DSSM

Deep Structured Semantic Model

a framework of learning semantic embedding

Background

- Traditionally, search engines retrieve web documents by matching terms in documents with those in a search query – **lexical matching**
- However, lexical matching can be suboptimal due to language discrepancy between documents and queries
  - E.g., a concept can often be expressed using different vocabularies and language styles
- Need to bridge the lexical gaps between queries and documents– **semantic matching**

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

Deep Semantic Similarity Model (DSSM)

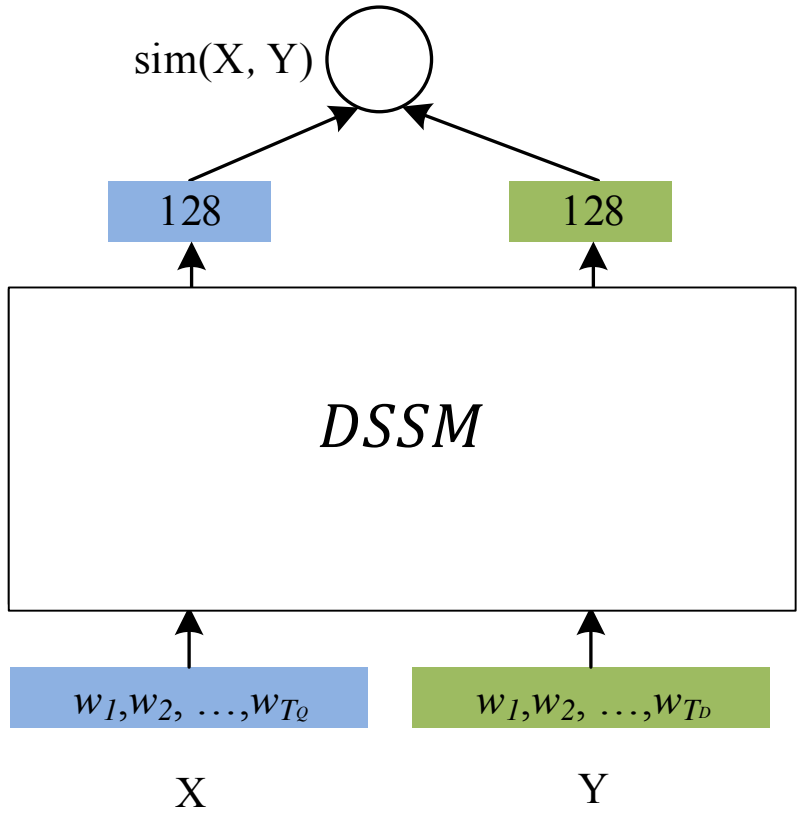
[Huang et al. 2013; Gao et al. 2014a; Gao et al. 2014b; Shen et al. 2014]

