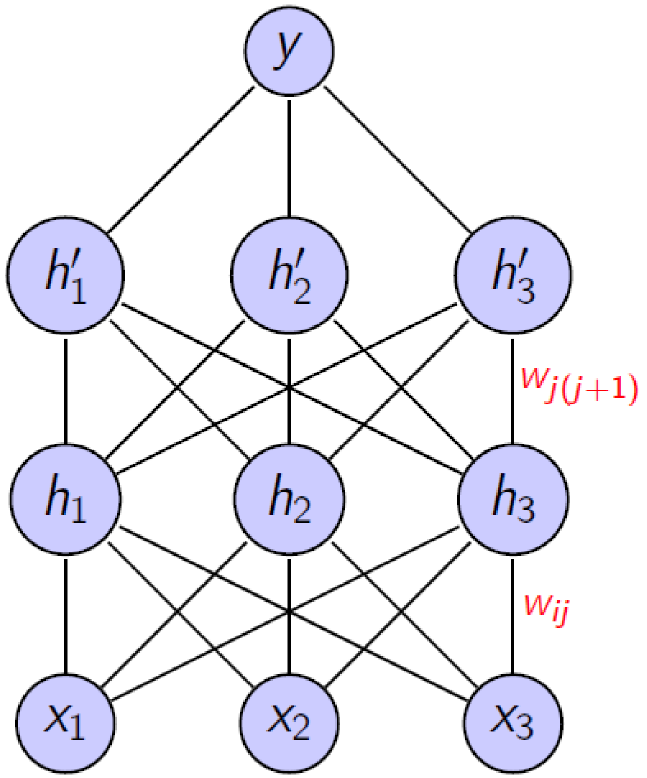
This is exactly the **single neuron model** with **hidden** features.

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

DNN is difficult to training

- Vanishing gradient problem in backpropagation
  - $\frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial z_j} \frac{\partial z_j}{\partial w_{ij}} = \delta_j x_i$
  - $\delta_j = \left(\sum_k \delta_k w_{jk}\right)\sigma'(z_j)$
  - δ_j may vanish after repeated multiplication





- Scalability problem

Many, but NOT ALL, limitations of early DNNs have been overcome

→better learning algorithms and different nonlinearities.

- →SGD can often allow the training to jump out of local optima due to the noisy gradients estimated from a small batch of samples.
- →SGD effective for parallelizing over many machines with an asynchronous mode

- Vanishing gradient problem?

→ Try deep belief net (DBN) to initialize it – Layer-wise pre-training (Hinton et al. 2006)

- Scalability problem

→ Computational power due to the use of GPU and large-scale CPU clusters

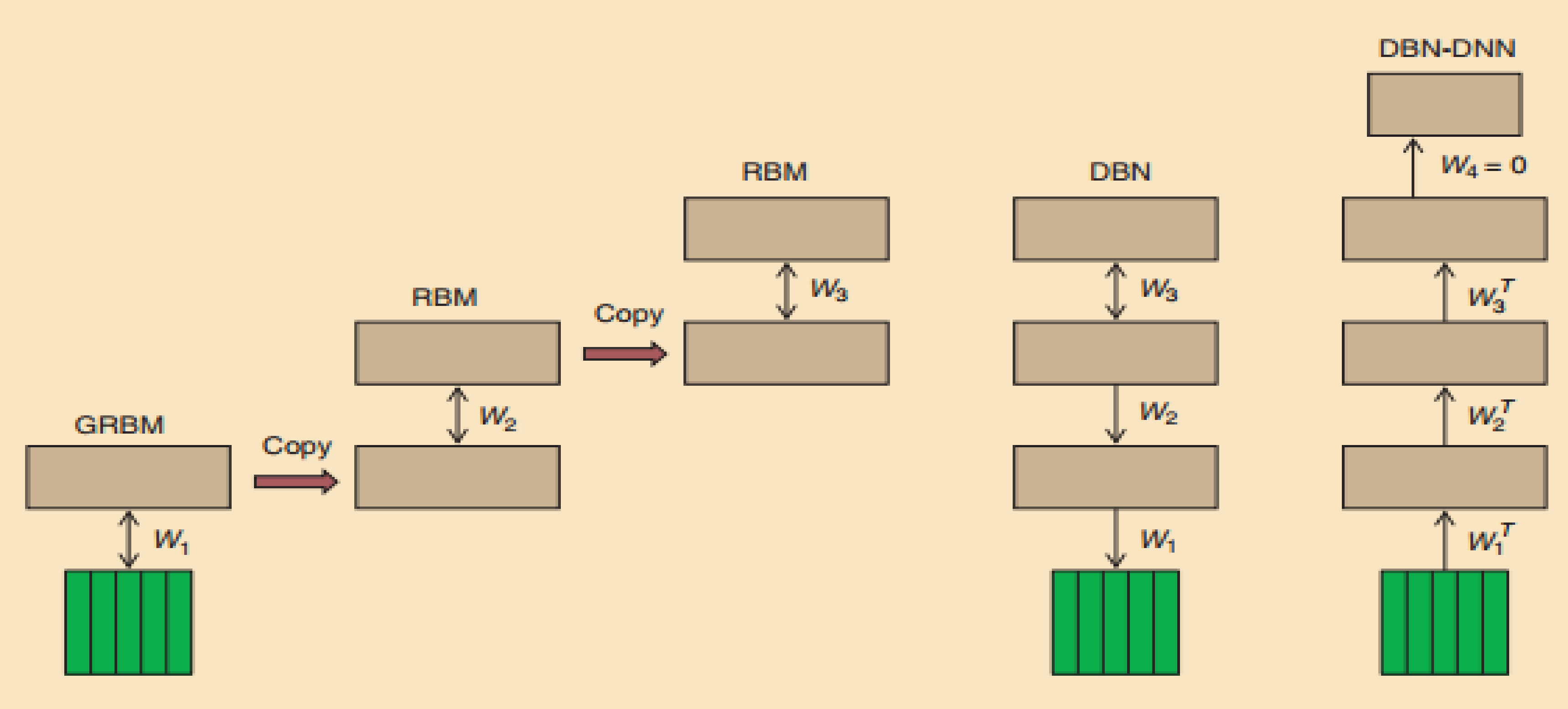
Geoff Hinton

Li Deng

Dong Yu

DNN: (Fully-Connected) Deep Neural Networks

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

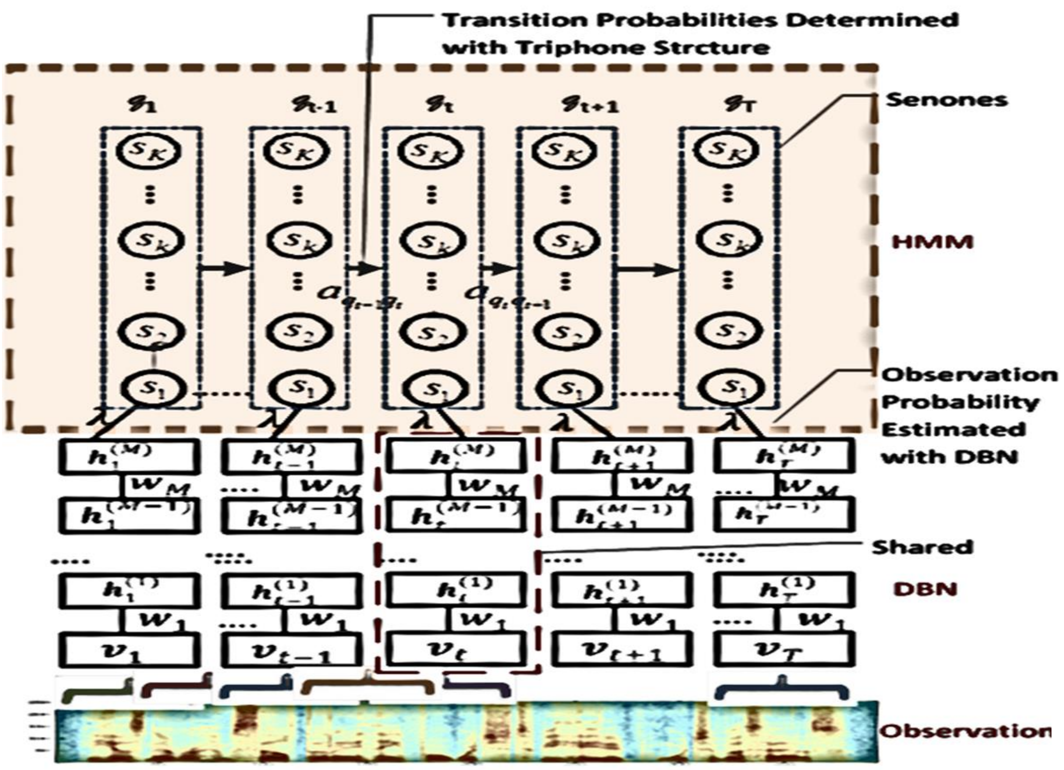


First train a stack of N models each of which has one hidden layer. Each model in the stack treats the hidden variables of the previous model as data.

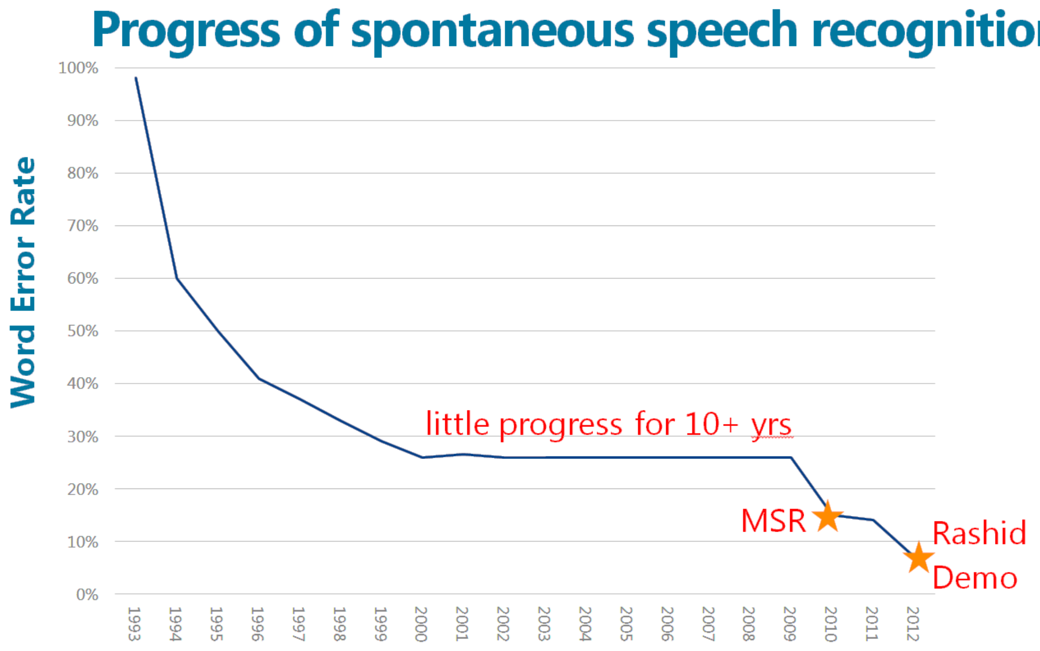
Then compose them into a single Deep Belief Network.

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



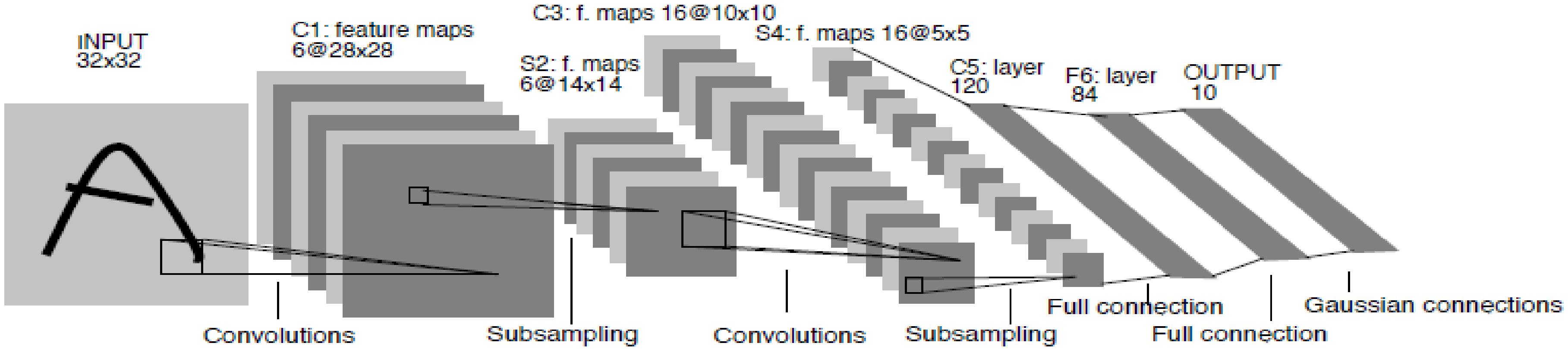
After no improvement for 10+ years by the research community…

…MSR reduced error from **~23%** to **<13%** (and under 7% for Rick Rashid’s S2S demo)!

Deep Convolutional Neural Network for Images

Yann LeCun

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

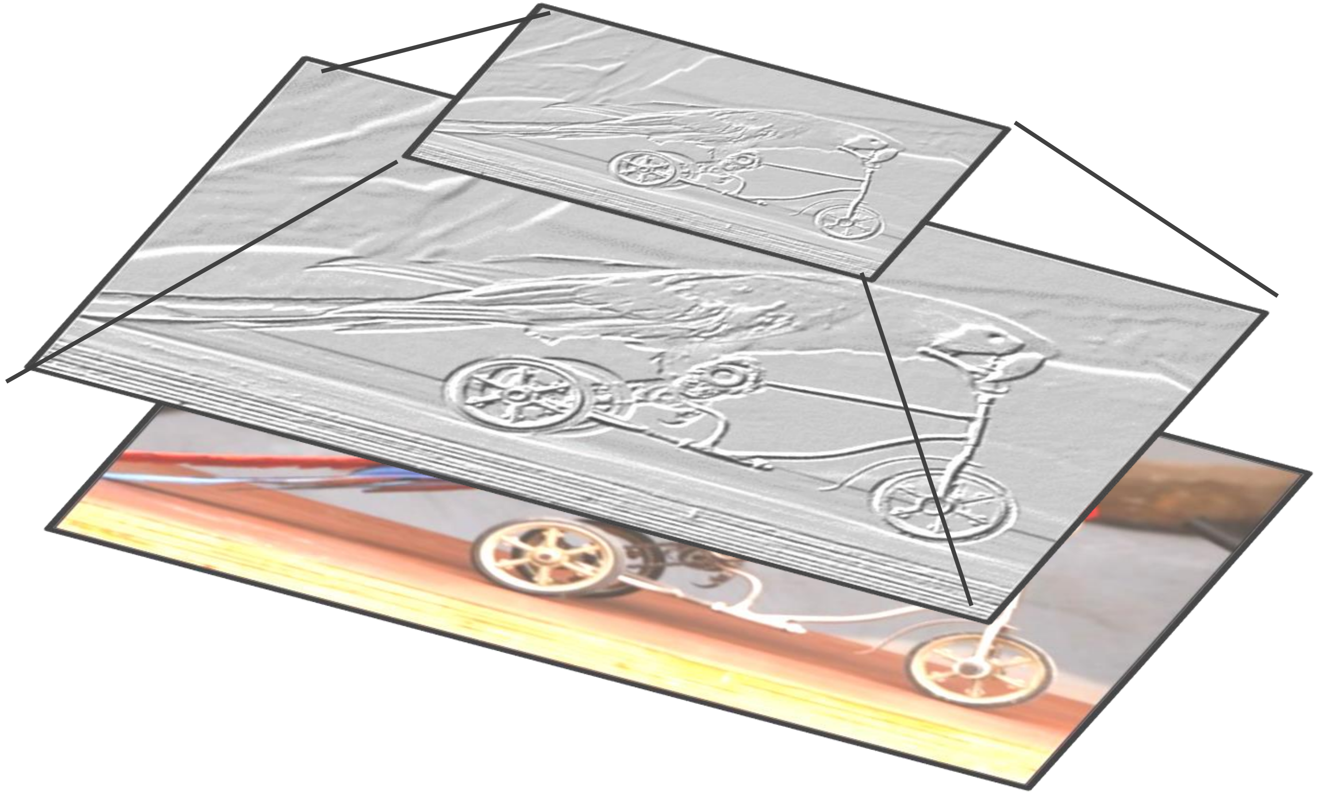


Image

Output

[LeCun et al., 1998]

A basic module of the CNN



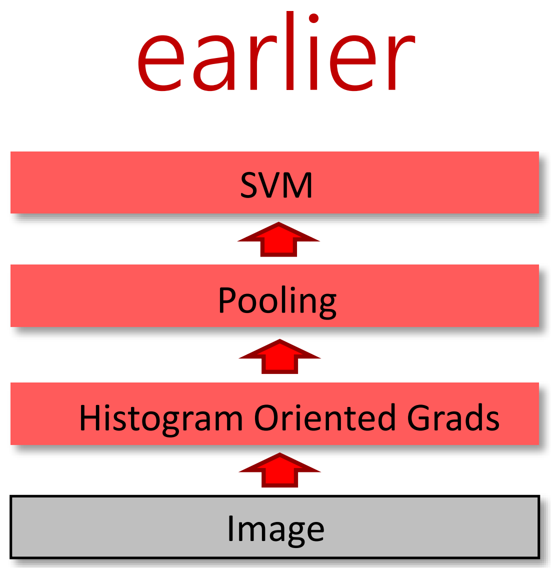
Pooling

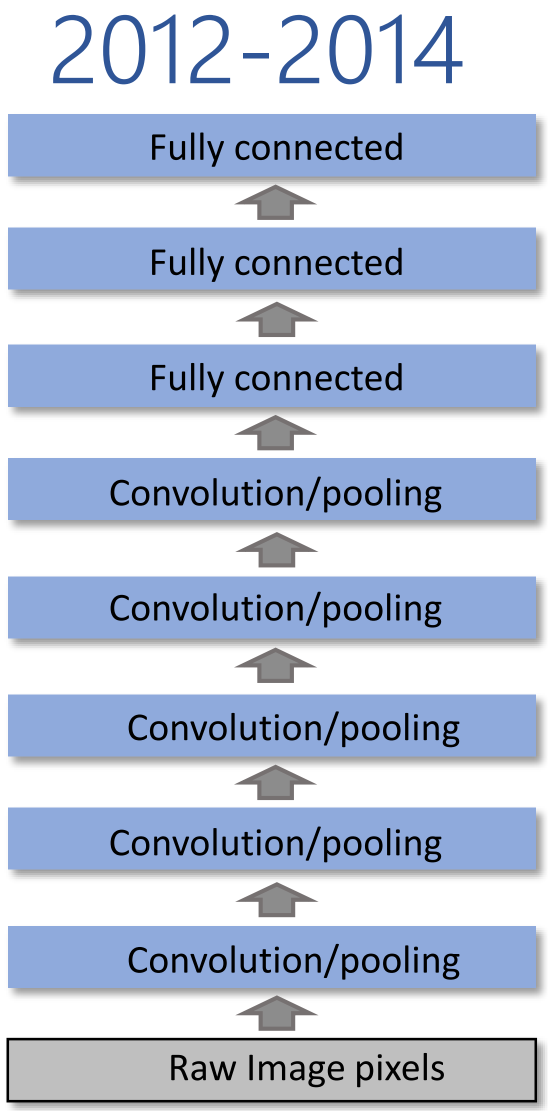
Convolution

Image

Deep Convolutional NN for Images

A paradigm shift!