- 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*

DSSM: Compute Similarity in Semantic Space

Relevance measured by cosine similarity

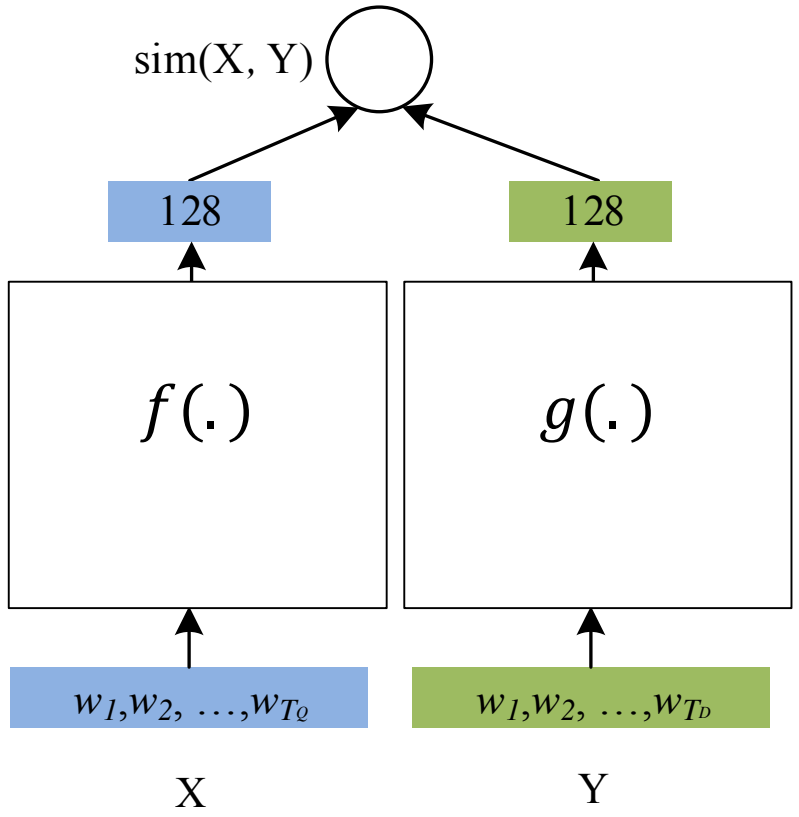


Word sequence x_t

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

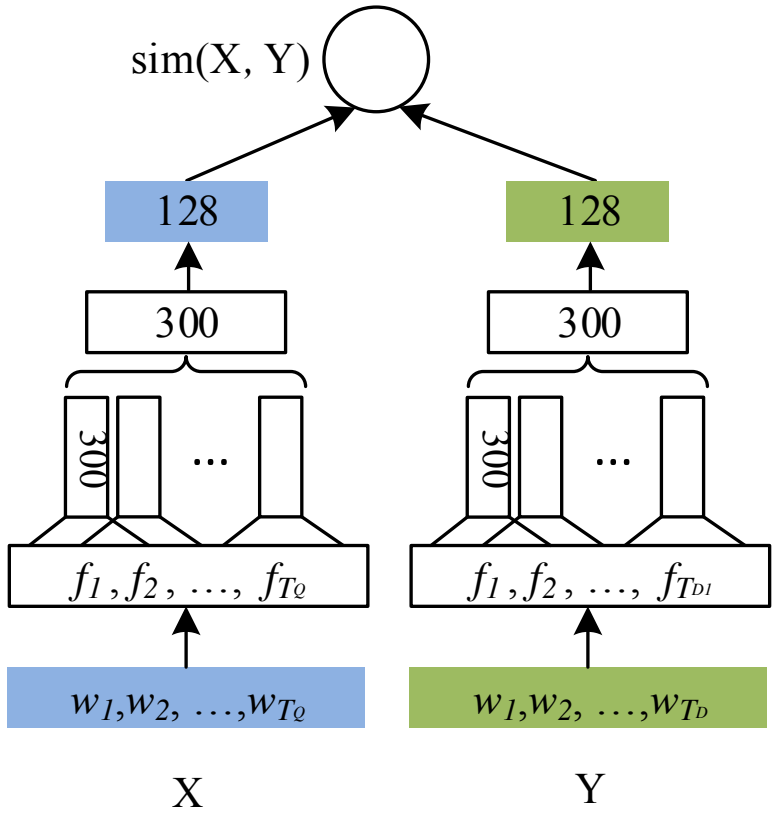
Semantic layer h

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

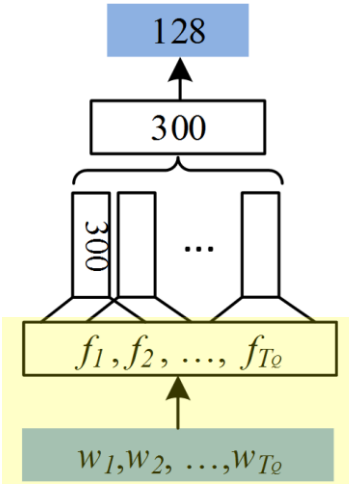
Convolutional and Max-pooling layer: identify key words/concepts in X and Y

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

DSSM solution

- Using the ***tri-letter*** based word hashing for scalable word representation
- Using the ***deep neural net*** to extract high-level semantic representations
- Using the ***click signal*** to guide the learning

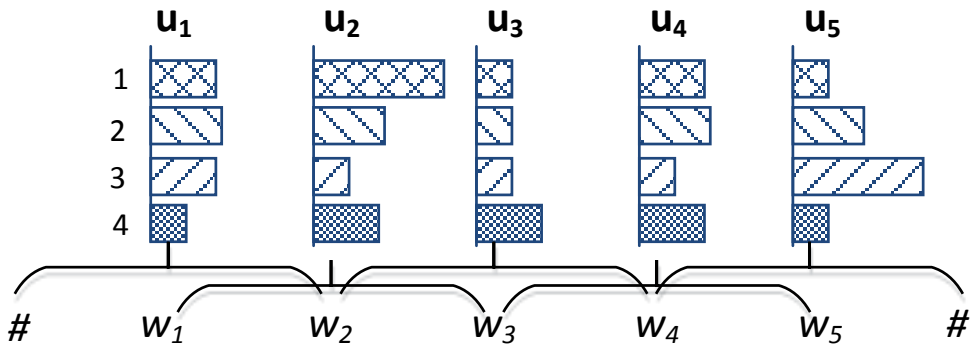
Letter-trigram Representation

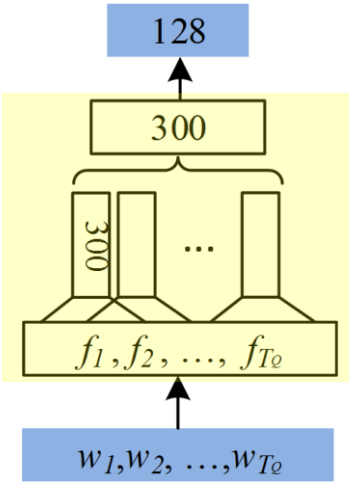


- Control the dimensionality of the input space
  - e.g., cat → #cat# → #-c-a, c-a-t, a-t-#
  - Only ~50K letter-trigrams in English; no OOV issue
- Capture sub-word semantics (e.g., prefix & suffix)
- Words with small typos have similar raw representations
- Collision: different words with same letter-trigram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

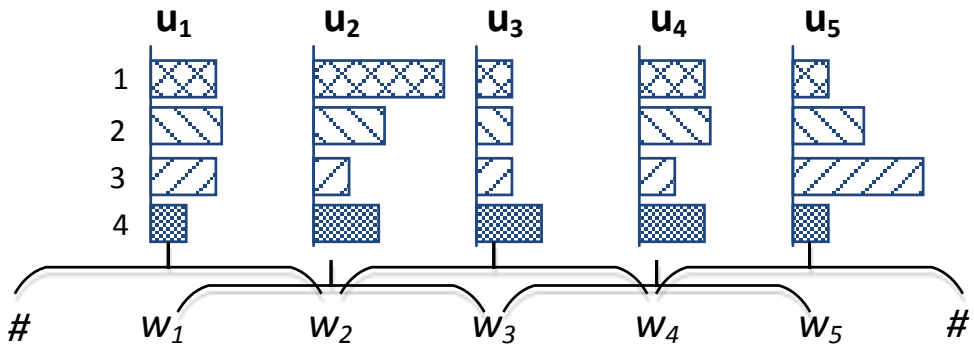
Convolutional Layer

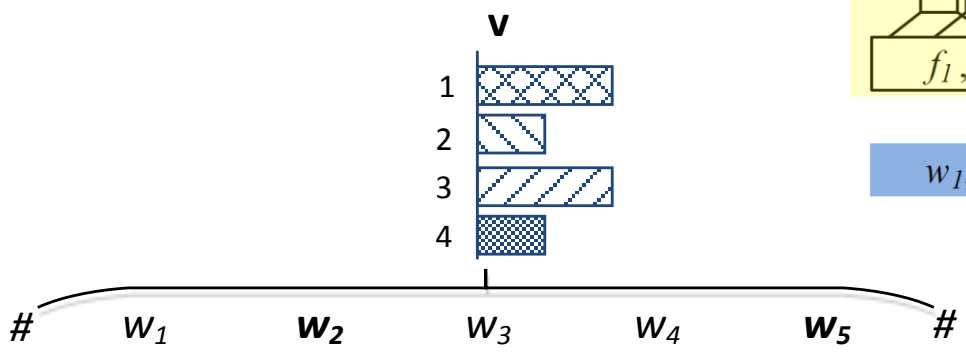


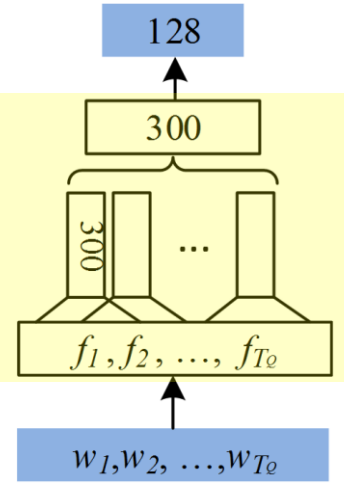


- Extract local features using convolutional layer
  - {w1, w2, w3} → topic 1
  - {w2, w3, w4} → topic 4