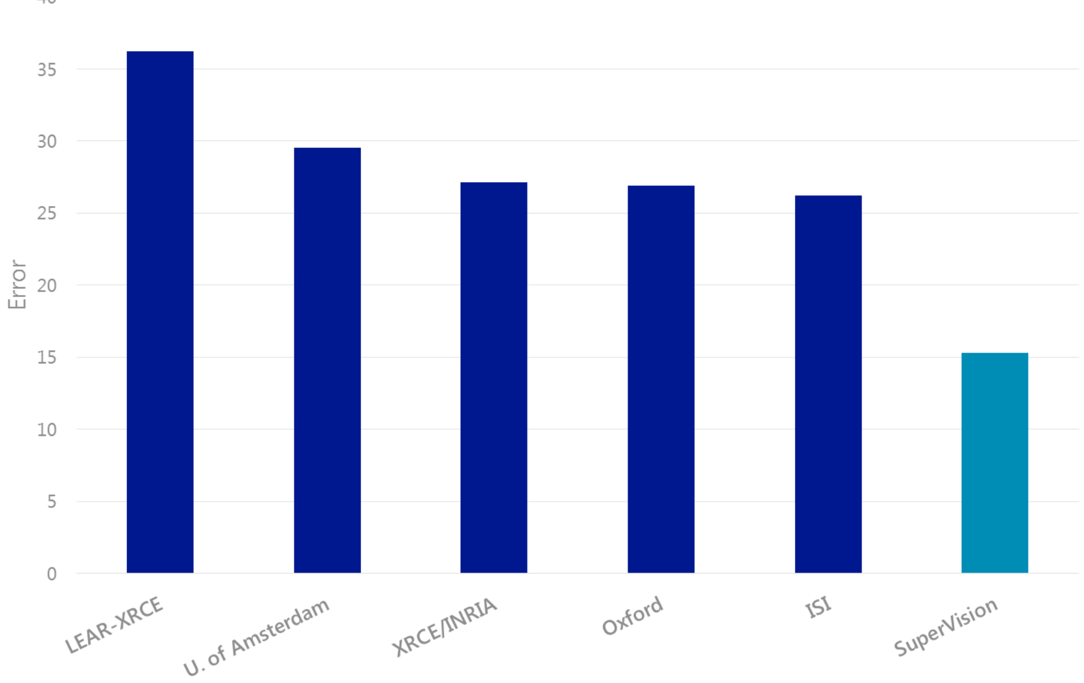




ImageNet 1K Competition

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

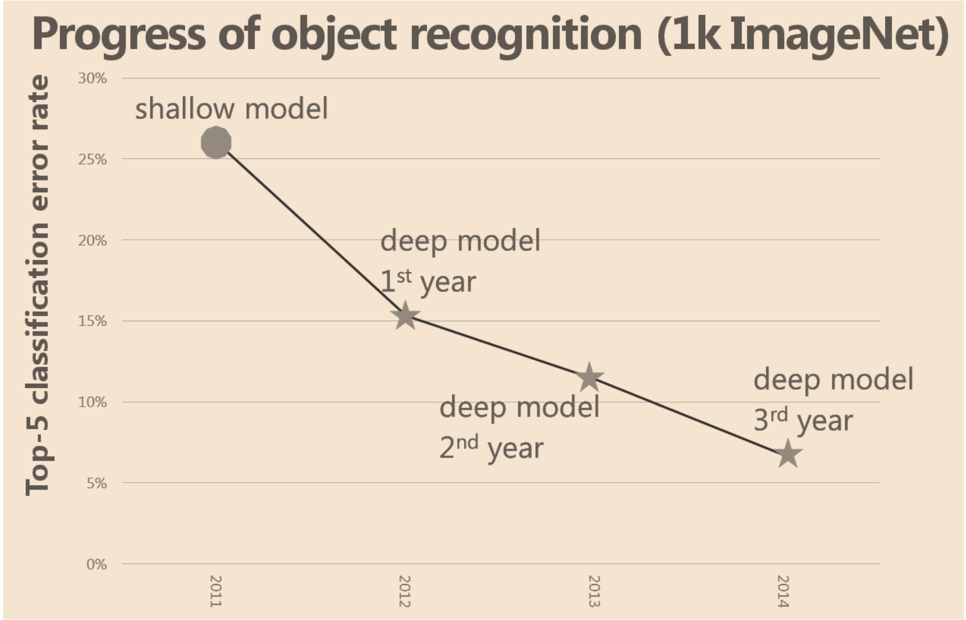
Fall 2012



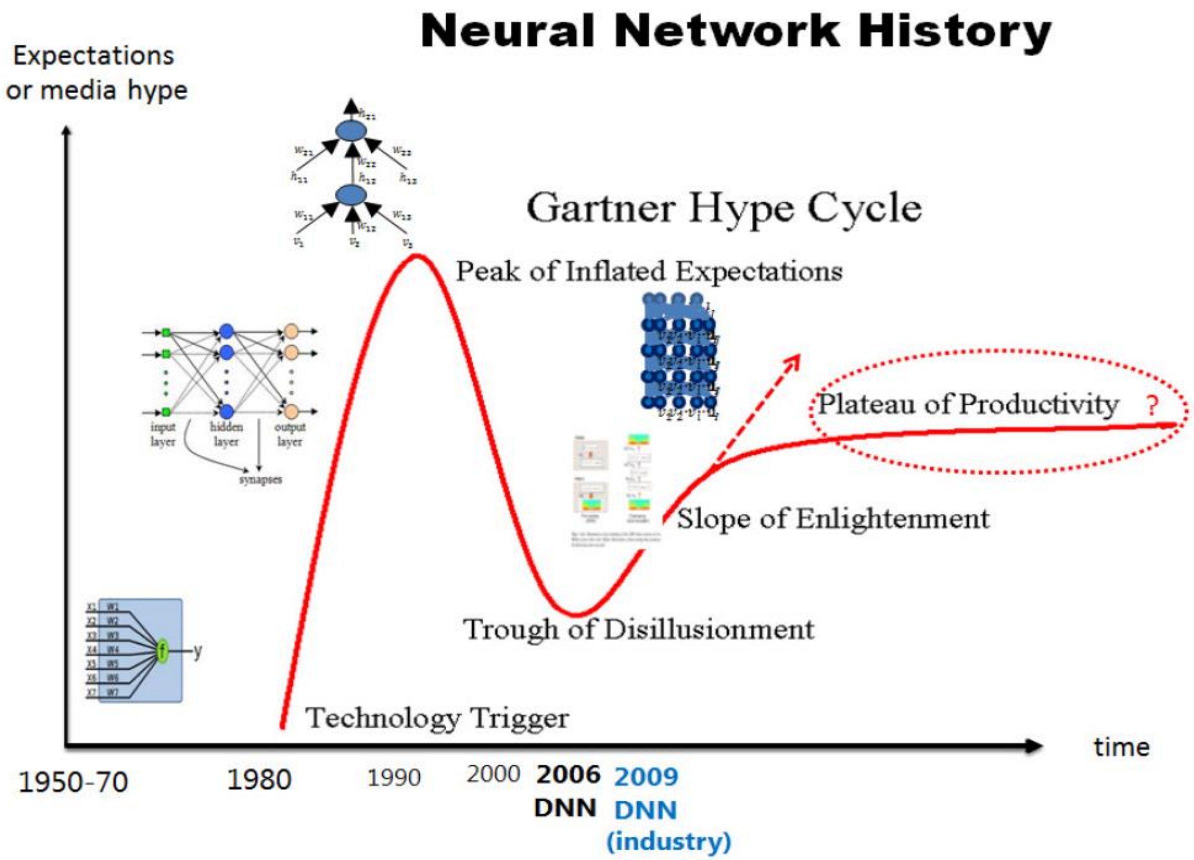
Top-5 classification error rate

Deep CNN
Univ. Toronto team

2012 - 2014



Gartner hyper cycle graph for NN history



[Deng and Yu 2014]

Useful Sites on Deep Learning

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

- Google+ Deep Learning community

Outline

- The basics
- Deep Semantic Similarity Models (DSSM) for text processing
  - What is DSSM
  - DSSM for web search ranking
  - DSSM for recommendation
  - DSSM for automatic image captioning
- Recurrent Neural Networks

Computing Semantic Similarity

- Fundamental to almost all Web search and NLP tasks, e.g.,
  - Machine translation: similarity between sentences in different languages
  - Web search: similarity between queries and documents

- Problems of the existing approaches
  - Lexical matching cannot handle language discrepancy.
  - Unsupervised word embedding or topic models are not optimal for the task of interest.

Deep Semantic Similarity Model (DSSM)

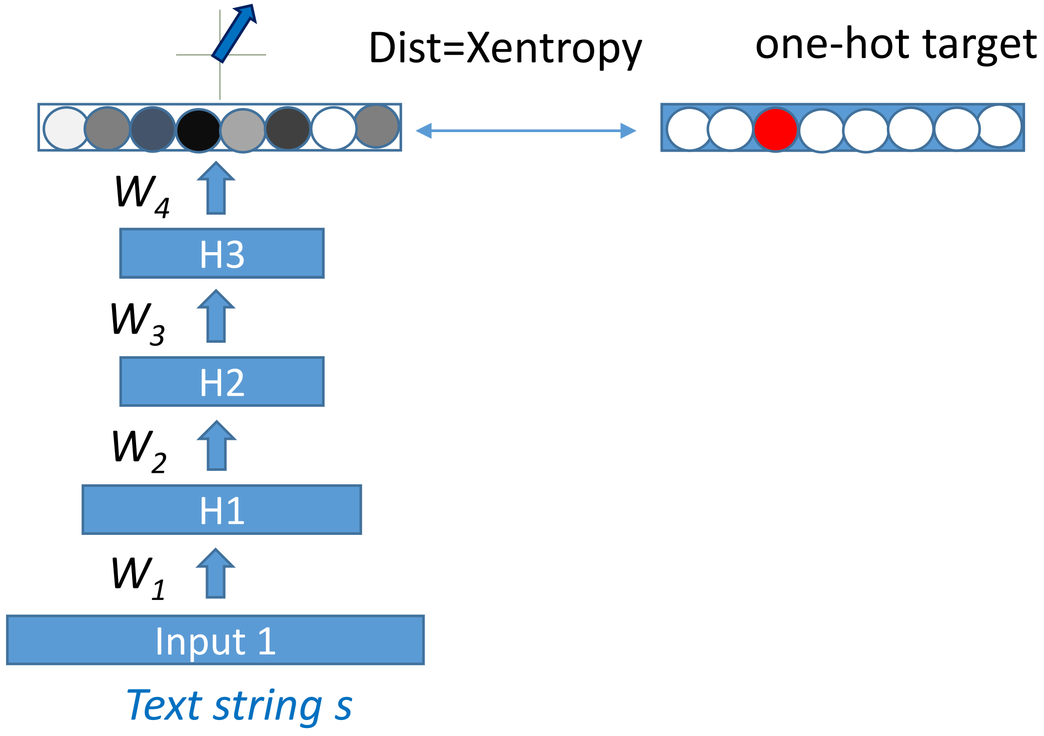
[Huang et al. 2013; Gao et al. 2014a; Gao et al. 2014b; Shen et al. 2014]

- Compute semantic similarity between two 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 et al. (2013)
- DSSM for NLP tasks

Tasks	X	Y
Web search	*Search query*	*Web document*
Automatic highlighting	*Doc in reading*	*Key phrases to be highlighted*
Contextual entity search	*Key phrase and context*	*Entity and its corresponding page*
Machine translation	*Sentence in language A*	*Translations in language B*

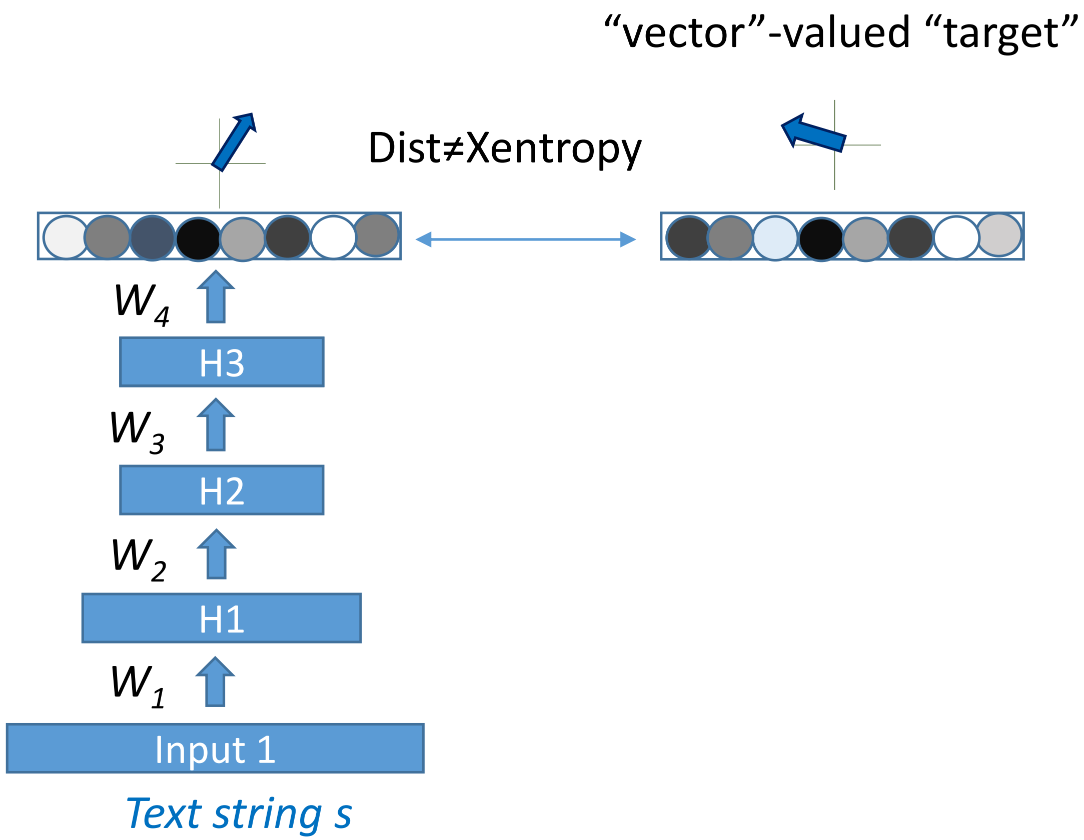
From Common Deep Models to DSSM

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



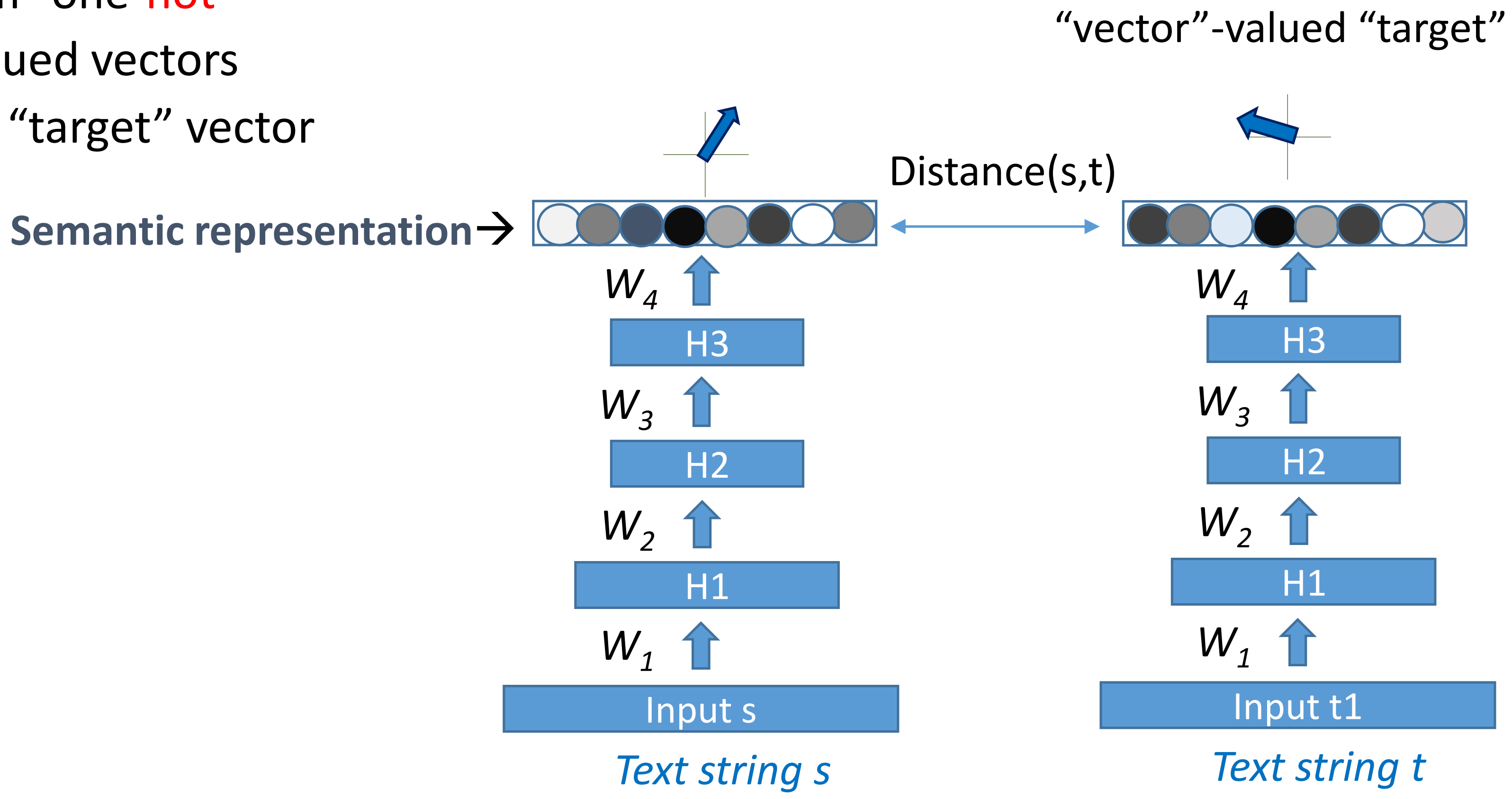
From DNN to DSSM

- DSSM
  - Deep-Structured Semantic Model, or
  - Deep Semantic Similarity Model
  - For ranking (not classification with DNN)
  - Step 1: target from “one-hot”
    to continuous-valued vectors



From DNN to DSSM

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



From DNN to DSSM

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

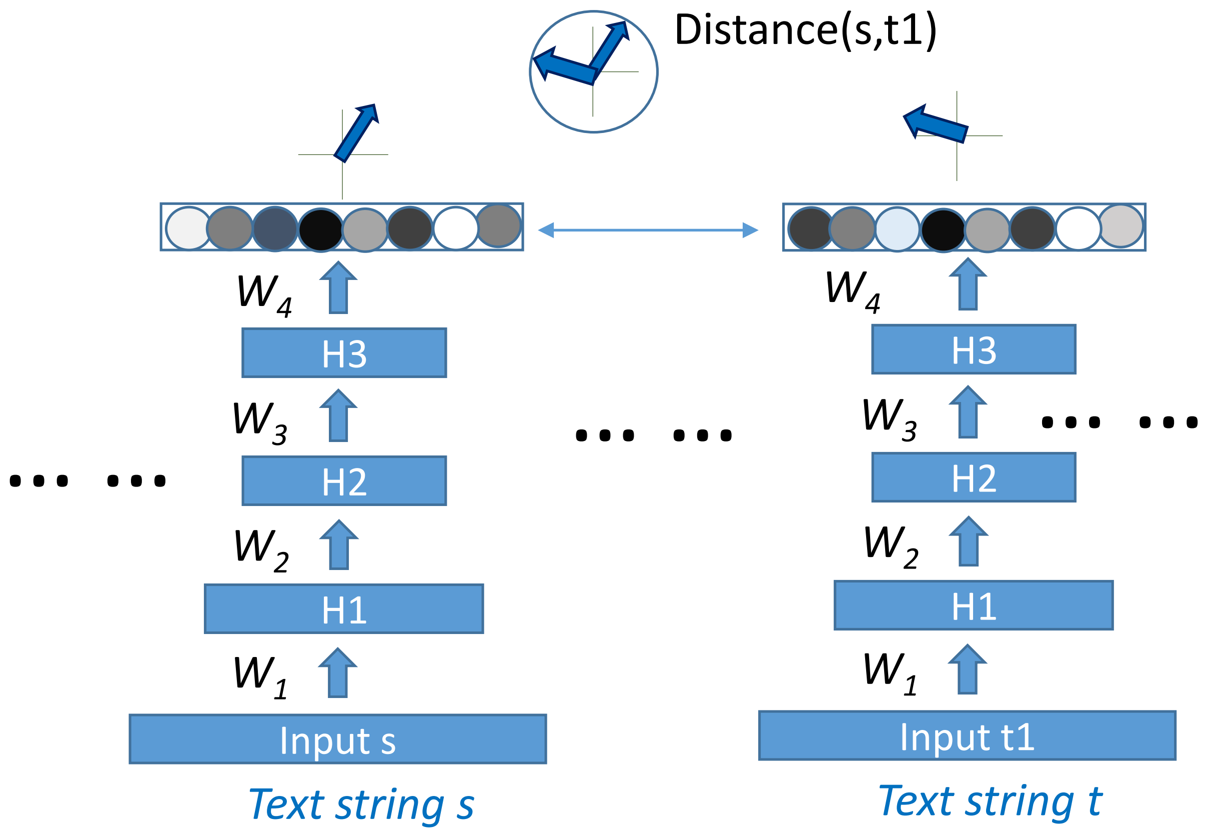
Use semantic similarity to rank documents/entities

cos(s,t1)

cos(s,t2)

cos(s,t3)

……



DSSM for web search ranking

- Task
- Model architecture
- Model training
- Evaluation
- Analysis

[Huang et al. 2013; Shen et al. 2014]

An example of web search

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 then describe the best home remedies for these specific patterns that you can use to treat the cold or influenza virus.

Cold and Flu Basics

The basic pathogenic influences are:

- **Wind**
- **Cold**
- **Heat**
- **Damp**

Wind

Theoretically, wind enters the body through the back of the neck area or nose carrying the pathogen. It first attacks the Lung system (including the sinuses) because the Lung organ system is the most external Yin organ, a thus the most vulnerable to an external invasion. External Wind invasion is marked by acute conditions with a sudden onset of symptoms.

- cold home remedy
- cold remeedy
- flu treatment
- how to deal with stuffy nose

Semantic matching between Q and D

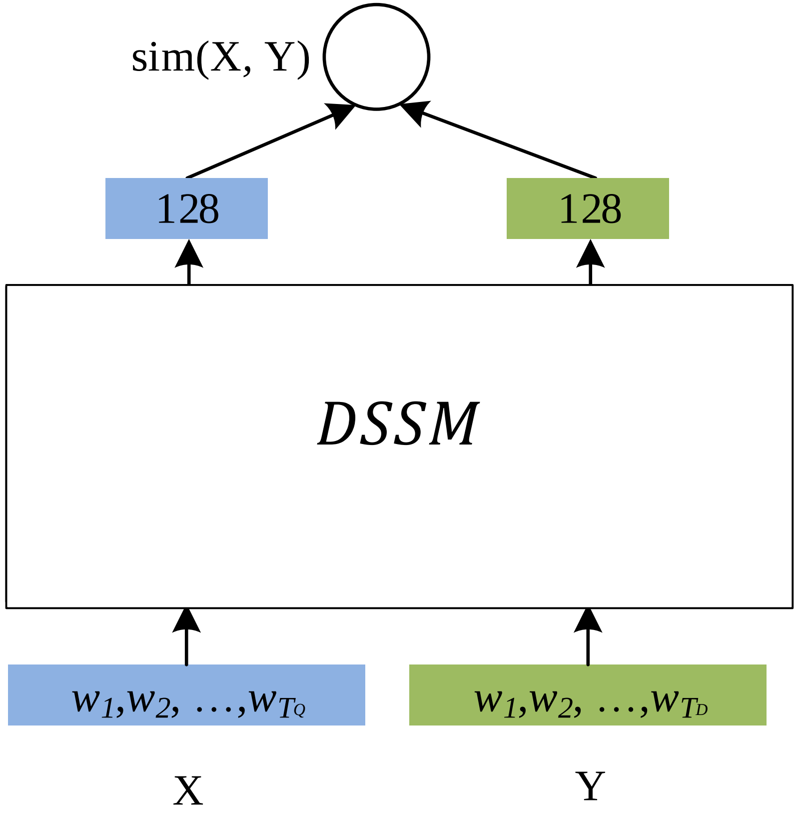
- Fuzzy keyword matching
  - Q: cold home remedy
  - D: best home remedies for cold and flu
- Spelling correction
  - Q: cold remeedies
  - D: best home remedies for cold and flu
- Query alteration/expansion
  - Q: flu treatment
  - D: best home remedies for cold and flu
- **Query/document semantic matching**
  - Q: how to deal with stuffy nose
  - D: best home remedies for cold and flu
  - Q: auto body repair cost calculator software
  - D: free online car body shop repair estimates

R&D progress

DSSM: Compute Similarity in Semantic Space

Relevance measured by cosine similarity

Word sequence x_t

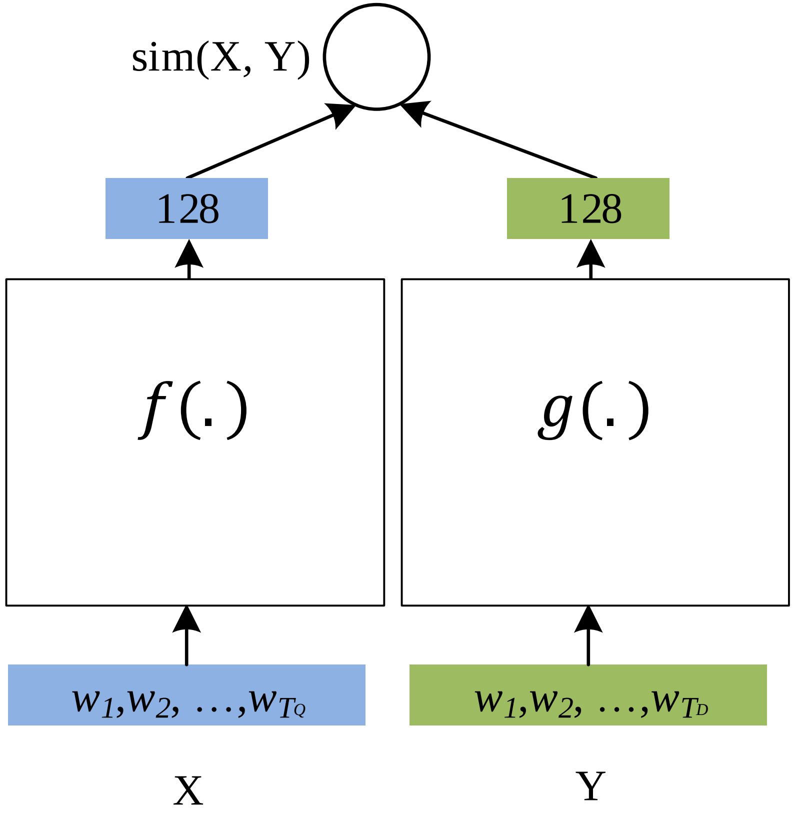


Learning: maximize the similarity between X (source) and Y (target)

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

DSSM: Compute Similarity in Semantic Space

Relevance measured by cosine similarity

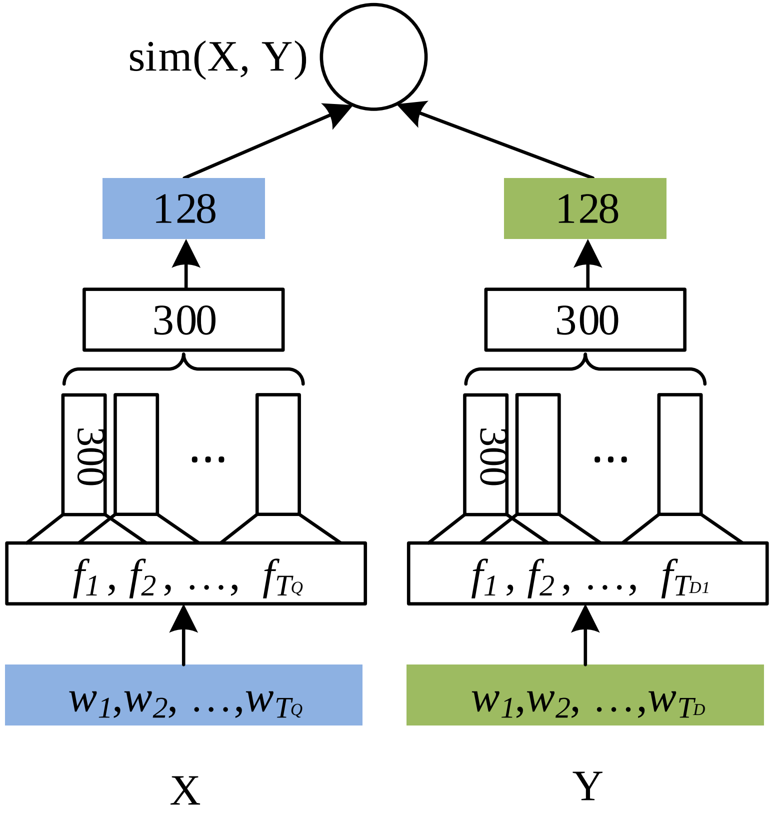
Semantic layer h

Max pooling layer v

Convolutional layer c_t

Word hashing layer f_t

Word sequence x_t



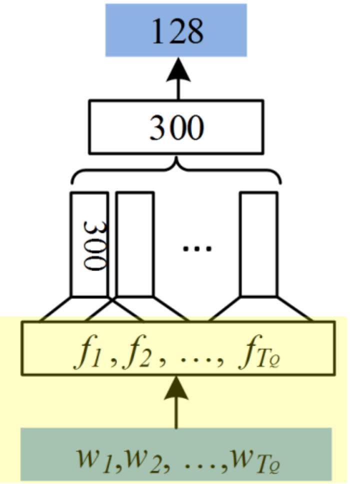
Learning: maximize the similarity between X (source) and Y (target)

Representation: use DNN to extract abstract semantic representations

Convolutional and Max-pooling layer: identify key words/concepts in X and Y

Word hashing: use sub-word unit (e.g., letter n-gram) as raw input to handle very large vocabulary

Letter-trigram Representation

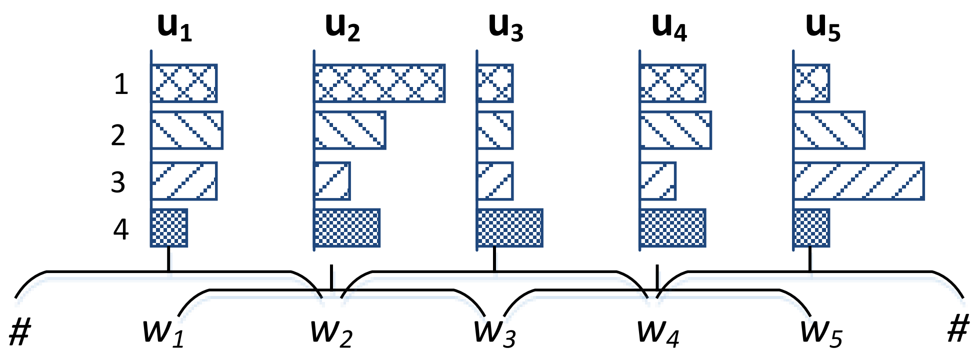


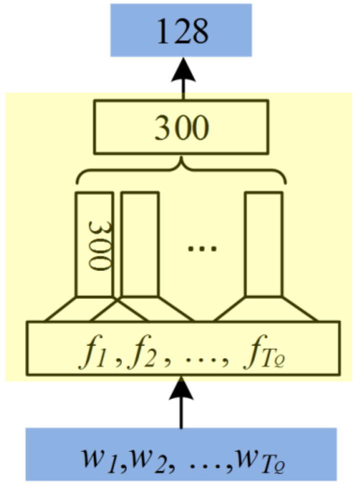
- Control the dimensionality of the input space
  - e.g., cat → #cat# → #-c-a, c-a-t, a-t-#
  - Only ~50K letter-trigrams in English; no OOV issue
- Capture sub-word semantics (e.g., prefix & suffix)
- Words with small typos have similar raw representations

- Collision: different words with same letter-trigram representation?

Vocabulary size	# of unique letter-trigrams	# of Collisions	Collision rate
40K	10,306	2	0.0050%
500K	30,621	22	0.0044%
5M	49,292	179	0.0036%

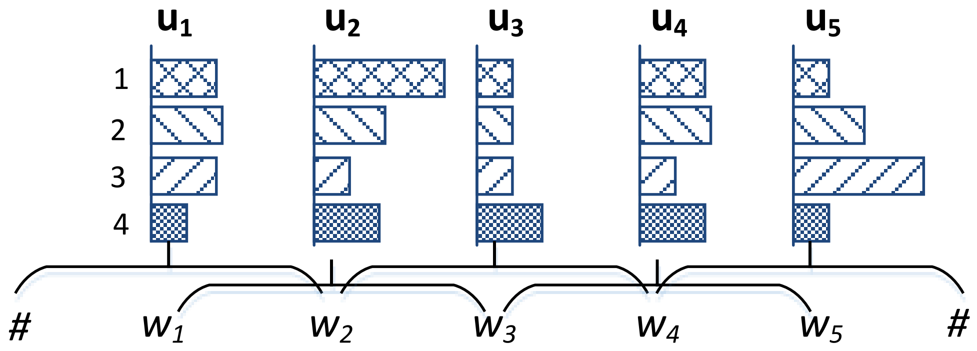
Convolutional Layer

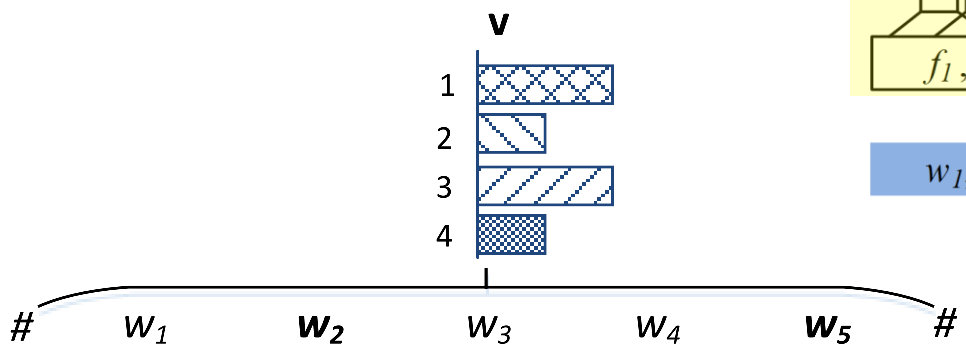


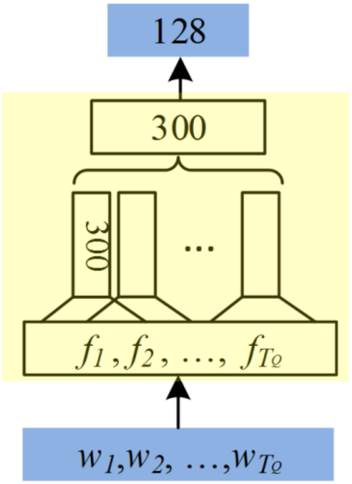


- Extract local features using convolutional layer
  - {w1, w2, w3} → topic 1
  - {w2, w3, w4} → topic 4

Max-pooling Layer

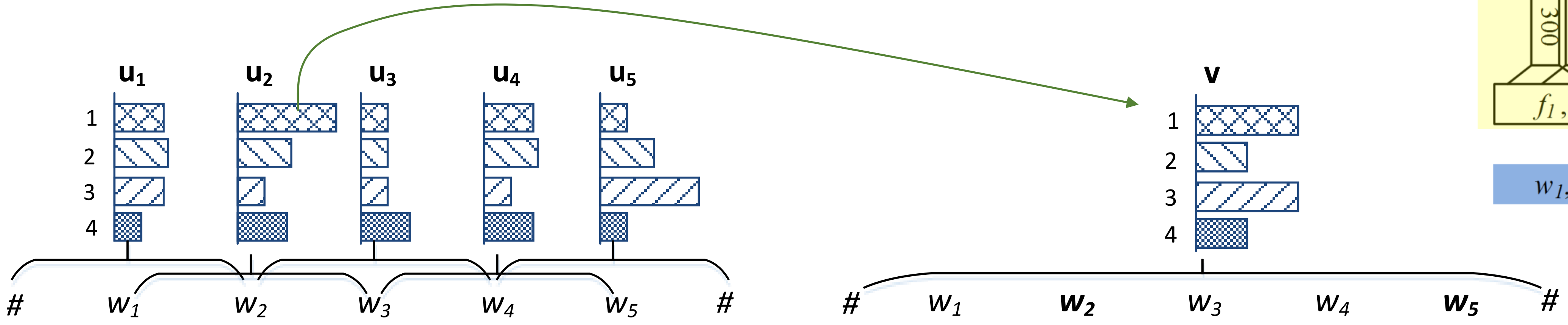


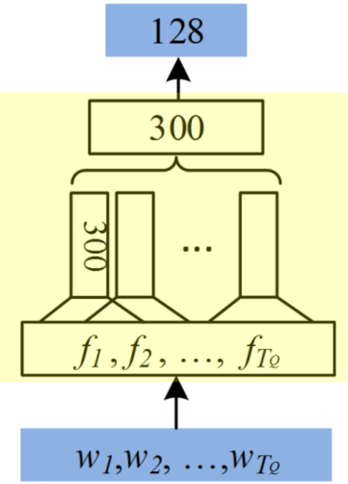




- Extract local features using convolutional layer
  - {w1, w2, w3} → topic 1
  - {w2, w3, w4} → topic 4
- Generate global features using max-pooling
  - Key topics of the text → topics 1 and 3
  - keywords of the text: w2 and w5

Max-pooling Layer



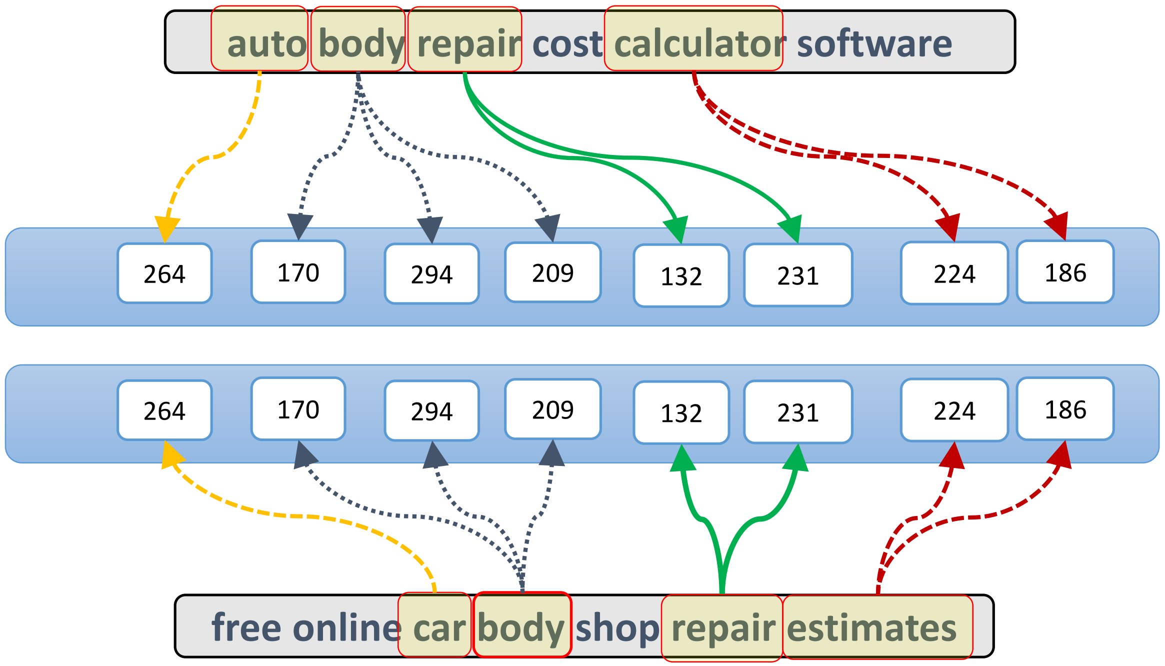


- Extract local features using convolutional layer
  - {w1, w2, w3} → topic 1
  - {w2, w3, w4} → topic 4
- Generate global features using max-pooling
  - Key topics of the text → topics 1 and 3
  - keywords of the text: w2 and w5

… the **comedy festival** formerly known as the us **comedy arts** festival is a comedy festival held each year in **las vegas nevada** from its 1985 inception to 2008 . it was held annually at the **wheeler opera house** and other venues in **aspen colorado** . the primary sponsor of the festival was hbo with co-sponsorship by caesars palace . the primary venue tbs **geico insurance** twix candy bars and **smirnoff vodka hbo** exited the festival business in 2007 …

Intent matching via convolutional-pooling

- Semantic matching of query and document



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

More examples

Query	Title of the top-1 returned document retrieved by CLSM
warm environment arterioles do what	**thermoregulation** wikipedia the free encyclopedia
auto body repair cost **calculator** software	free online **car body shop** repair **estimates**
what happens if our body **absorbs excessive** amount **vitamin** d	**calcium** supplements and **vitamin** d discussion stop **sarcoidosis**
how do **camera** use **ultrasound focus** automatically	**wikianswers how** does a **camera focus**
how to **change font excel office 2013**	**change font** default styles in **excel 2013**
where do i get my **federal tax return transcript**	how to get **trasncripts** of **federal** income **tax returns fast** ehow
12 **fishing boats trailers**	**trailer kits** and accessories **motorcycle** utility **boat snowmobile**
acp **ariakon combat pistol** 2.0	paintball acp **combat** pistol paintball **gun paintball pistol** package deal marker and gun

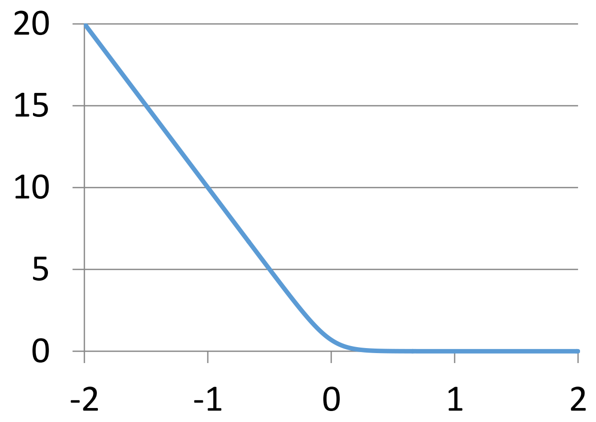
Learning DSSM from Labeled X-Y Pairs

- Consider a query X and two docs Y^+ and Y^-
  - Assume Y^+ is more relevant than Y^- with respect to X
- $\mathrm{sim}_{\boldsymbol{\theta}}(X, Y)$ is the cosine similarity of X and Y in semantic space, mapped by DSSM parameterized by $\boldsymbol{\theta}$

Learning DSSM from Labeled X-Y Pairs

- Consider a query X and two docs Y^+ and Y^-
  - Assume Y^+ is more relevant than Y^- with respect to X
- $\mathrm{sim}_{\boldsymbol{\theta}}(X, Y)$ is the cosine similarity of X and Y in semantic space, mapped by DSSM parameterized by $\boldsymbol{\theta}$

- $\Delta = \mathrm{sim}_{\boldsymbol{\theta}}(X, Y^+) - \mathrm{sim}_{\boldsymbol{\theta}}(X, Y^-)$
  - We want to maximize Δ
- $Loss(\Delta; \boldsymbol{\theta}) = \log(1 + \exp(-\gamma\Delta))$
- Optimize $\boldsymbol{\theta}$ using mini-batch SGD on GPU



Mine “labeled” X-Y pairs from search logs

how to deal with stuffy nose?	⟷	NO CLICK
stuffy nose treatment	⟷	NO CLICK
cold home remedies	⟷	http://www.agelessherbs.com/BestHomeRemediesColdFlu.html

[Gao, He, Nie, 2010]

Mine “labeled” X-Y 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 then describe the best home remedies for these specific patterns that you can use to treat the cold or influenza virus.