Max-pooling Layer

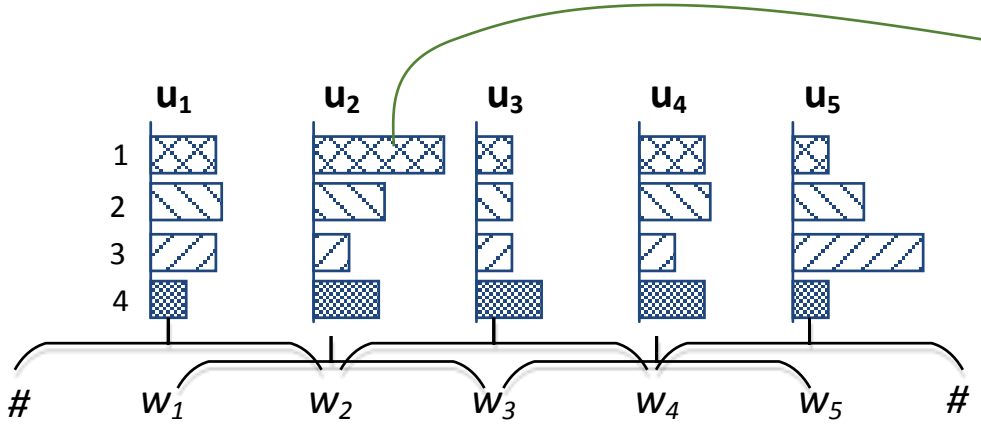


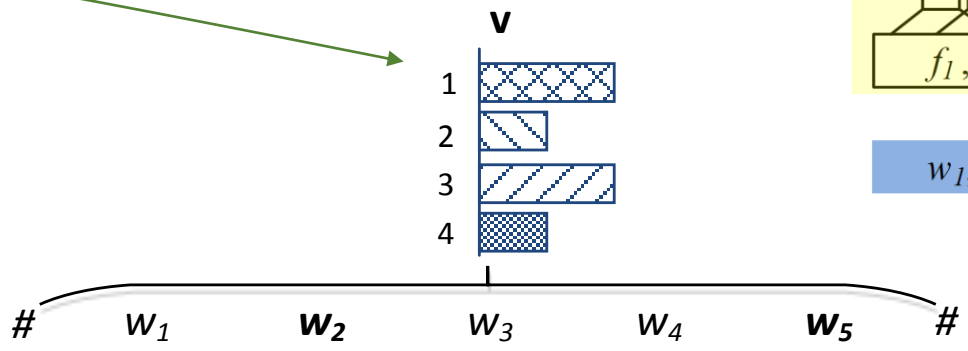


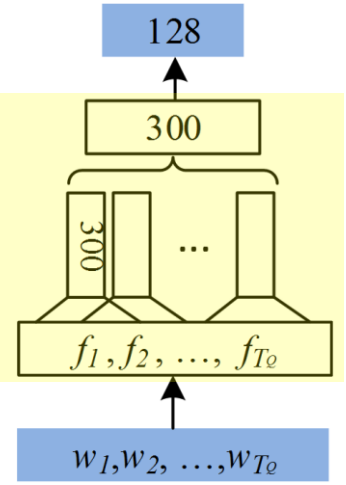


- Extract local features using convolutional layer
  - {w1, w2, w3} → topic 1
  - {w2, w3, w4} → topic 4
- Generate global features using max-pooling
  - Key topics of the text → topics 1 and 3
  - keywords of the text: w2 and w5

Max-pooling Layer



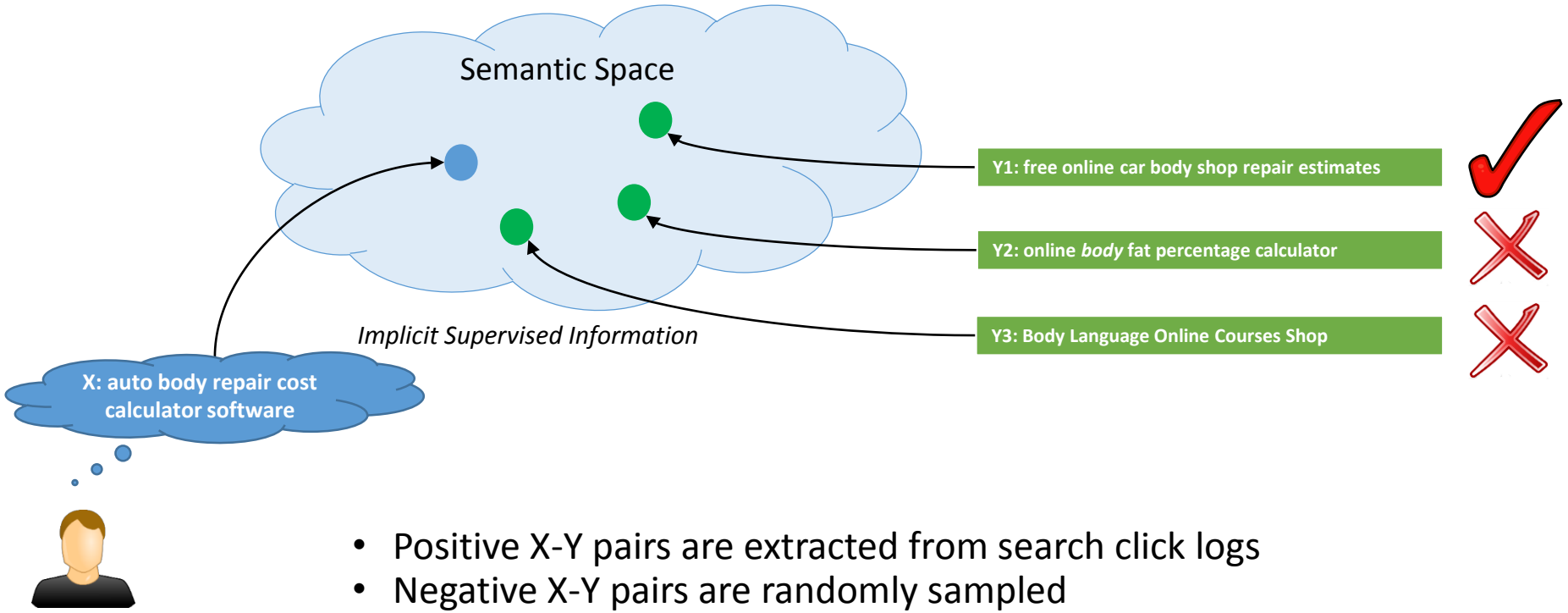




- Extract local features using convolutional layer
  - {w1, w2, w3} → topic 1
  - {w2, w3, w4} → topic 4
- Generate global features using max-pooling
  - Key topics of the text → topics 1 and 3
  - keywords of the text: w2 and w5

… 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 …

Learning DSSM from Labeled X-Y Pairs

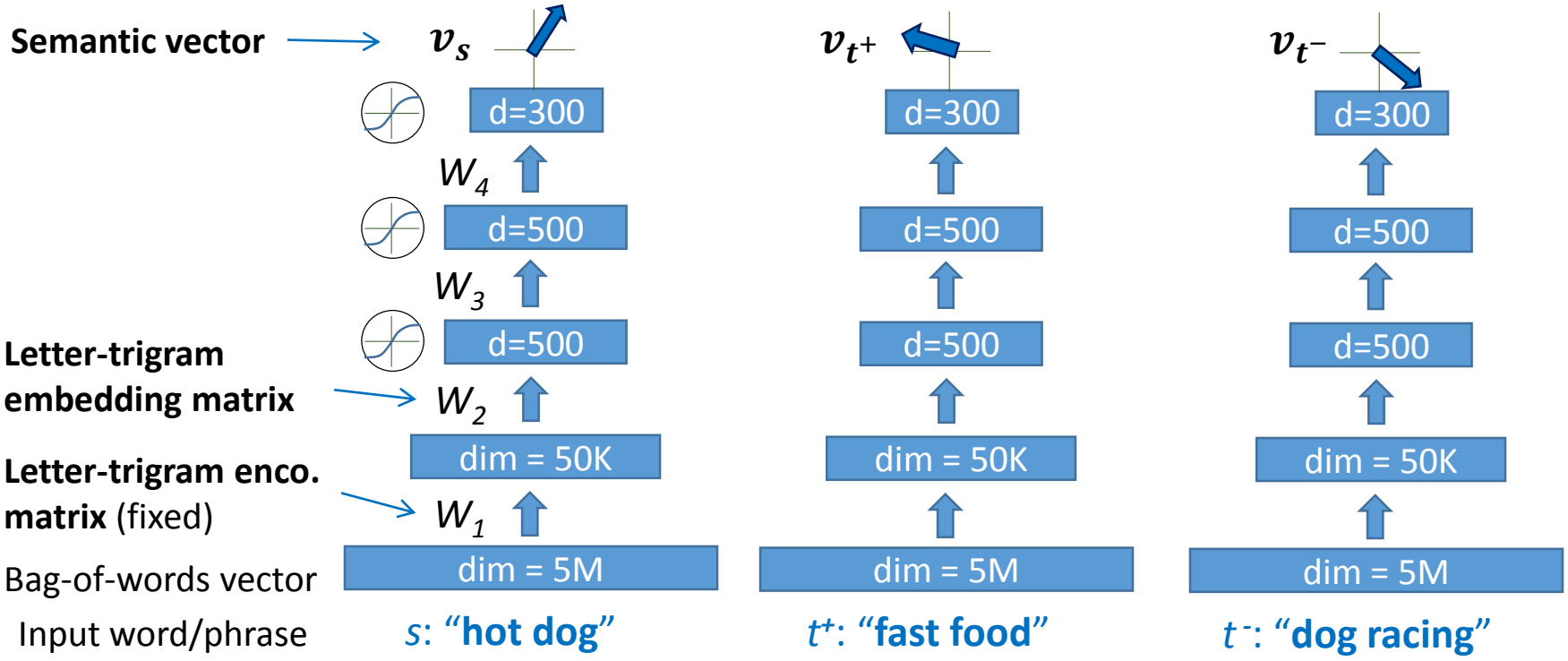


- Positive X-Y pairs are extracted from search click logs
- Negative X-Y pairs are randomly sampled
- Map X and Y into the same semantic space via deep neural net