Cold and Flu Basics

The basic pathogenic influences are:

- **Wind**
- **Cold**
- **Heat**
- **Damp**

Wind

Theoretically, wind enters the body through the back of the neck area or nose carrying the pathogen. It first attacks the Lung system (including the sinuses) because the Lung organ system is the most external Yin organ, a thus the most vulnerable to an external invasion. External Wind invasion is marked by acute conditions with a sudden onset of symptoms.

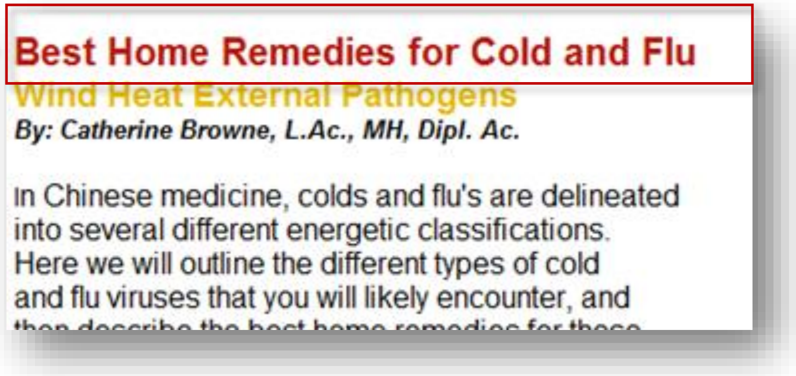
[Gao, He, Nie, 2010]

Mine “labeled” X-Y pairs from search logs

how to deal with stuffy nose?

stuffy nose treatment

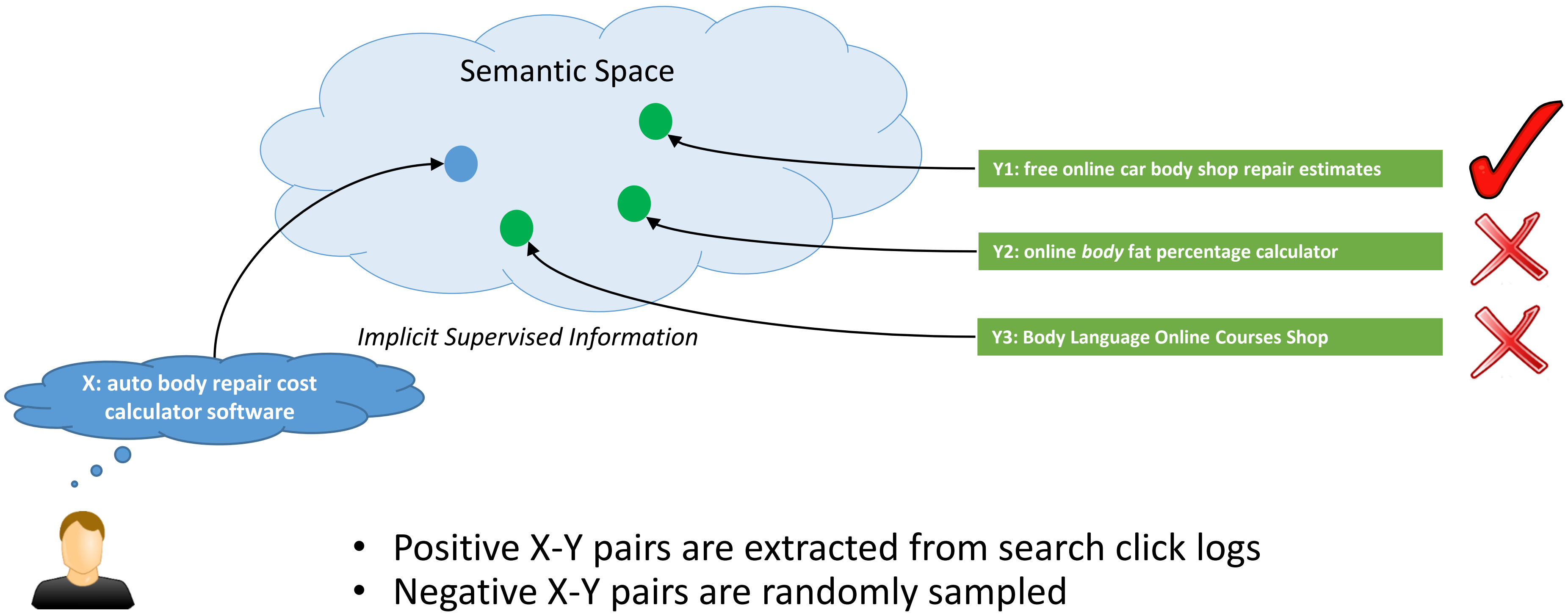
cold home remedies



QUERY (Q)	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

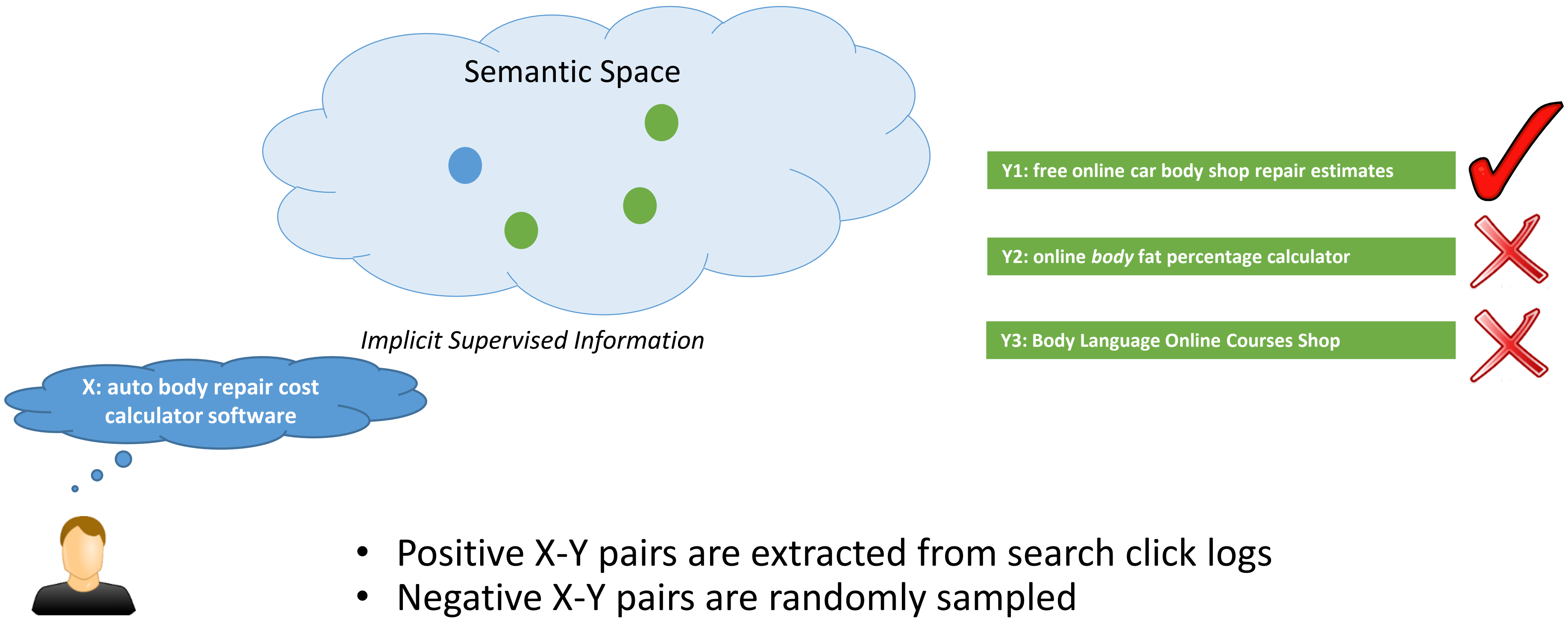
[Gao, He, Nie, 2010]

Learning DSSM from Labeled X-Y Pairs



- Positive X-Y pairs are extracted from search click logs
- Negative X-Y pairs are randomly sampled
- Map X and Y into the same semantic space via deep neural net

Learning DSSM from Labeled X-Y Pairs

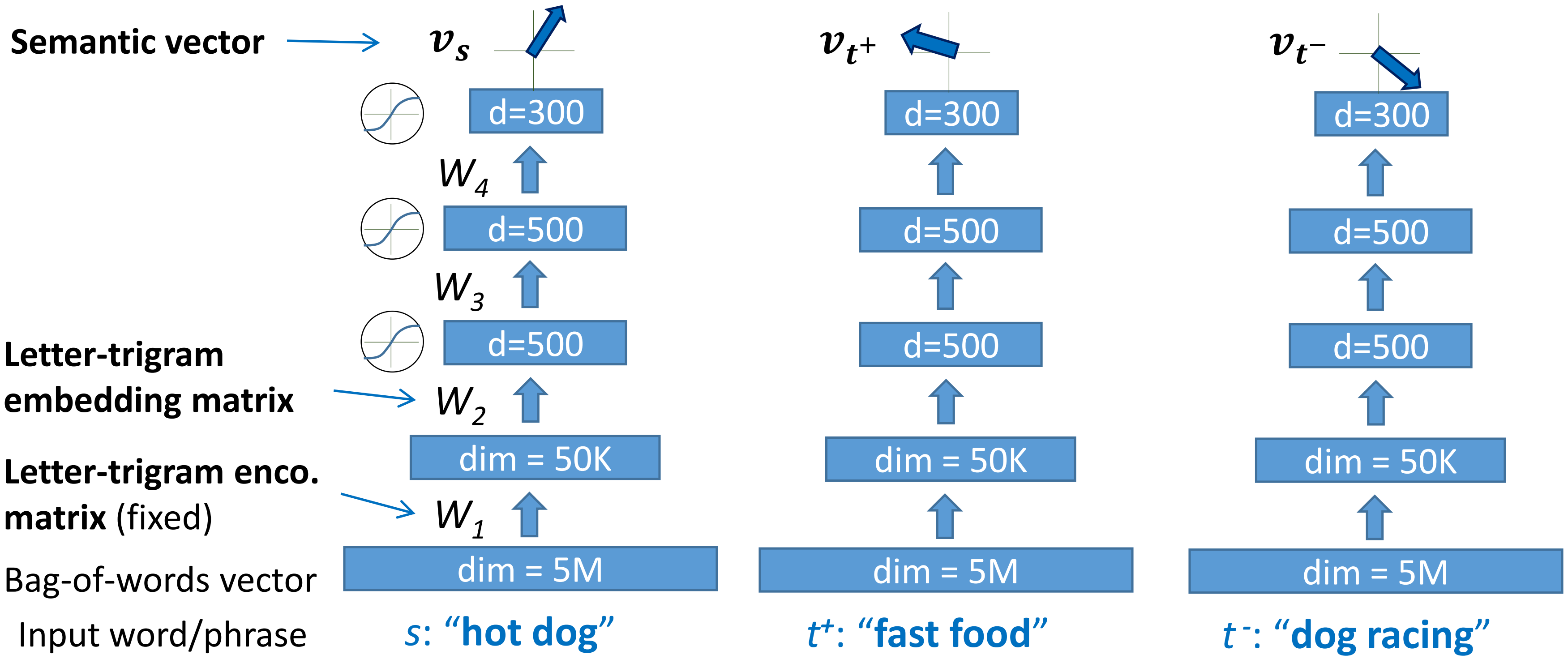


- Positive X-Y pairs are extracted from search click logs
- Negative X-Y pairs are randomly sampled
- Map X and Y into the same semantic space via deep neural net
- Positive Y are closer to X than negative Y in that space

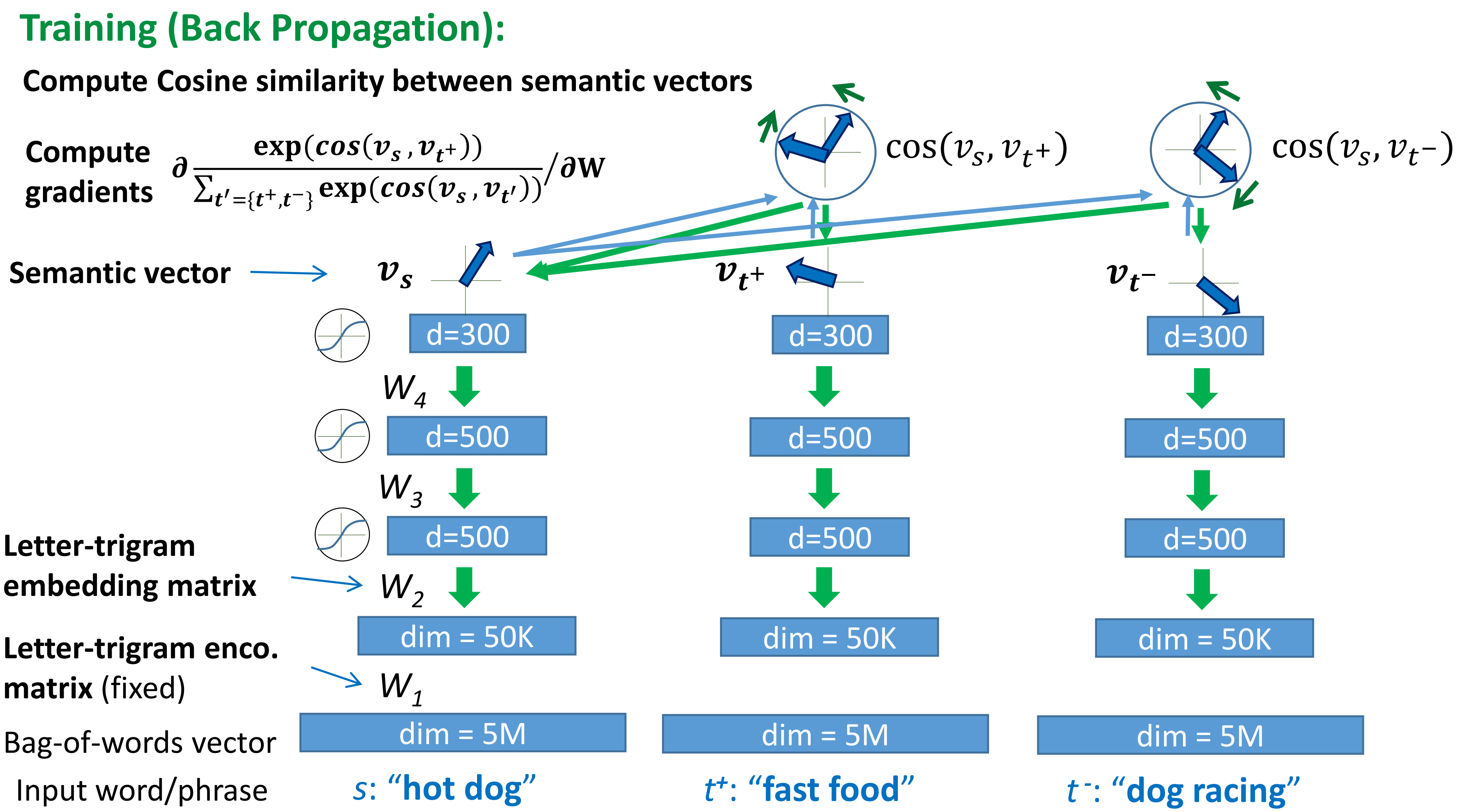
Learning DSSM on X-Y pairs via SGD

Initialization:

Neural networks are initialized with random weights

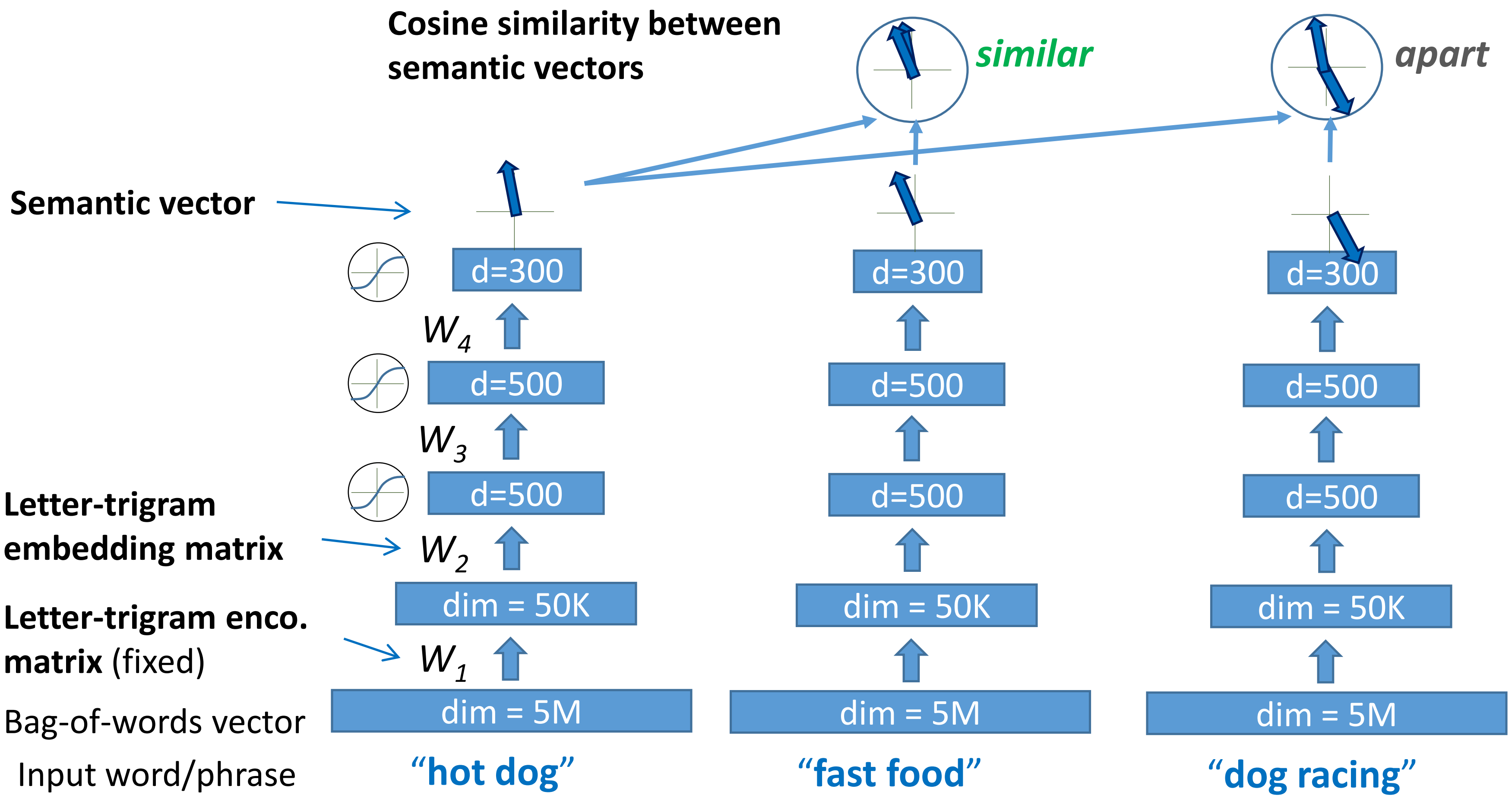


Learning DSSM on X-Y pairs via SGD



Learning DSSM on X-Y pairs via SGD

After training converged:



Evaluation Methodology

- Measurement: NDCG, t-test
- Test set:
  - 12,071 English queries sampled from 1-y log
  - 5-level relevance label for each query-doc pair
- Training data for translation models:
  - 82,834,648 query-title pairs
- Baselines
  - Lexicon matching models: BM25, ULM
  - Translation models
  - Topic models
  - Deep auto-encoder [Hinton & Salakhutdinov 2010]

Translation models for web search

D: best home remedies for cold and flu

Q: how to deal with stuffy nose

- Leverage statistical machine translation (SMT) technologies and infrastructures to improve search relevance
- Model documents and queries as different languages, cast mapping queries to documents as bridging the language gap via translation
- Given a Q, D can be ranked by how likely it is that Q is “translated” from D, $P(\mathrm{Q}|\mathrm{D})$
  - Word translation model
  - Phrase translation model

[Gao, He, Nie, 2010]

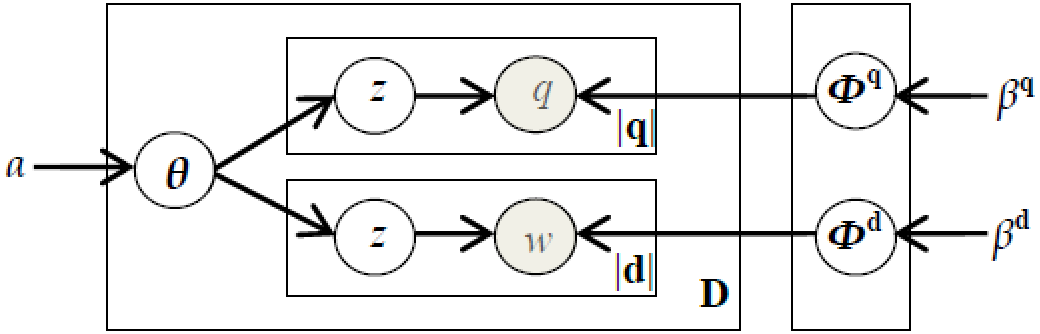
Generative Topic Models

Q: stuffy nose treatment ← D: cold home remedies

Q: stuffy nose treatment ← Topic ← D: cold home remedies

- Probabilistic latent Semantic Analysis (PLSA)
  - $P(\mathrm{Q}|\mathrm{D}) = \prod_{q \in \mathrm{Q}} \sum_{z} P(q|\boldsymbol{\phi}_z) P(z|\mathrm{D}, \boldsymbol{\theta})$
  - D is assigned a single most likely topic vector
  - Q is generated from the topic vectors
- Latent Dirichlet Allocation (LDA) generalizes PLSA
  - a posterior distribution over topic vectors is used
  - PLSA = LDA with MAP inference