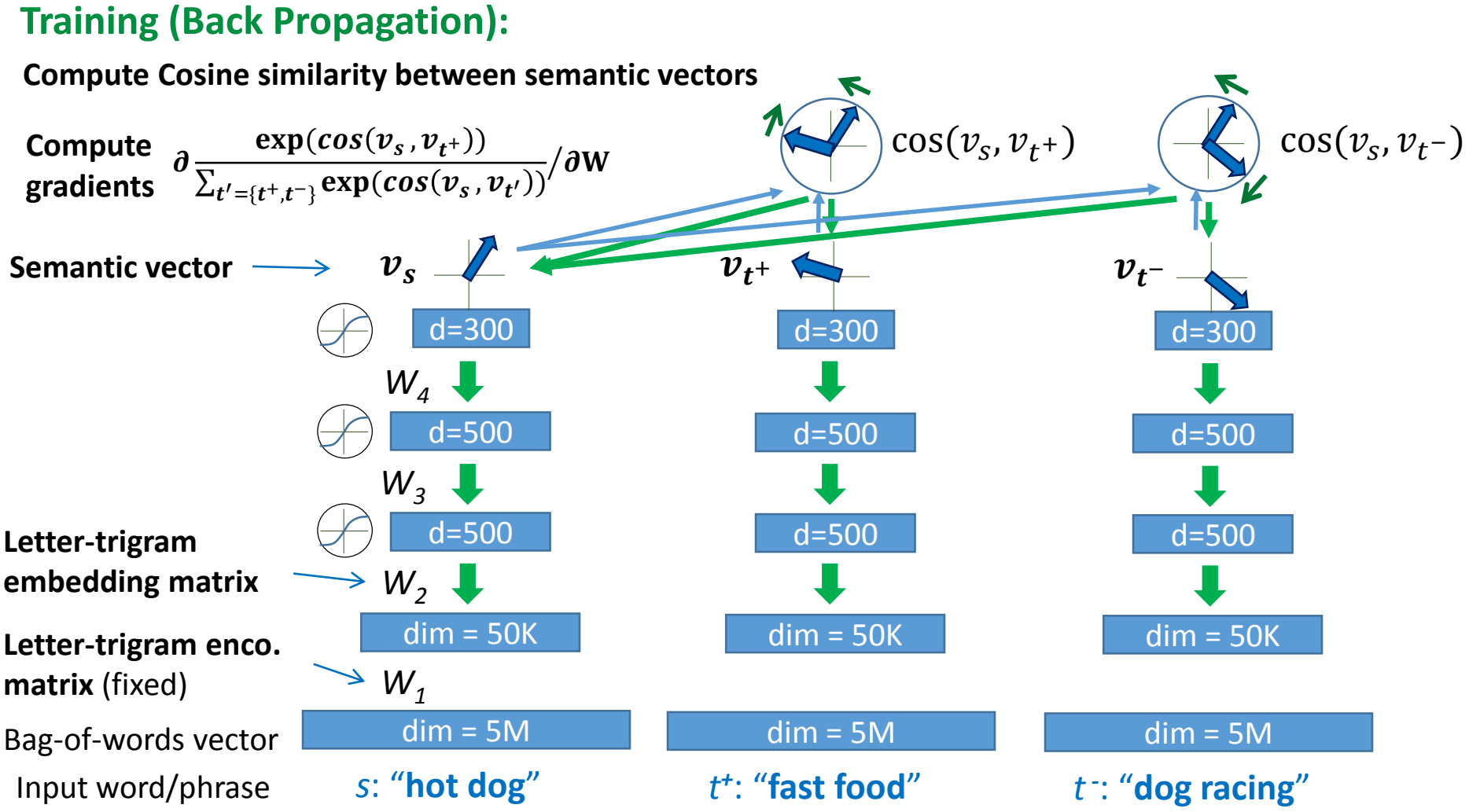
Learning DSSM on X-Y pairs via SGD

Initialization:

Neural networks are initialized with random weights

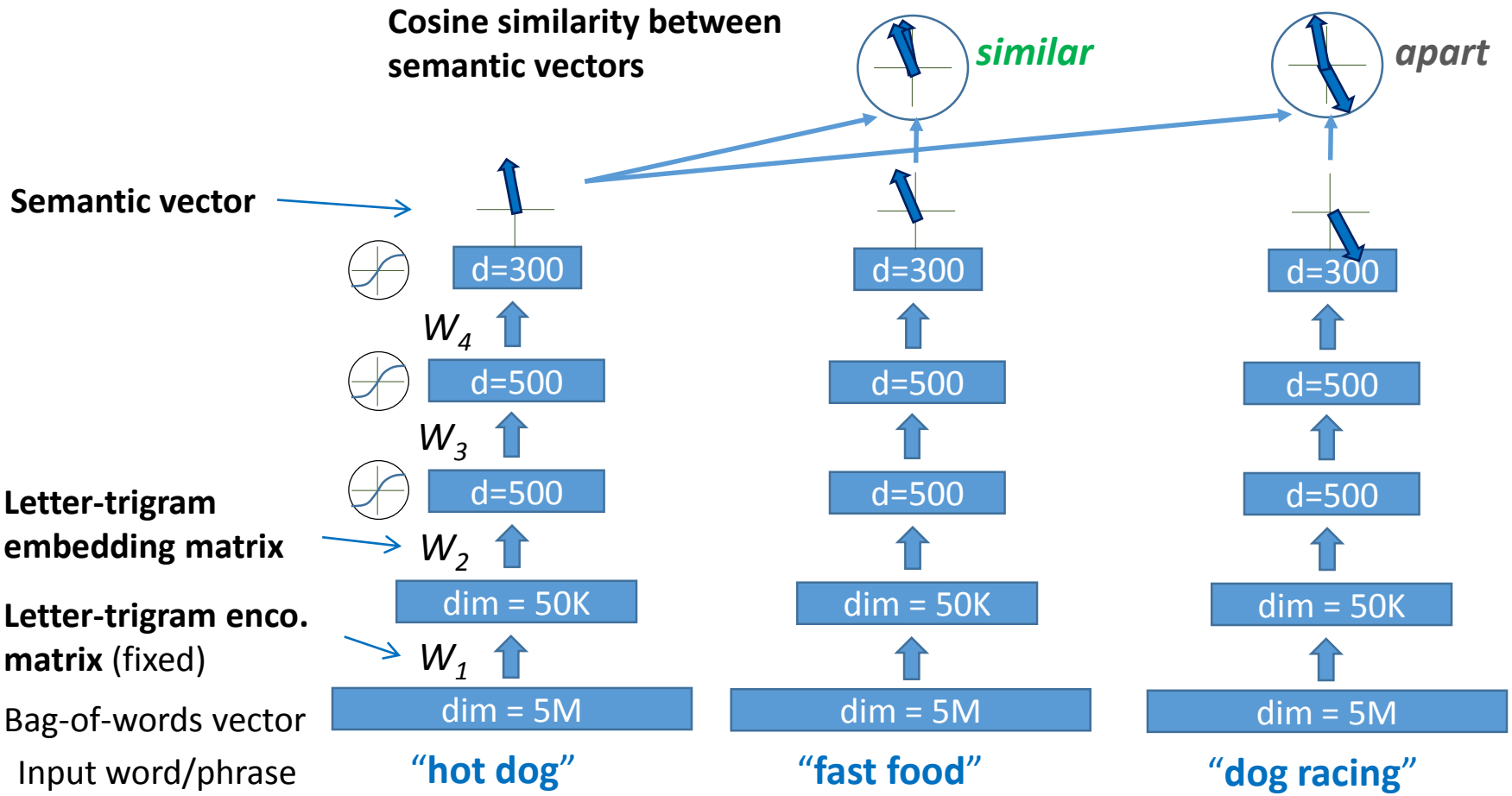


Learning DSSM on X-Y pairs via SGD



Learning DSSM on X-Y pairs via SGD

After training converged:



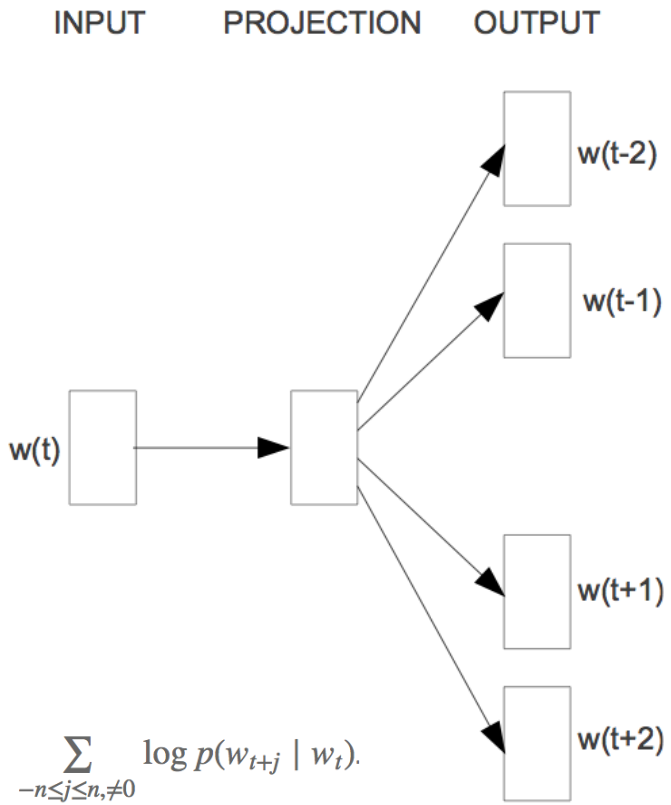
DSSM vs word2vec

Word2vec =

- Skip gram
- CBOW

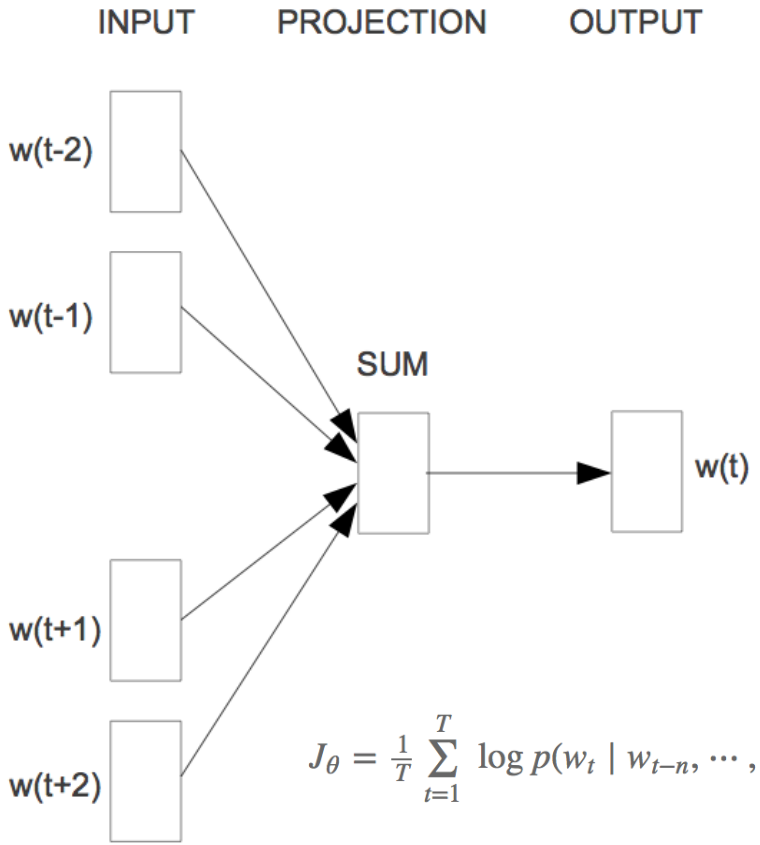
X

- Hierarchical Softmax
- Negative Sampling



$$J_\theta = \frac{1}{T}\sum_{t=1}^{T} \sum_{-n \le j \le n, \neq 0} \log p(w_{t+j} \mid w_t).$$

Skip gram



$$J_\theta = \frac{1}{T}\sum_{t=1}^{T} \log p(w_t \mid w_{t-n}, \cdots, w_{t-1}, w_{t+1}, \cdots, w_{t+n}).$$