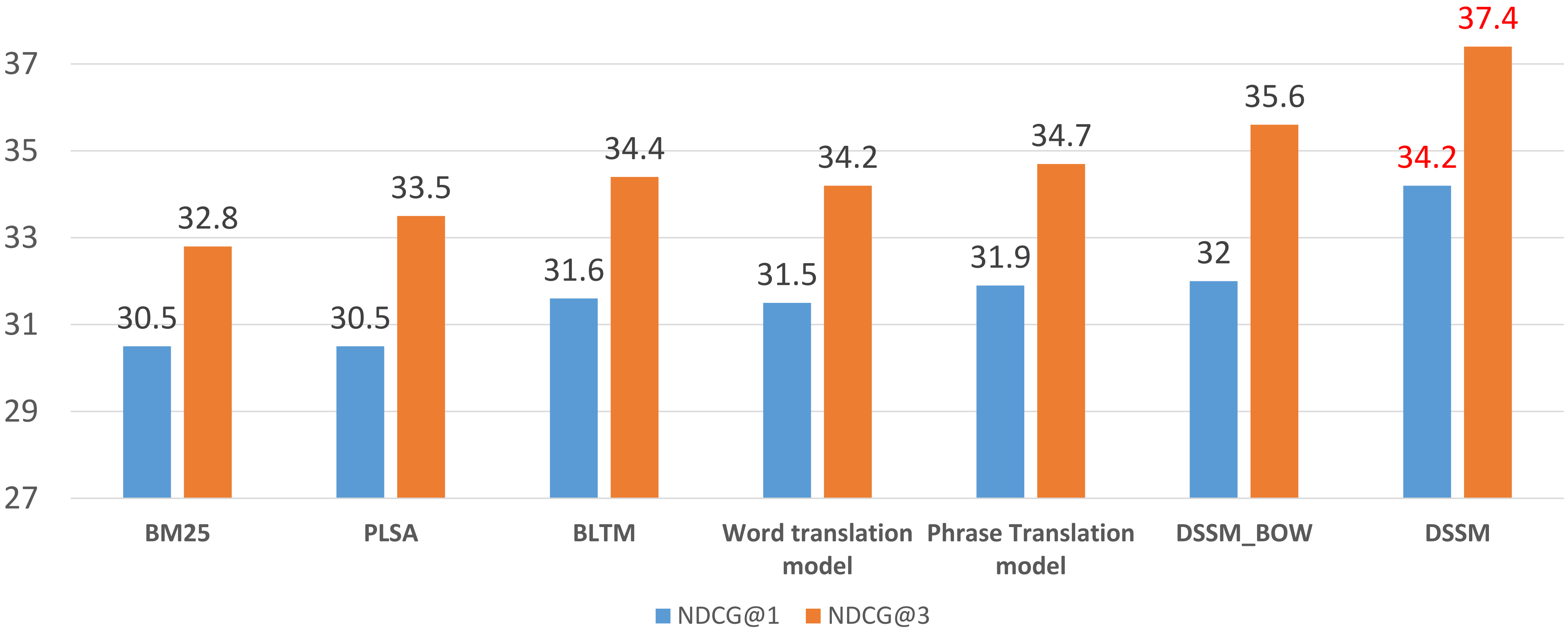
Bilingual topic model for web search



- For each topic *z*: $\left(\boldsymbol{\phi}_z^{\mathrm{Q}}, \boldsymbol{\phi}_z^{\mathrm{D}}\right) \sim \mathrm{Dir}(\boldsymbol{\beta})$
- For each Q-D pair: $\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})$
- Each *q* is generated by $z \sim \boldsymbol{\theta}$ and $q \sim \boldsymbol{\phi}_z^{\mathrm{Q}}$
- Each *w* is generated by $z \sim \boldsymbol{\theta}$ and $w \sim \boldsymbol{\phi}_z^{\mathrm{D}}$

[Gao, Toutanova, Yih, 2011]

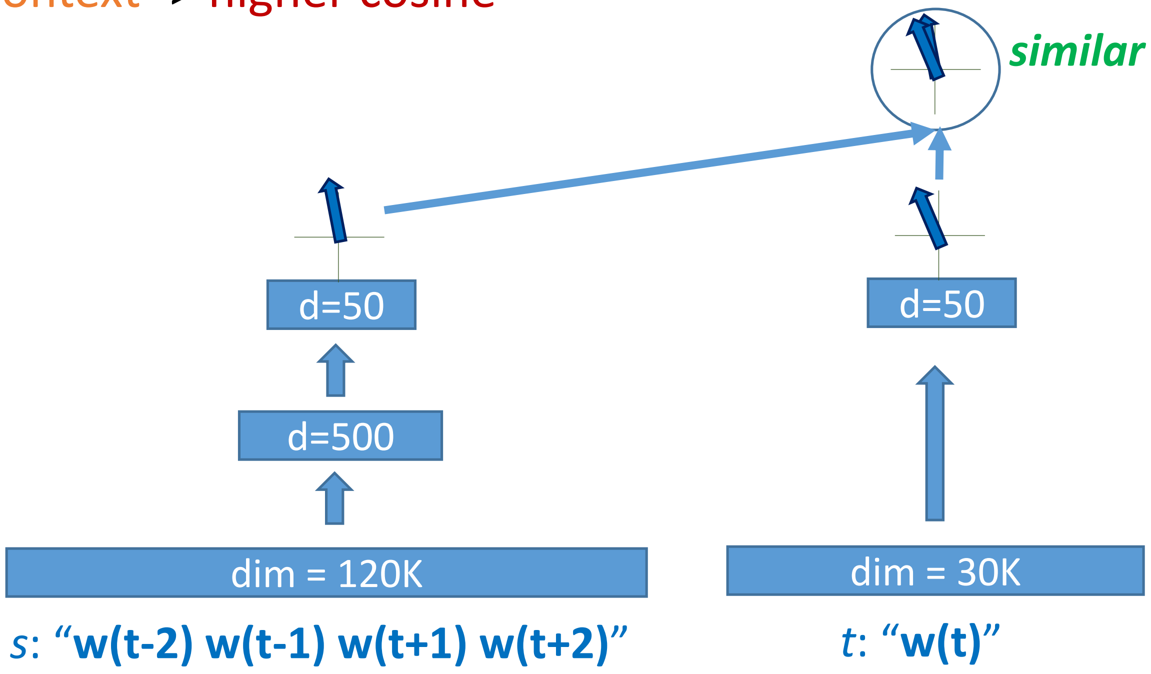
Web doc ranking results



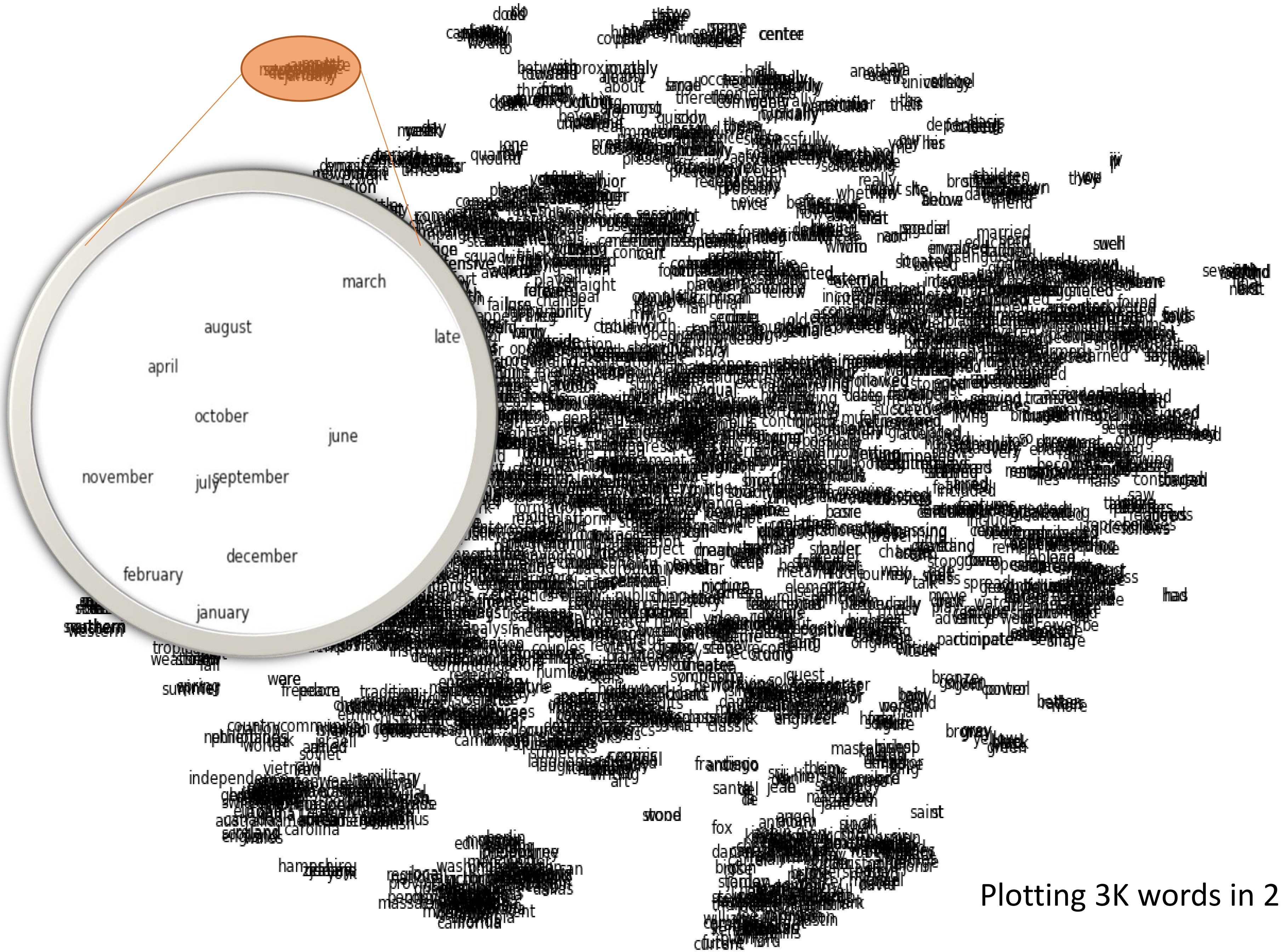
Analysis: DSSM for semantic word clustering and analogy

- Learn word embedding by means of its neighbors (context)
  - Construct context <-> word training pair for DSSM
  - Similar words with similar context -> higher cosine

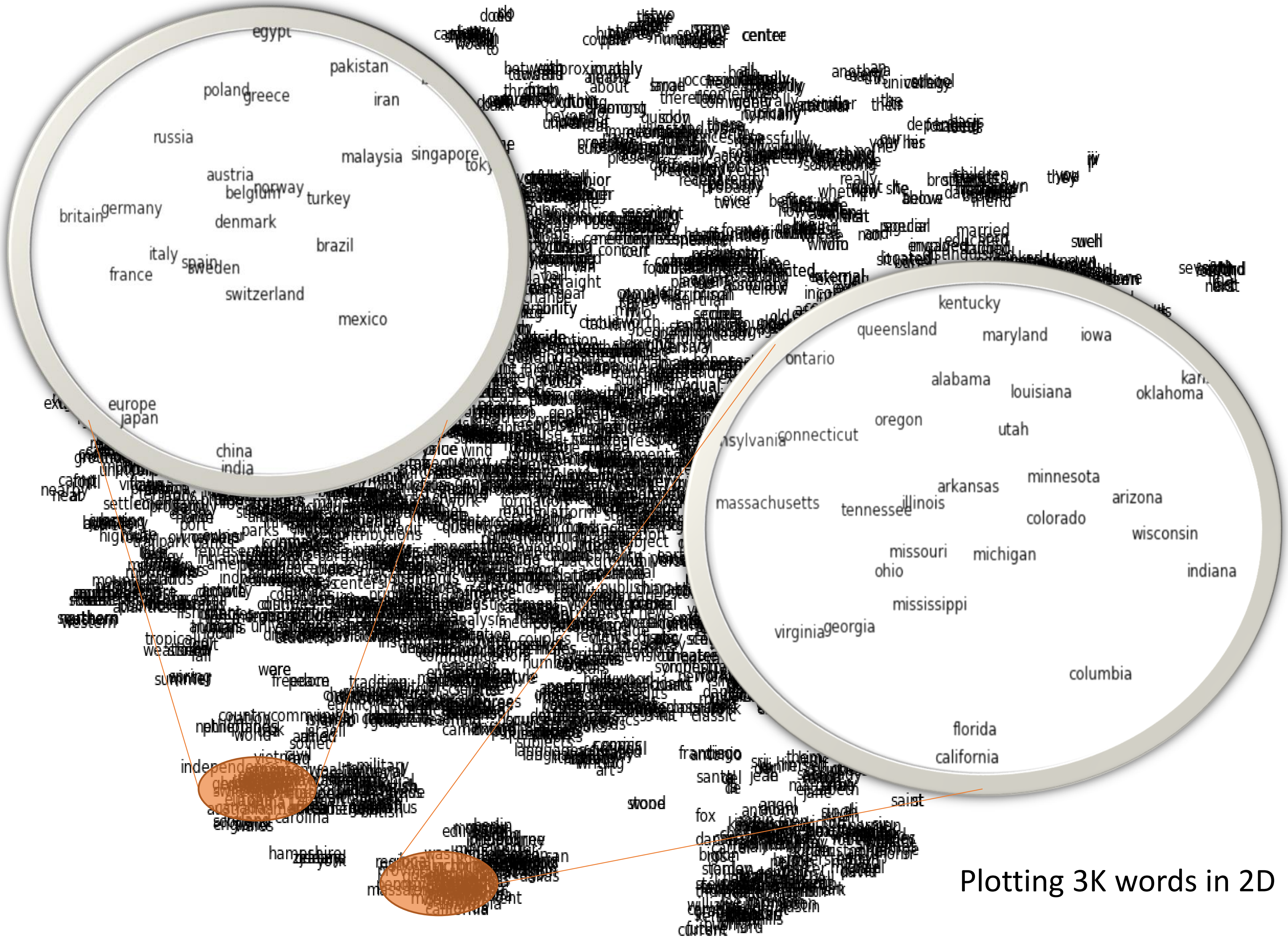
- Training setting:
  - 30K vocabulary size
  - 10M words from Wikipedia
  - 50-dimentional vector



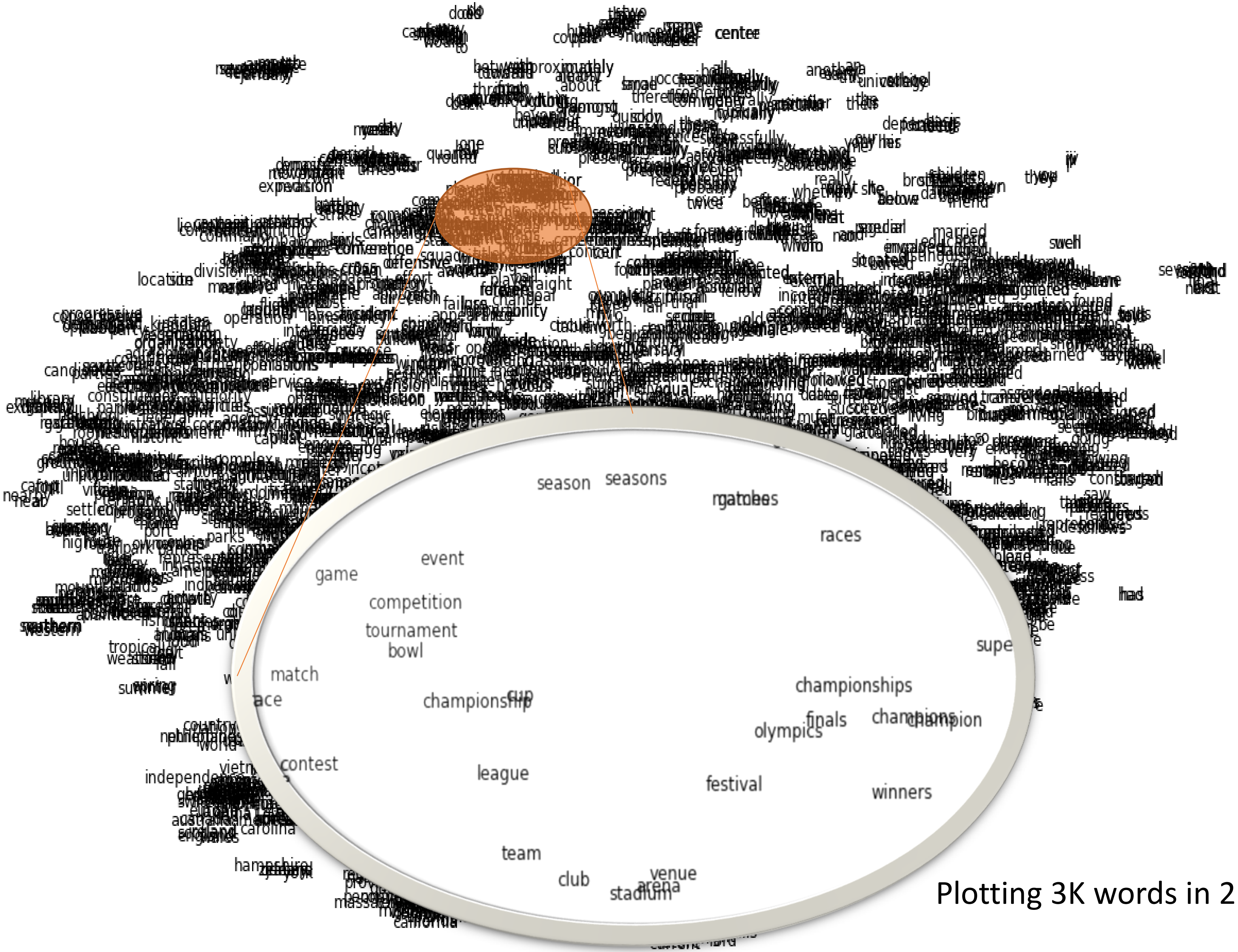
[Song et al. 2014]



Plotting 3K words in 2D



Plotting 3K words in 2D



Plotting 3K words in 2D

DSSM: semantic similarity vs. semantic reasoning

Semantic clustering examples (how similar words are)
Top 3 neighbors of each word

king	earl (0.77)	pope (0.77)	lord (0.74)
woman	person (0.79)	girl (0.77)	man (0.76)
france	spain (0.94)	italy (0.93)	belgium (0.88)
rome	constantinople (0.81)	paris (0.79)	moscow (0.77)
winter	summer (0.83)	autumn (0.79)	spring (0.74)

Semantic reasoning examples (how words relate to one another)

$$w_1 : w_2 = w_3 : x \quad \Rightarrow \quad V_x = V_3 - V_1 + V_2$$

summer : rain = winter : x
italy : rome = france : x
man : eye = car : x
man : woman = king : x
read : book = listen : x

*Note that the DSSM used in these examples are trained in an unsupervised manner, as Google's word2vec.

DSSM: semantic similarity vs. semantic reasoning

Semantic clustering examples (how similar words are)
Top 3 neighbors of each word

king	earl (0.77)	pope (0.77)	lord (0.74)
woman	person (0.79)	girl (0.77)	man (0.76)
france	spain (0.94)	italy (0.93)	belgium (0.88)
rome	constantinople (0.81)	paris (0.79)	moscow (0.77)
winter	summer (0.83)	autumn (0.79)	spring (0.74)

Semantic reasoning examples (how words relate to one another)

$$w_1 : w_2 = w_3 : x \quad \Rightarrow \quad V_x = V_3 - V_1 + V_2$$

summer : rain = winter : x	snow (0.79)	rainfall (0.73)	wet (0.71)
italy : rome = france : x	paris (0.78)	constantinople (0.74)	egypt (0.73)
man : eye = car : x	motor (0.64)	brake (0.58)	overhead (0.58)
man : woman = king : x	mary (0.70)	prince (0.70)	queen (0.68)
read : book = listen : x	sequel (0.65)	tale (0.63)	song (0.60)

*Note that the DSSM used in these examples are trained in an unsupervised manner, as Google's word2vec.

Summary

- Map the queries and documents into the same latent semantic space
- Doc ranking score is the cosine distance of Q/D vectors in that space
- DSSM outperforms all the competing models
- The learning DSSM vectors capture semantic similarities and relations btw words

DSSM for recommendation

- Two interestingness tasks for recommendation
- Modeling interestingness via DSSM
- Training data acquisition
- Evaluation
- Summary

[Gao et al. 2014b]

Two Tasks of Modeling Interestingness

- Automatic highlighting
  - Highlight the key phrases which represent the entities (person/loc/org) that interest a user when reading a document
  - Doc semantics influences what is perceived as interesting to the user
  - e.g., article about movie → articles about an actor/character

- Contextual entity search
  - Given the highlighted key phrases, recommend new, interesting documents by searching the Web for supplementary information about the entities
  - A key phrase may refer to different entities; need to use the contextual information to disambiguate

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

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

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.