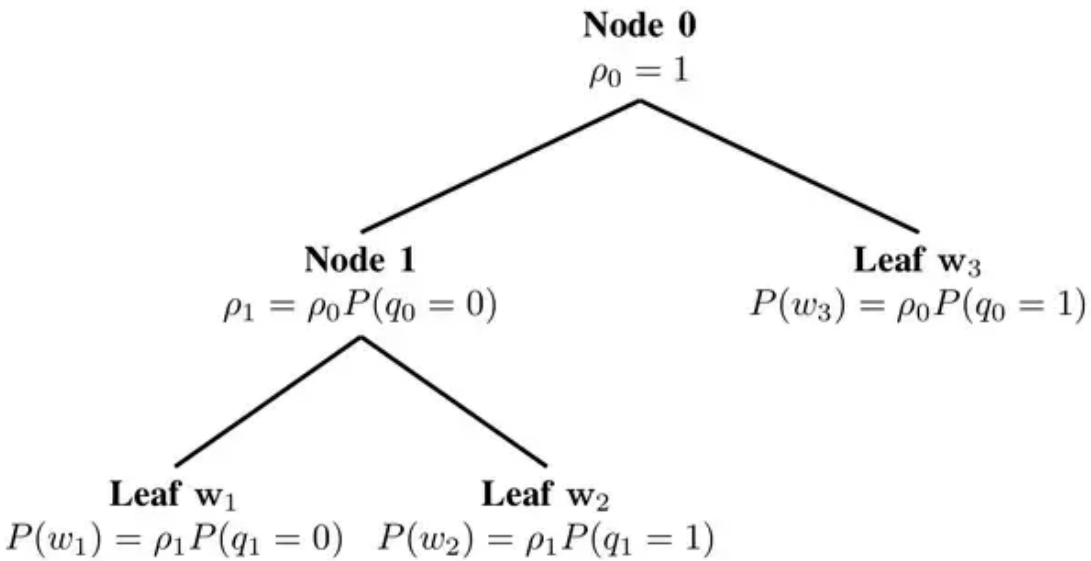
CBOW

Word2vec =

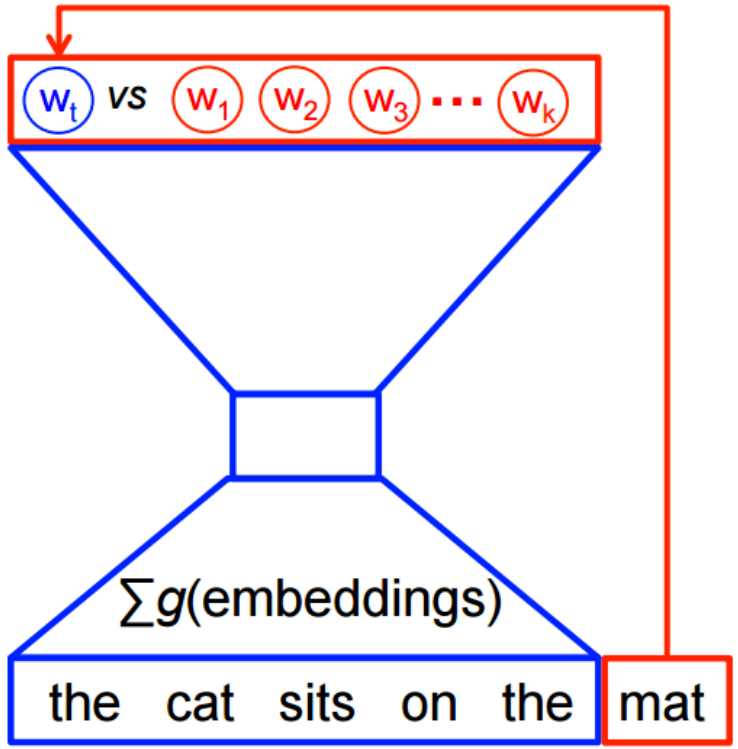
- Skip gram
- CBOW

x

- Hierarchical Softmax
- Negative Sampling



Hierarchical Softmax



Negative Sampling

DSSM vs word2vec

- Word embedding

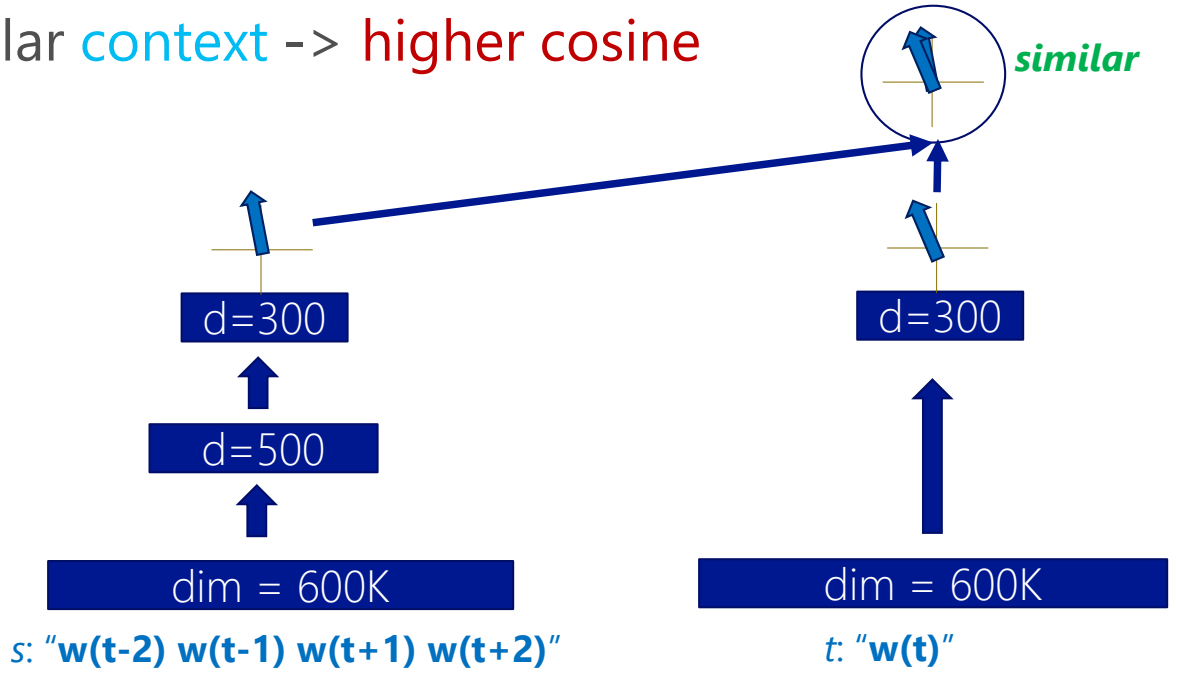
DSSM for word embedding

- Treat word as a sentence (performs badly)
- Learning word embedding using DSSM

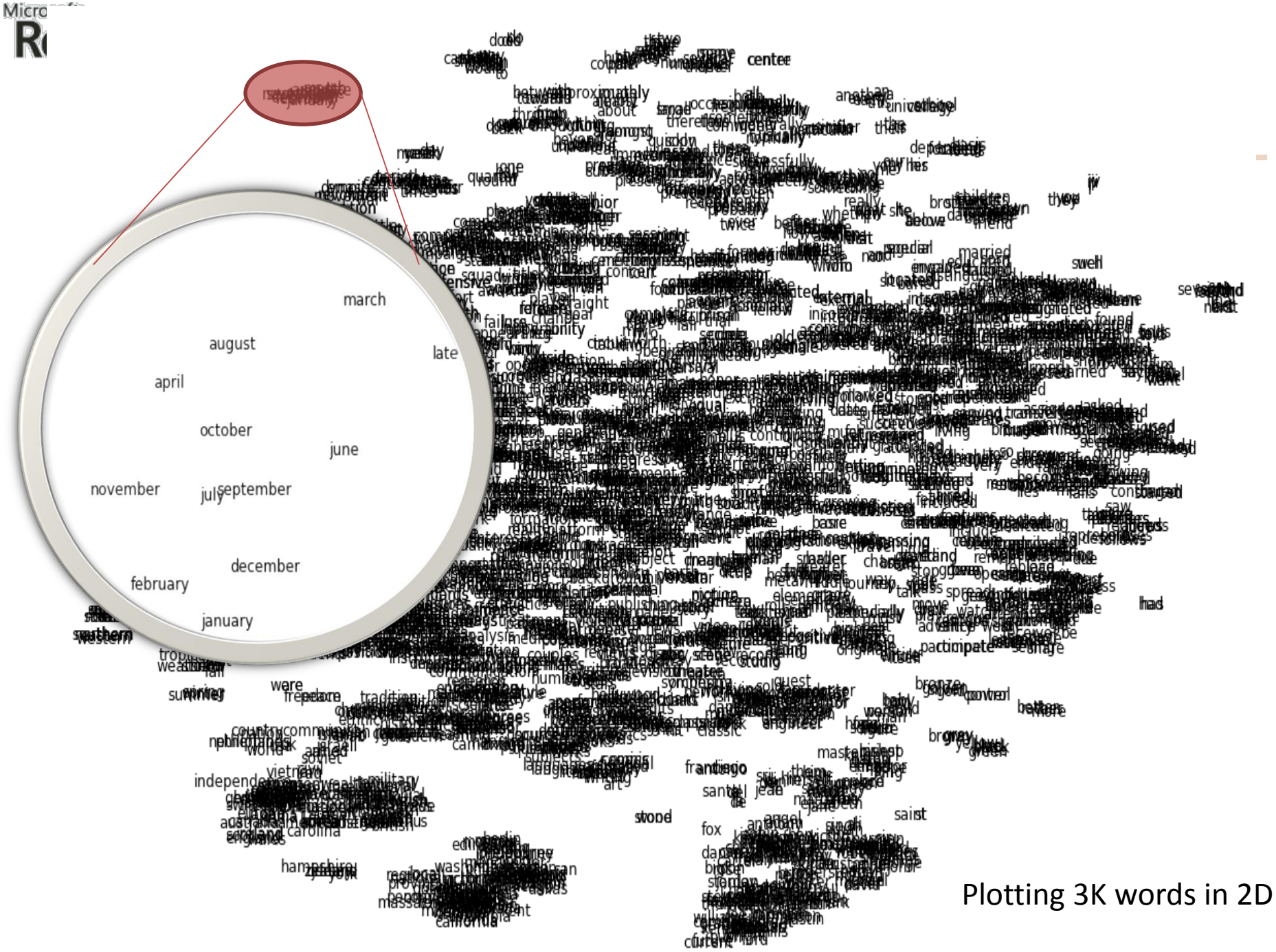
DSSM: learning words' meaning

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

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



[Song, He, Gao, Deng, Shen, 2014]



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	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.

DSSM vs word2vec

- Word embedding - enwik9
  - Similarity

Model \ $\rho \times 100$	F. et al. WS353	B. et al. MEN	R. et al. MT	S. et al. Rel122	R. et al. RG
skipgram (neg=10)	63.2	59.8	61.8	53.3	64.0
skipgram (neg=100)	59.5	58.1	60.8	53.8	64.1
char n-gram	59.5	21.7	60.3	51.0	51.3
GloVe	62.6	65.1	60.2	48.8	58.1
DSSM	51.1	41.5	44.1	31.6	37.7

DSSM vs word2vec