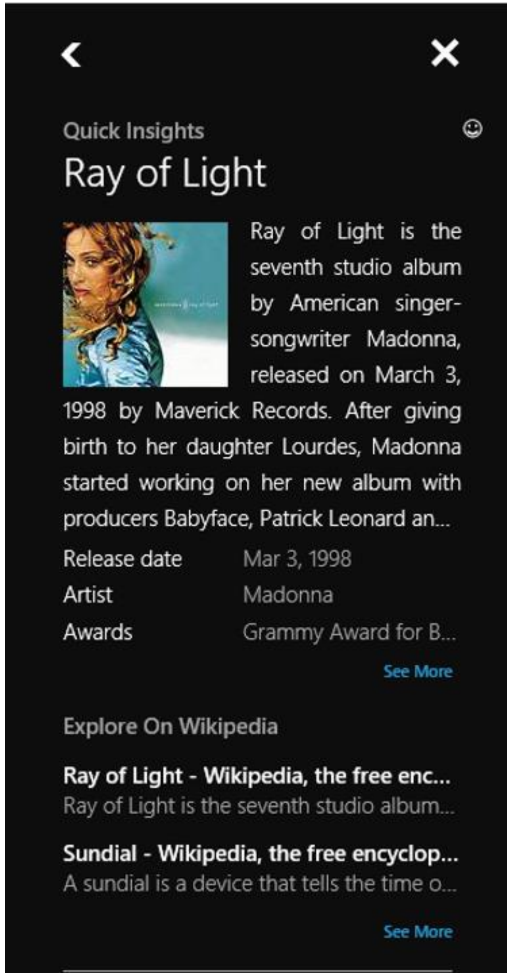
(2) Modern physicists were willing to suppose that light might be subject to gravitation—i.e., that a ray of light passing near a great mass like the sun might be deflected to the extent to which a particle moving with the same velocity would be deflected according to the orthodox theory of gravitation. 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

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

Entity



The Einstein Theory of Relativity

Context

(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

Quick Insights

Ray of Light

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

Explore On Wikipedia

Ray of Light - Wikipedia, the free enc...
Ray of Light is the seventh studio album...

Sundial - Wikipedia, the free encyclop...
A sundial is a device that tells the time o...

See More

The Einstein Theory of Relativity

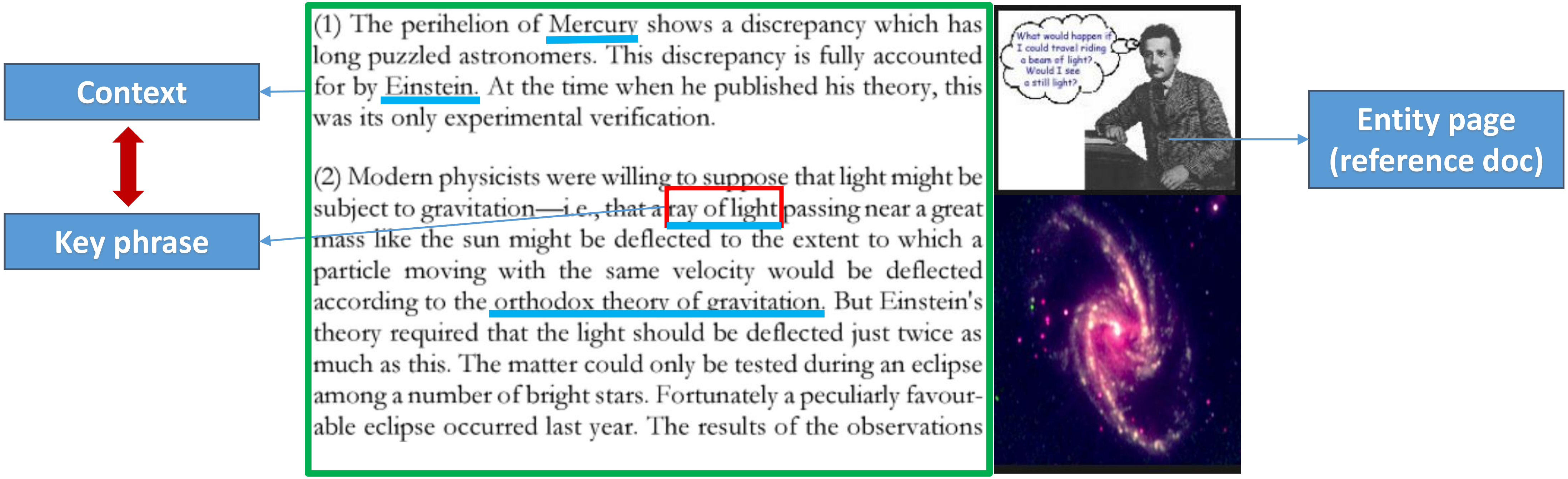
Context

(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

DSSM for Modeling Interestingness



Tasks	X (source text)	Y (target text)
Automatic highlighting	***Doc in reading***	***Key phrases to be highlighted***
Contextual entity search	*Key phrase and context*	*Entity and its corresponding (wiki) page*

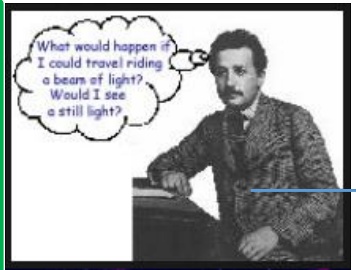
DSSM for Modeling Interestingness

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.

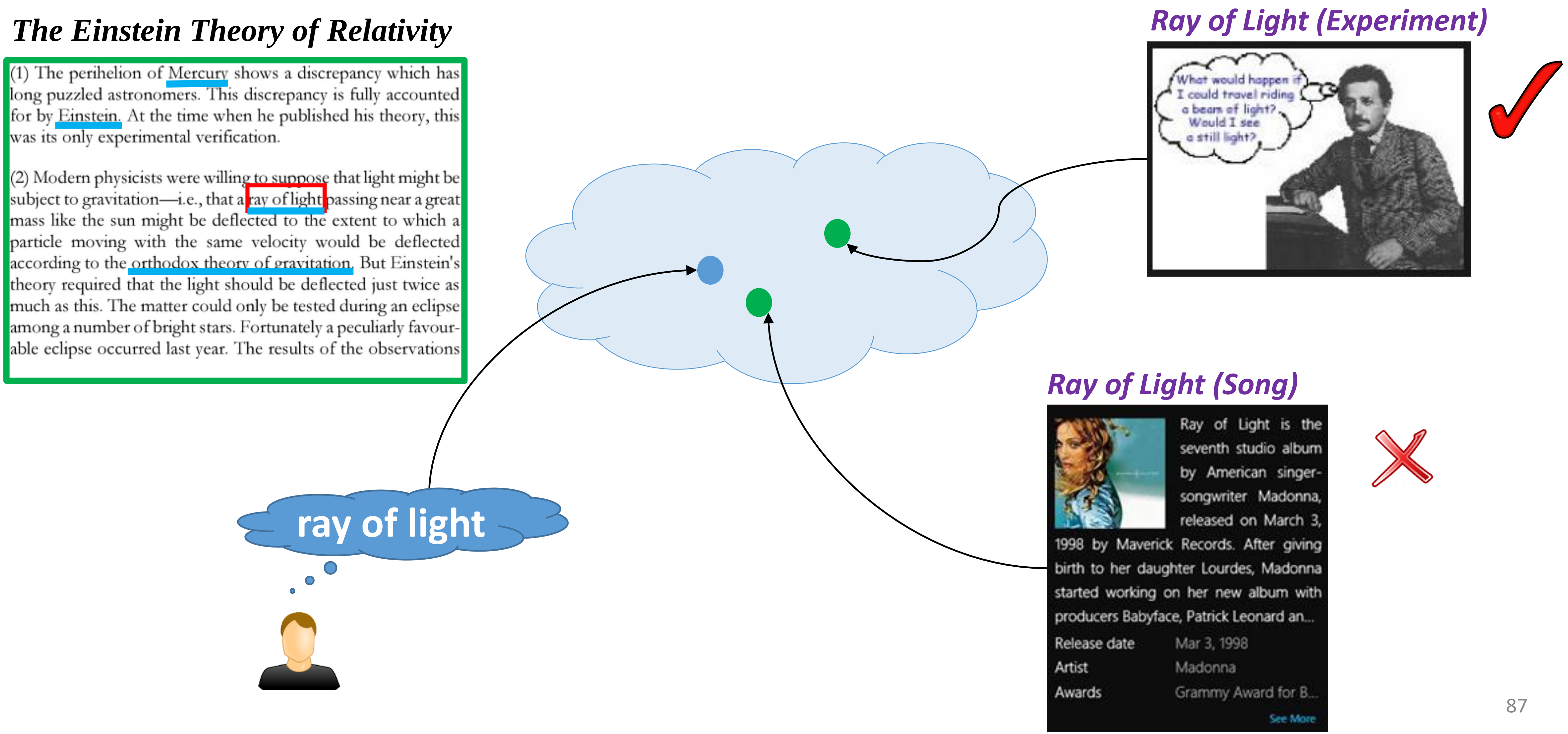
(2) Modern physicists were willing to suppose that light might be subject to gravitation—i.e., that a ray of light passing near a great mass like the sun might be deflected to the extent to which a particle moving with the same velocity would be deflected according to the orthodox theory of gravitation. 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



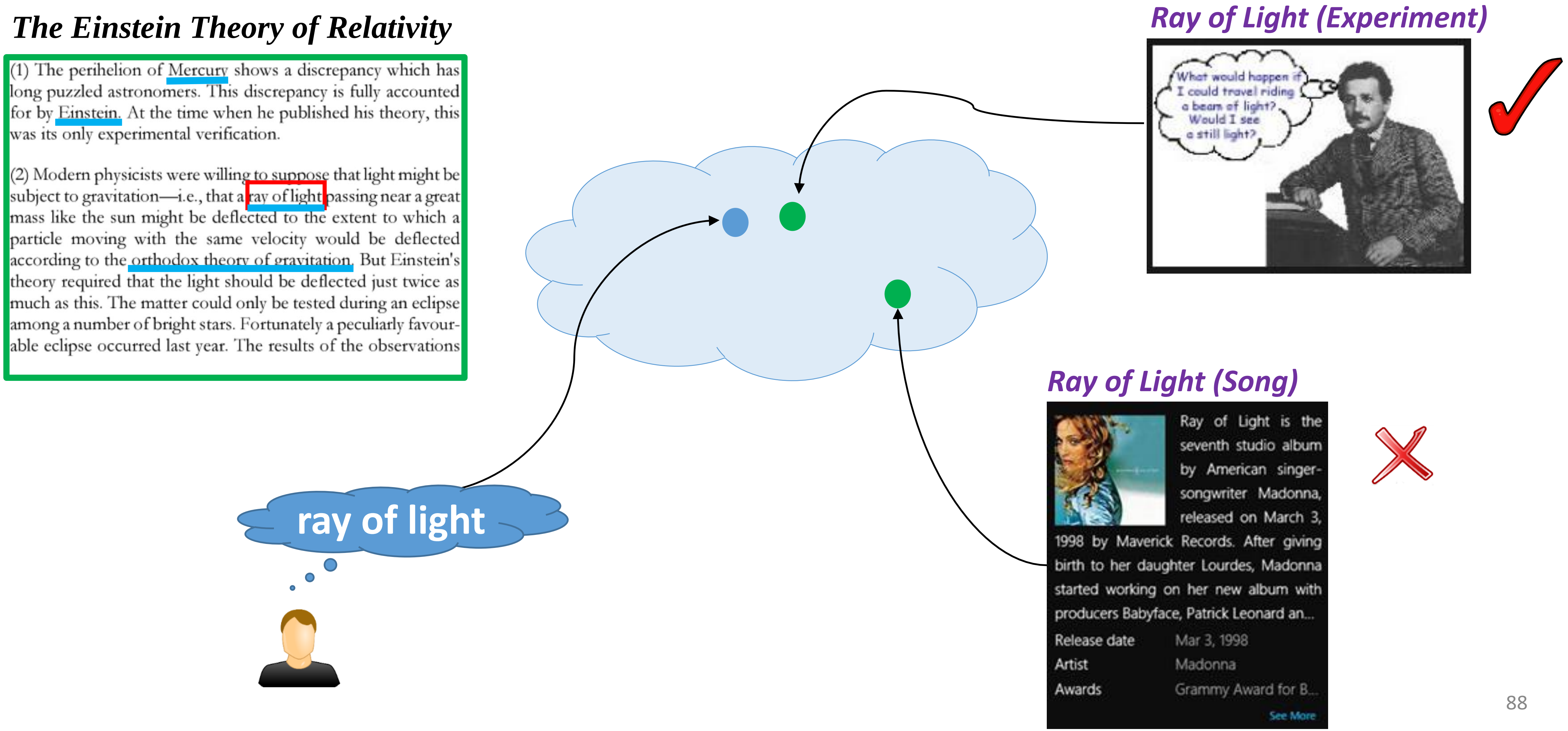
Entity page (reference doc)

Tasks	*X (source text)*	*Y (target text)*
Automatic highlighting	*Doc in reading*	*Key phrases to be highlighted*
Contextual entity search	***Key phrase and context***	***Entity and its corresponding (wiki) page***

Learning DSSM from Labeled X-Y Pairs



Learning DSSM from Labeled X-Y Pairs



DSSM for recommendation

- Two interestingness tasks for recommendation
- Modeling interestingness via DSSM
- Training data acquisition
- Evaluation
- Summary

Extract Labeled Pairs from Web Browsing Logs
Automatic Highlighting

- When reading a page P, the user *clicks* a hyperlink H

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.

…

H

- (text in P, anchor text of H)

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)

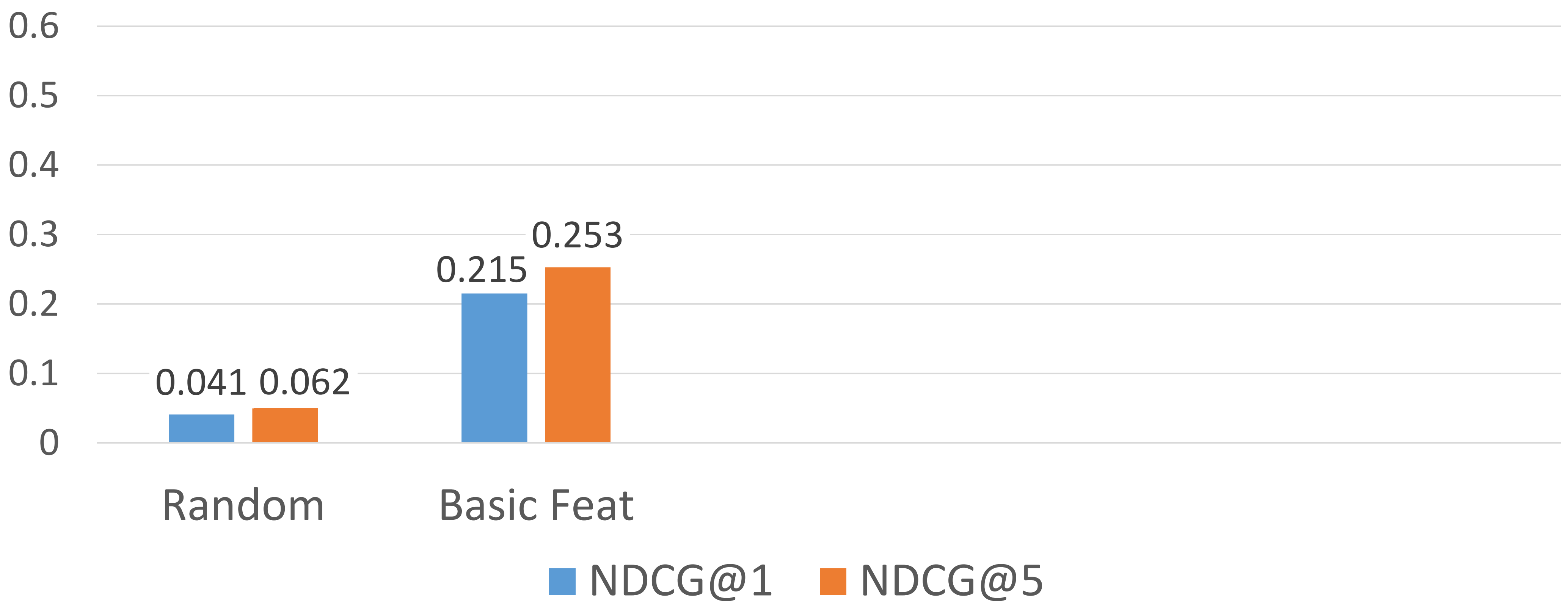
- (anchor text of H & surrounding words, text in P')

Automatic Highlighting: Settings

- Simulation
  - Use a set of anchors as candidate key phrases to be highlighted
  - Gold standard rank of key phrases – determined by # user clicks
  - Model picks top-k keywords from the candidates
  - Evaluation metric: NDCG

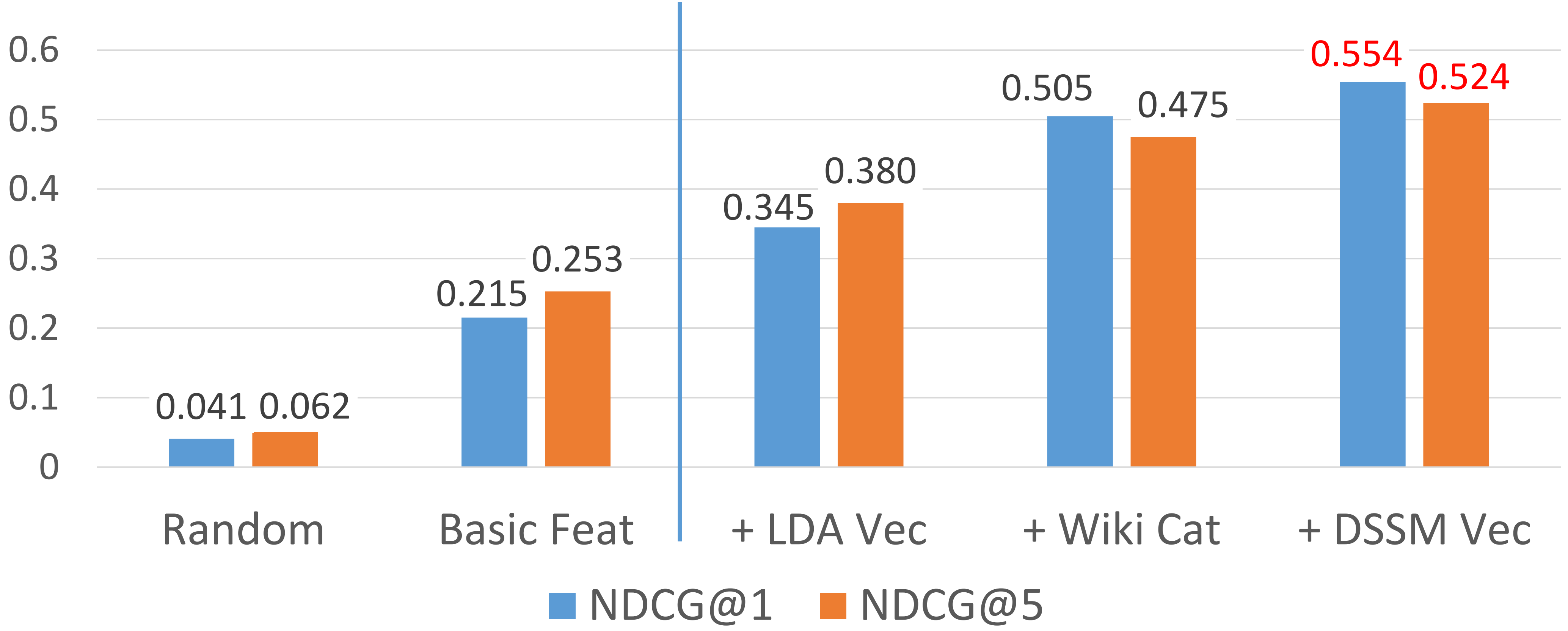
- Data
  - 18 million occurrences of user clicks from a Wiki page to another, collected from 1-year Web browsing logs
  - 60/20/20 split for training/validation/evaluation

Automatic Highlighting Results: Baselines



- Random: Random baseline
- Basic Feat: Boosted decision tree learner with document features, such as anchor position, freq. of anchor, anchor density, etc.

Automatic Highlighting Results: Semantic Features

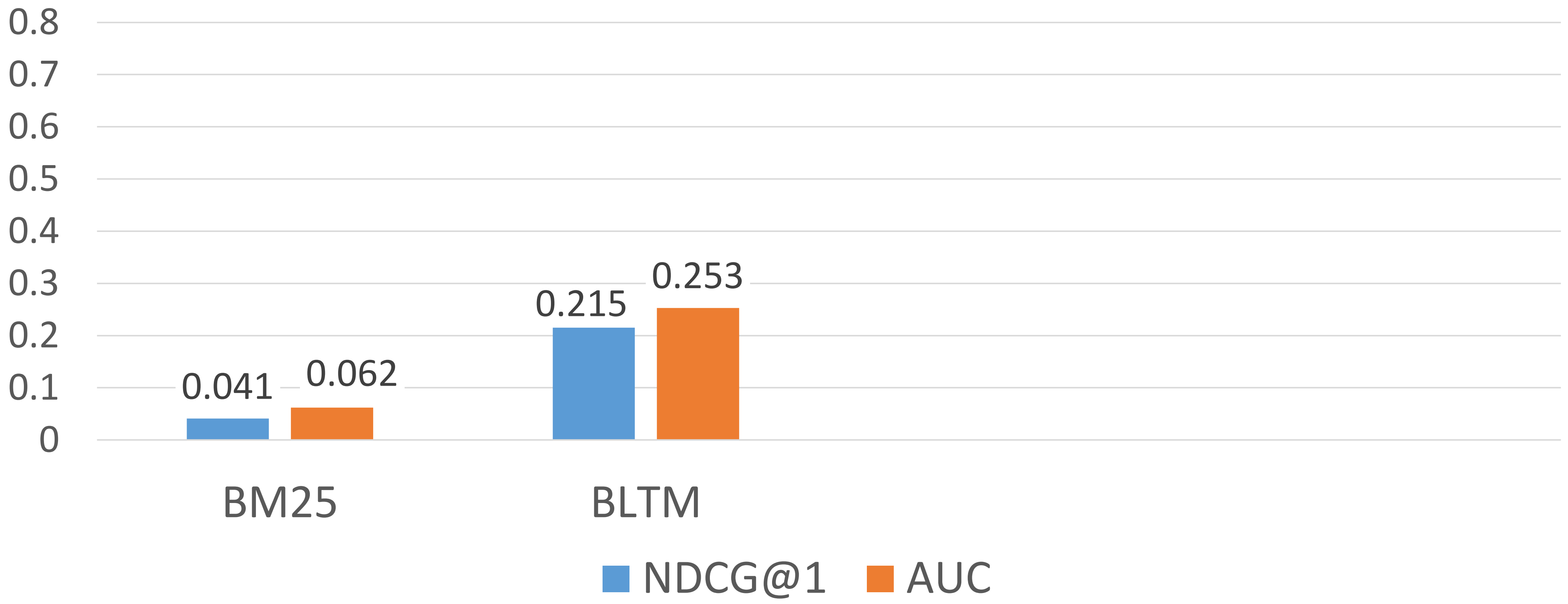


- + LDA Vec: Basic + Topic model (LDA) vectors [Gamon+ 2013]
- + Wiki Cat: Basic + Wikipedia categories (do not apply to general documents)
- + DSSM Vec: Basic + DSSM vectors

Contextual Entity Search: Settings

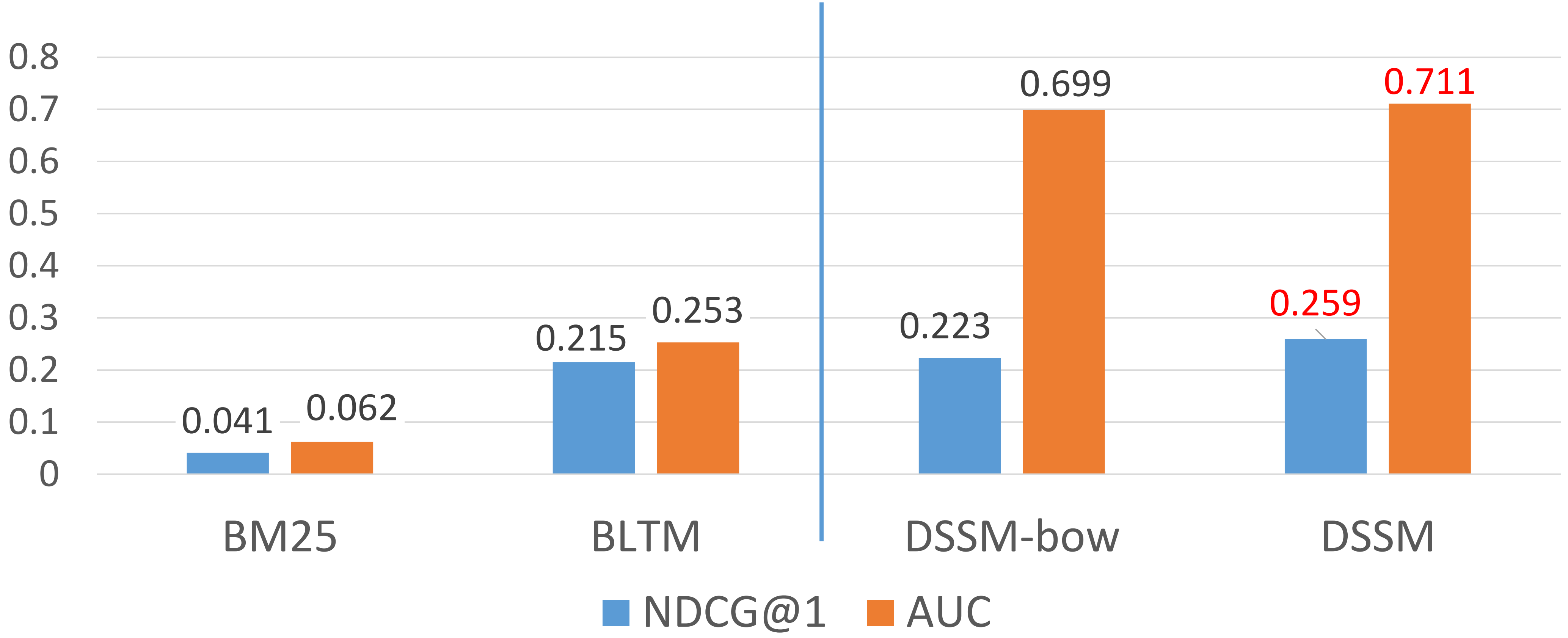
- Training/validation data: same as in *automatic highlighting*
- Evaluation data
  - Sample 10k Web documents as the source documents
  - Use named entities in the doc as query; retain up to 100 returned documents as target documents
  - Manually label whether each target document is a good page describing 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

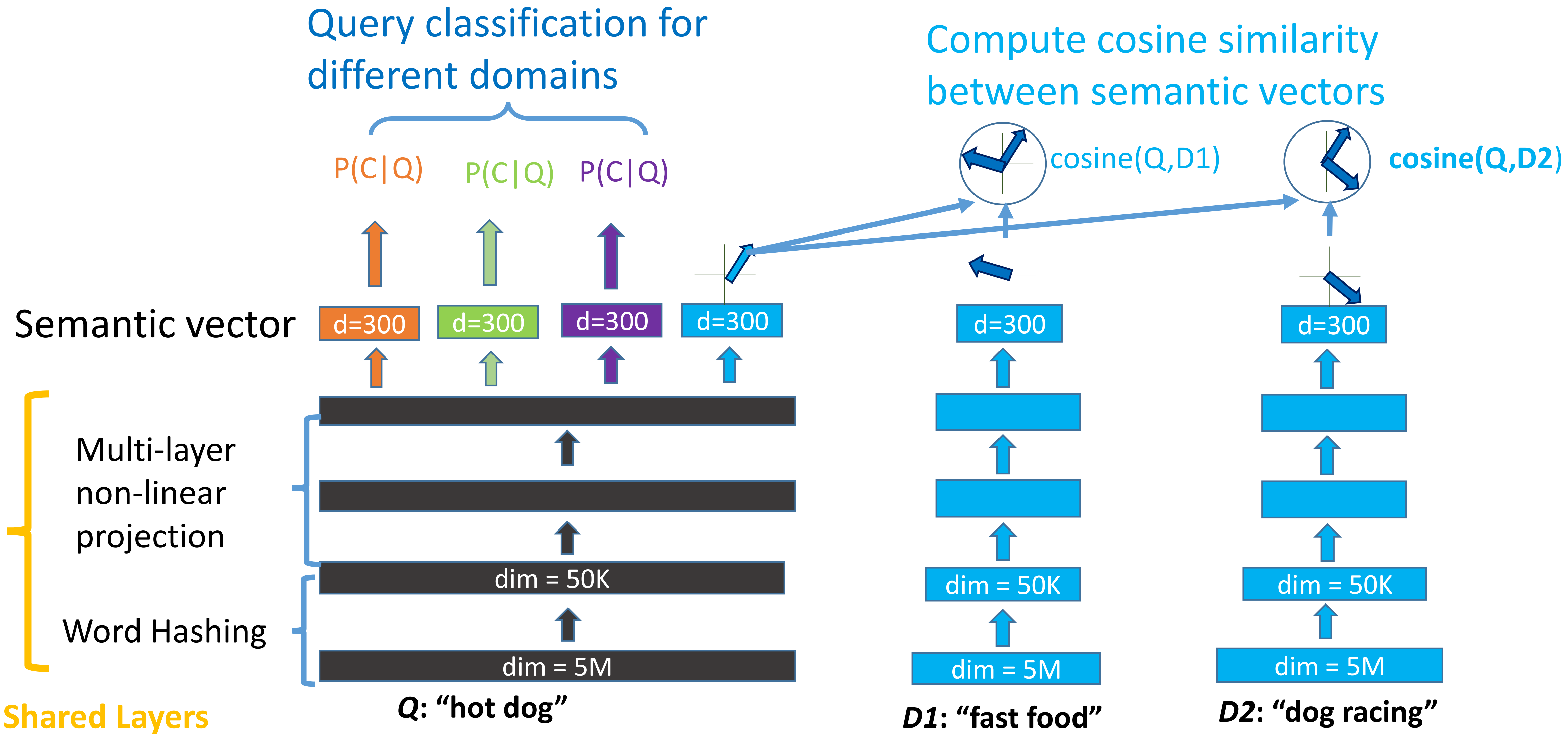


- DSSM-bow: DSSM without convolutional layer and max-pooling structure
- DSSM outperforms classic doc model and state-of-the-art topic model

Summary

- Extract labeled pairs from Web browsing logs
- DSSM outperforms state-of-the-art topic models
- DSSM learned semantic features outperform the thousands of features coming from the manually assigned semantic labels

Multi-task DSSM for scalable intent modeling



Deep Semantic Similarity Model (DSSM): learning semantic similarity between *X* and *Y*

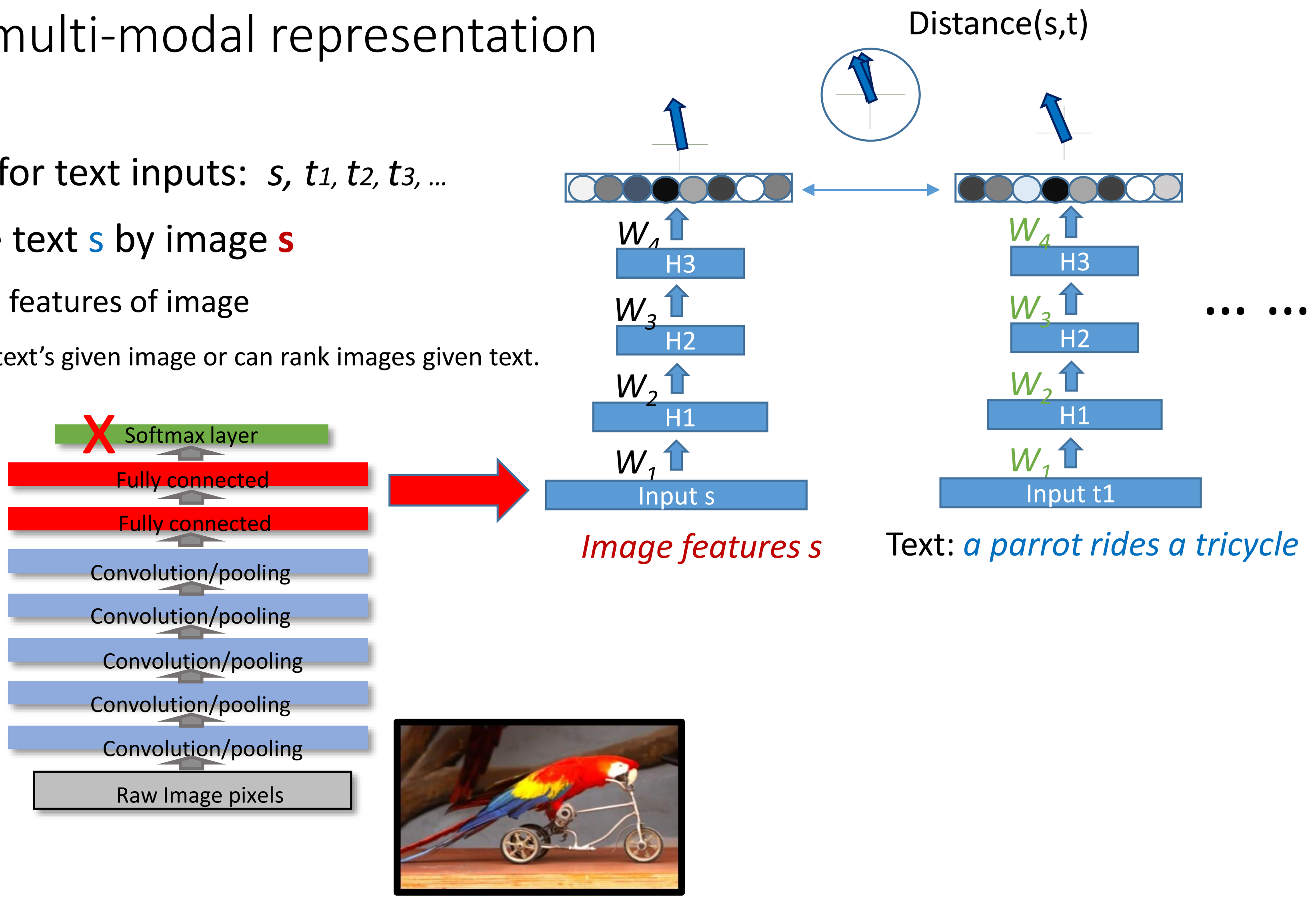
Tasks	*X*	*Y*
Web search	*Search query*	*Web documents*
Ad selection	*Search query*	*Ad keywords*
Entity ranking	*Mention (highlighted)*	*Entities*
Recommendation	*Doc in reading*	*Interesting things in doc or other docs*
Machine translation	*Sentence in language A*	*Translations in language B*
Nature User Interface	*Command (text/speech)*	*Action*
Summarization	*Document*	*Summary*
Query rewriting	*Query*	*Rewrite*
Image captioning	***Text string***	***Images***
…	…	…

[Huang et al. 2013; Shen et al. 2014; Gao et al. 2014a; Gao et al. 2014b]

Go beyond text

DSSM for multi-modal representation learning

- Recall DSSM for text inputs: *s*, t_1, t_2, t_3, …
- Now: replace text s by image **s**
- Using DNN/CNN features of image
- Can rank/generate text's given image or can rank images given text.



SIP: Automatic image captioning at a human-level of performance

Detector Models, Deep Neural Net Features

Computer Vision System

stree t
signs
unde r
on
light
pole
red
stop
sign
building
bus
city
traffi c

Language Model

Caption Generation System

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

DSSM Model

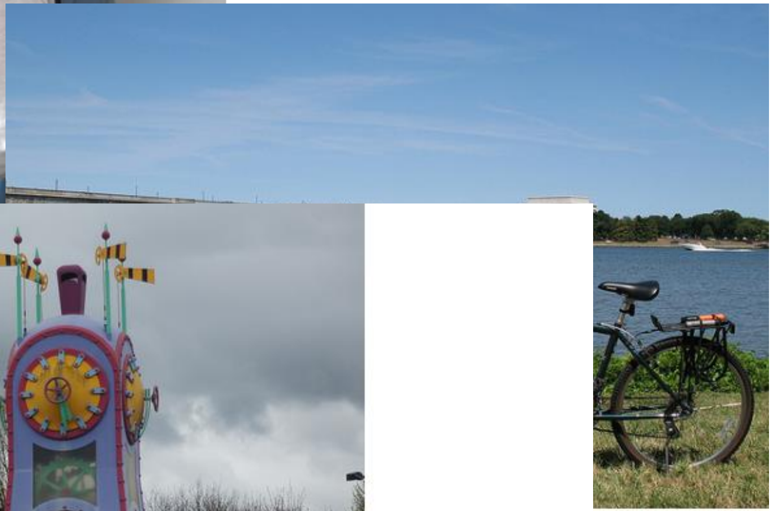
Semantic Ranking System

a stop sign at an intersection on a city street

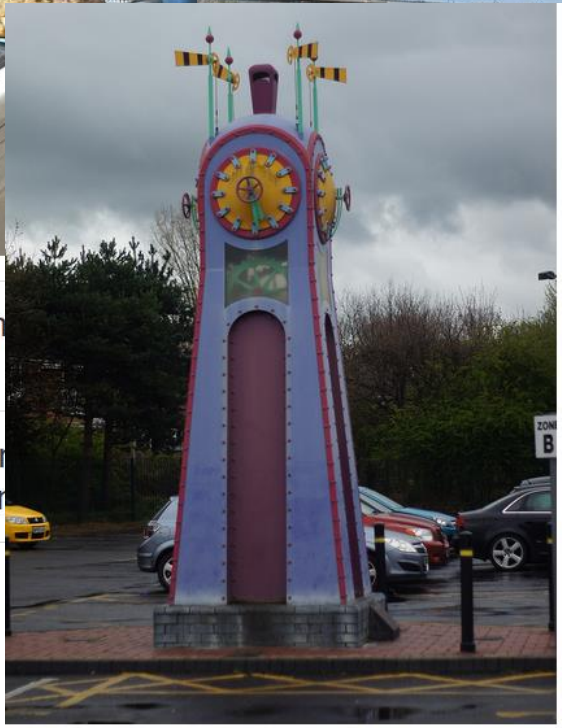
Fang, Gupta, Iandola, Srivastava, Deng, Dollar, Gao, He, Mitchell, Platt, Zitnick, Zweig, “Automatic image captioning at a human-level of performance” to appear

Machine-generated (but turker prefered)	a group of motorcycles parked next to a motorcycle
Human-annotated (but turker not prefered)	two girls wearing are wearing short skirts and one of them sits on a motorcycle while the other stands nearby

Machine-generated (but turker prefered)	a wom food
Human-annotated (but turker not prefered)	womar kitcher



next to a
next to a



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

Outline

- The basics
- Deep Semantic Similarity Models (DSSM) for text processing
- Recurrent Neural Networks (RNN)
  - N-gram language models
  - RNN language models
  - Potentials and difficulties of RNN

Statistical language modeling

- Goal: how to incorporate *language structure* into a probabilistic model
- Task: next word prediction
  - Fill in the blank: “*The dog of our neighbor* ___”
- Starting point: word *n*-gram model
  - Very simple, yet surprisingly effective
  - Words are generated from left-to-right
  - Assumes no other structure than words themselves

Word-based n-gram model

- Using **chain rule** on its *history* i.e., preceding words

$$
\begin{aligned}
P(\textit{the dog of our neighbor barks}) = {} & P(the|\langle \mathrm{BOS} \rangle) \\
& \times P(dog|\langle \mathrm{BOS} \rangle, the) \\
& \times P(of|\langle \mathrm{BOS} \rangle, the, dog) \\
& \dots \dots \\
& \times P(barks|\langle \mathrm{BOS} \rangle, the, dog, of, our, neighbor) \\
& \times P(\langle \mathrm{EOS} \rangle|\langle \mathrm{BOS} \rangle, the, dog, of, our, neighbor, barks)
\end{aligned}
$$

$$
\begin{aligned}
P(w_1 w_2 \dots w_n) &= P(w_1)P(w_2|w_1)P(w_3|w_1 w_2) \dots \\
&= P(w_1) \prod_{i=2\dots n} P(w_i|w_1 \dots w_{i-1})
\end{aligned}
$$

Word-based n-gram model

- How do we get n-gram probability estimates?
  - Get text and count: $P(w_2|w_1) = Cnt(w_1 w_2)/Cnt(w_1)$
  - Smoothing to ensure non-zero probabilities
- Problem of using long history
  - Rare events: unreliable probability estimates
  - Assuming a vocabulary of 20,000 words,

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}
fourgram $P(w_4/w_1w_2w_3)$	1.6×10^{17}

From Manning and Schütze 1999: 194

Word-based n-gram model

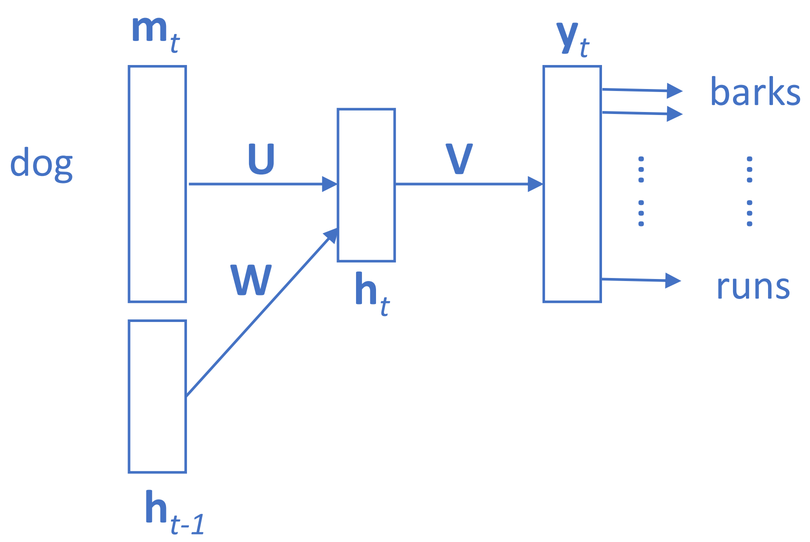
- Markov independence assumption
  - A word depends only on *n-1* preceding words, e.g.,
- Word-based tri-gram model

$$P(w_1 w_2 \dots w_n) = P(w_1)P(w_2|w_1)P(w_3|w_2)\dots$$
$$= P(w_1)\prod_{i=2\dots n} P(w_i|w_{i-2}w_{i-1})$$

- Cannot capture any long-distance dependency

the dog of our neighbor barks

Recurrent Neural Network 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

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

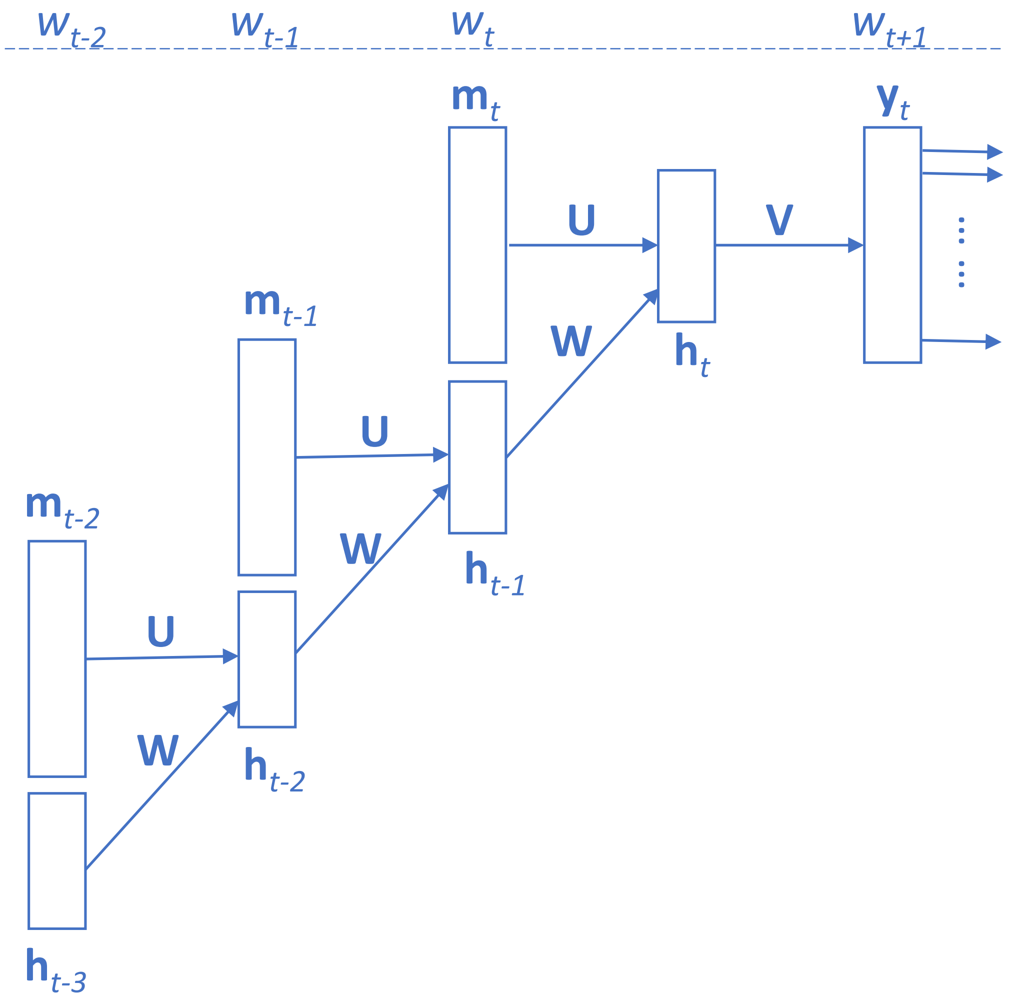
$g(.)$ is called the *softmax* function

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 et al., 2011]

RNN unfolds into a DNN over time

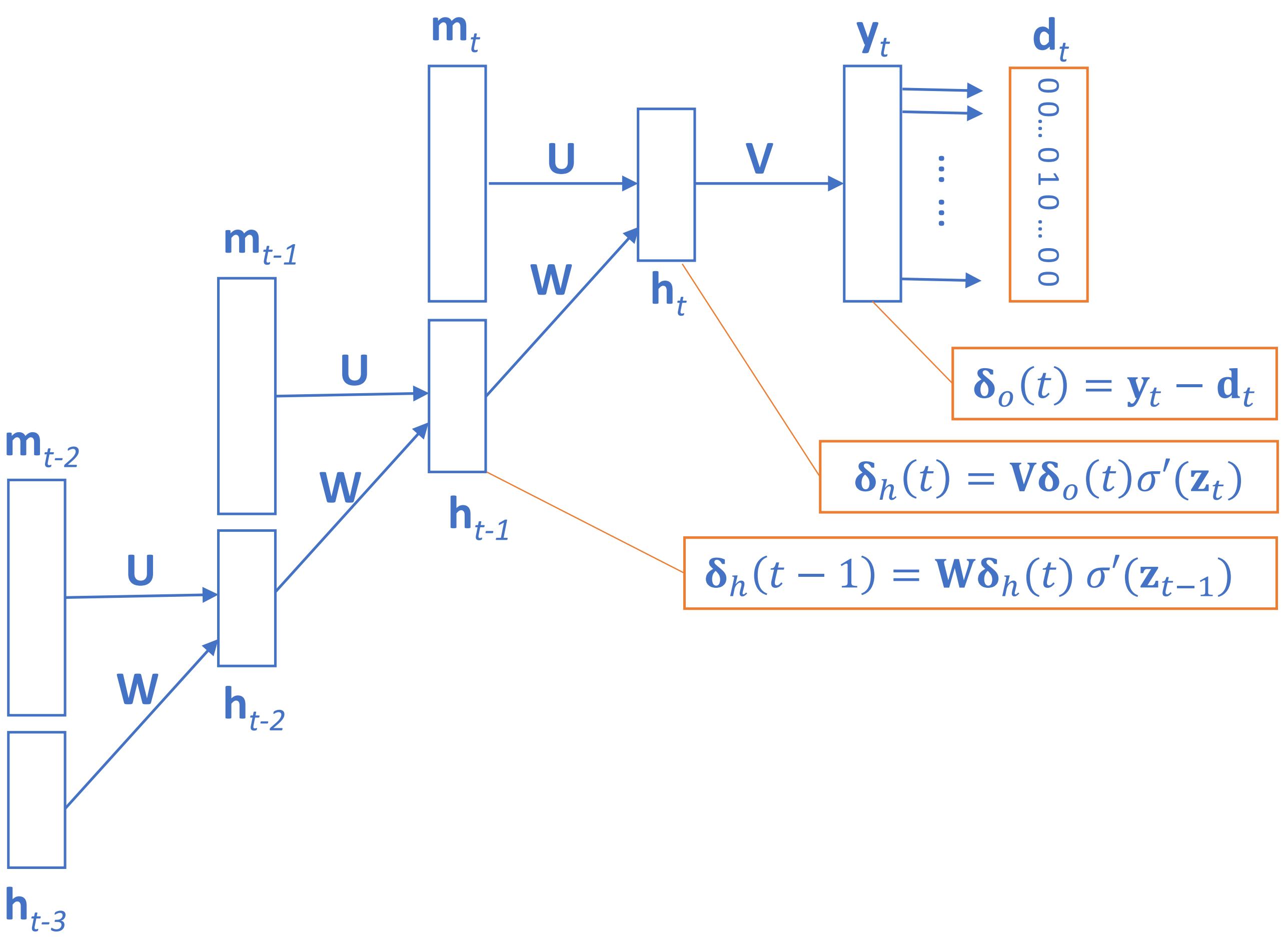


$$\mathbf{z}_t = \mathbf{U}\mathbf{m}_t + \mathbf{W}\mathbf{h}_{t-1}$$
$$\mathbf{h}_t = \sigma(\mathbf{z}_t)$$
$$\mathbf{y}_t = g(\mathbf{V}\mathbf{h}_t)$$

where

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

Training RNN-LM by backpropagation through time



Forward pass:

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

where

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

Parameter updates in backpropagation:

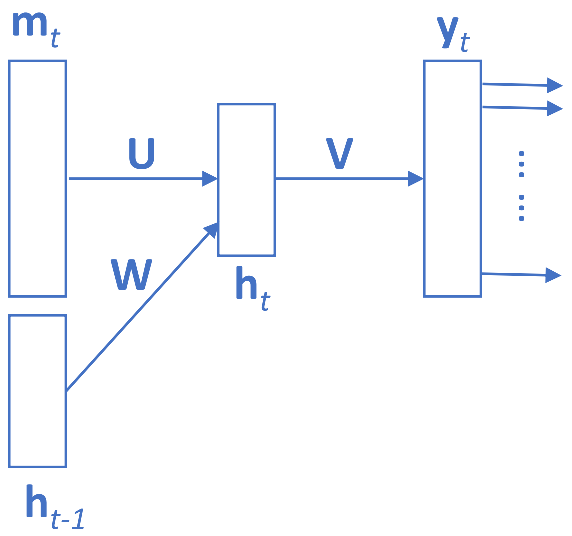
$$\mathbf{V}^{new} = \mathbf{V}^{old} - \eta\boldsymbol{\delta}_o(t)\mathbf{h}_t^T$$
$$\mathbf{U}^{new} = \mathbf{U}^{old} - \eta\sum_{\tau=0}^{T}\boldsymbol{\delta}_h(t-\tau)\mathbf{m}_{t-\tau}^T$$
$$\mathbf{W}^{new} = \mathbf{W}^{old} - \eta\sum_{\tau=0}^{T}\boldsymbol{\delta}_h(t-\tau)\mathbf{h}_{t-\tau-1}^T$$

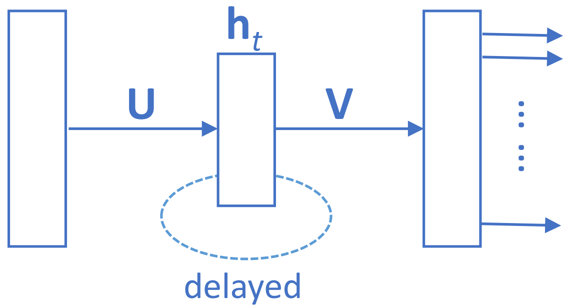
Pseudo code for BPTT

```
Back_Propagation_Through_Time(a, y)   // a[t] is the input at time t. y[t] is the output
    Unfold the network to contain k instances of f
    do until stopping criteria is met:
        x = the zero-magnitude vector;// x is the current context
        for t from 0 to n - 1         // t is time. n is the length of the training sequence
            Set the network inputs to x, a[t], a[t+1], ..., a[t+k-1]
            p = forward-propagate the inputs over the whole unfolded network
            e = y[t+k] - p;           // error = target - prediction
            Back-propagate the error, e, back across the whole unfolded network
            Update all the weights in the network
            Average the weights in each instance of f together, so that each f is identical
            x = f(x);                 // compute the context for the next time-step
```

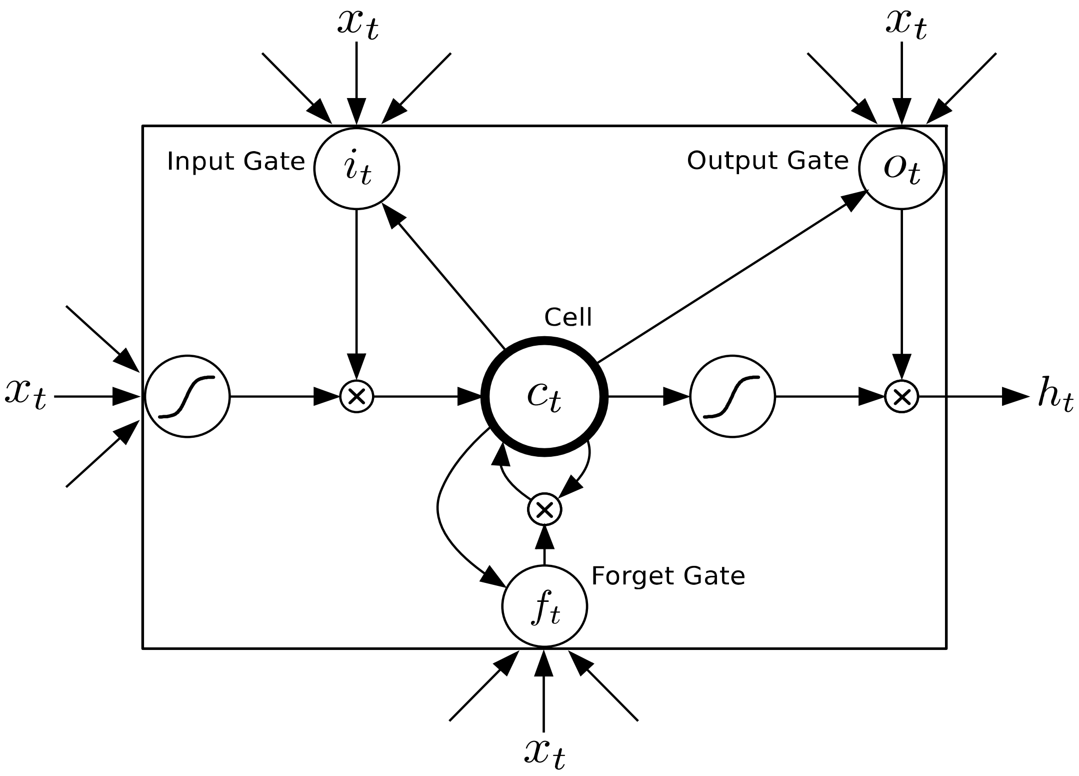
Potentials and difficulties of RNN

- In theory, RNN can "store" in h all information about past inputs.
- But in practice, standard RNN cannot capture very long distance dependency
- Vanishing gradient problem in backpropagation
  - $\boldsymbol{\delta}$ may vanish after repeated multiplication with $\sigma'(.)$
- Solution: long short-term memory (LSTM)





A Long Short-Term Memory cell in LSTM-RNN

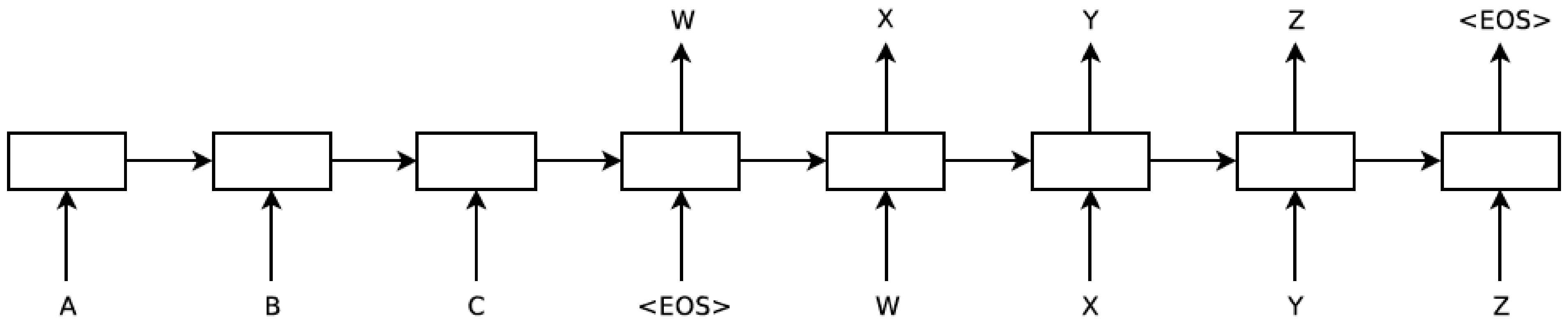


$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right)$$
$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right)$$
$$c_t = f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right)$$
$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right)$$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

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

LSTM for machine translation (MT)

- “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

[Sutskever et al. 2014]

Mission of Machine (Deep) Learning

"Real" world	Data (collected/labeled)
"Artificial" world	Model (architecture)
Link the two worlds	Training (algorithm)

Q&A

- http://research.microsoft.com/en-us/um/people/jfgao/
- jfgao@microsoft.com

- We are hiring!
- http://research.microsoft.com/en-us/groups/dltc/
- http://research.microsoft.com/en-us/projects/dssm/

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.
- Bengio, Y., 2009. Learning deep architectures for AI. Foundations and Trends in 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.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P., 2011. Natural language processing (almost) from scratch. in JMLR, vol. 12.
- Dahl, G., Yu, D., Deng, L., and Acero, 2012. A. Context-dependent, pre-trained deep neural networks for large vocabulary speech recognition, IEEE Trans. Audio, Speech, & Language Proc., Vol. 20 (1), pp. 30-42.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T., and Harshman, R. 1990. Indexing by latent semantic analysis. J. American Society for Information Science, 41(6): 391-407
- Deng, L., Seltzer, M., Yu, D., Acero, A., Mohamed, A., and Hinton, G., 2010. Binary Coding of Speech Spectrograms Using a Deep Auto-encoder, in Interspeech.
- Deng, L. and Yu, D. 2014. Deeping learning methods and applications. Foundations and Trends in Signal Processing 7:3-4.
- Deng, L., Yu, D., and Platt, J. 2012. Scalable stacking and learning for building deep architectures, Proc. ICASSP.
- Deoras, A., and Sarikaya, R., 2013. Deep belief network based semantic taggers for spoken language understanding, in INTERSPEECH.
- Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., and Makhoul, J., 2014. Fast and Robust Neural Network Joint Models for Statistical Machine Translation, ACL.
- Duh, K. 2014. Deep learning for natural language processing and machine translation. Tutorial. 2014.
- 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.
- Hochreiter, S. and Schmidhuber, J. 1997. Long short-term memory. Neural Computation, 9(8): 1735-1780, 1997.
- Gao, J., He, X., Yih, W-t., and Deng, L. 2014a. Learning continuous phrase representations for translation modeling. In ACL.
- Gao, J., He, X., and Nie, J-Y. 2010. Clickthrough-based translation models for web search: from word models to phrase models. In CIKM.
- Gao, J., Pantel, P., Gamon, M., He, X., and Deng, L. 2014b. Modeling interestingness with deep neural networks. 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.
- Graves, A., Jaitly, N., and Mohamed, A., 2013a. Hybrid speech recognition with deep bidirectional LSTM, Proc. ASRU.

References

- He, X. and Deng, L., 2013. Speech-Centric Information Processing: An Optimization-Oriented Approach, in Proceedings of the IEEE.
- He, X., Deng, L., and Chou, W., 2008. Discriminative learning in sequential pattern recognition, Sept. IEEE Sig. Proc. Mag.
- Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B., 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition, IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82-97.
- Hinton, G., 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.
- Kiros, R., Zemel, R., and Salakhutdinov, R. 2013. Multimodal Neural Language Models, Proc. NIPS Deep Learning Workshop.
- Krizhevsky, A., Sutskever, I, and Hinton, G., 2012. ImageNet Classification with Deep Convolutional Neural Networks, 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.
- Li, P., Hastie, T., and Church, K.. 2006. Very sparse random projections, in Proc. SIGKDD.
- Manning, C. and Schutze, H. 1999. Foundations of statistical natural language processing. The MIT Press.
- 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.

References

- Mohamed, A., Yu, D., and Deng, L. 2010. Investigation of full-sequence training of deep belief networks for speech recognition, Proc. Interspeech.
- Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. 2011. Multimodal deep learning, Proc. ICML.
- 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
- Sarikaya, R., Hinton, G., and Ramabhadran, B., 2011. Deep belief nets for natural language call-routing, in Proceedings of the ICASSP.
- Schwenk, H., Dchelotte, D., Gauvain, J-L., 2006. Continuous space language models for statistical machine translation, in COLING-ACL
- Seide, F., Li, G., and Yu, D. 2011. Conversational speech transcription using context-dependent deep neural networks, Proc. Interspeech
- Shen, Y., He, X., Gao, J., Deng, L., and Mesnil, G. 2014. A convolutional latent semantic model for web search. CIKM 2014.
- Socher, R., Huval, B., Manning, C., Ng, A., 2012. Semantic compositionality through recursive matrix-vector spaces. In EMNLP.
- Socher, R., and Manning, C. 2013. Deep learning for NLP (without magic). Tutorial In NAACL.
- Socher, R., Lin, C., Ng, A., and Manning, C. 2011. Learning continuous phrase representations and syntactic parsing with recursive neural networks, Proc. ICML.
- Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C., Ng A., and Potts. C. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, Proc. EMNLP
- Song, X. He, X., Gao. J., and Deng, L. 2014. Learning Word Embedding Using the DSSM. MSR Tech Report.
- Sutskever, I., Vinyals, O., and Le, Q. 2014. Sequence to sequence learning with neural networks. In *NIPS*.
- Xu, P., and Sarikaya, R., 2013. Convolutional neural network based triangular crf for joint intent detection and slot filling, in IEEE ASRU.
- 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.
- Zeiler, M. and Fergus, R. 2013. Visualizing and understanding convolutional networks, arXiv:1311.2901, pp. 1-11.