- Word embedding - enwik9
  - Analogy

	Google		Microsoft	
Model	3CosAdd	3CosMul	3CosAdd	3CosMul
skipgram (neg=10)	29.1	27.9	22.8	22.6
skipgram (neg=100)	30.4	29.9	22.9	22.6
char n-gram	38.0	38.0	38.7	38.8
GloVe	41.1	33.9	28.0	24.5
DSSM	31.6	31.8	41.4	41.7

What else can DSSM do ?

Deep Structured Semantic Model (DSSM) in practice

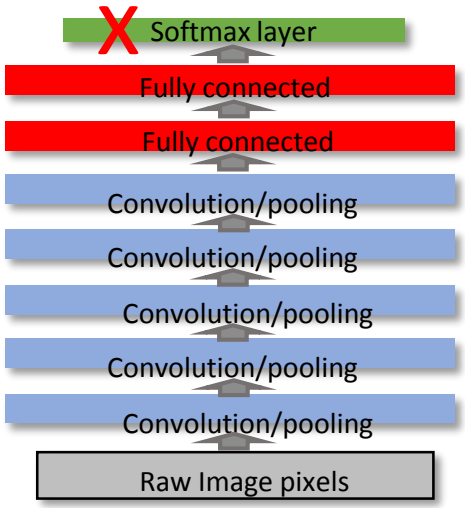
Tasks	***Source***	***Target***
Word semantic embedding	*context*	*word*
Web search	*search query*	*web documents*
Question answering	*pattern / mention* (in NL)	*relation / entity* (in KB)
Recommendation	***doc in reading***	***interesting things / other docs***
Machine translation	***sentence in language a***	***translations in language b***
Text/Image joint learning	*text / image*	*Image / text*
Ad selection	***search query***	***ad keywords***
Entity ranking	***mention (highlighted)***	***entities***
Knowledge-base expansion	***entity***	***entity***
…		

See our latest publication at DLTC:
http://research.microsoft.com/en-us/groups/dltc/

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.



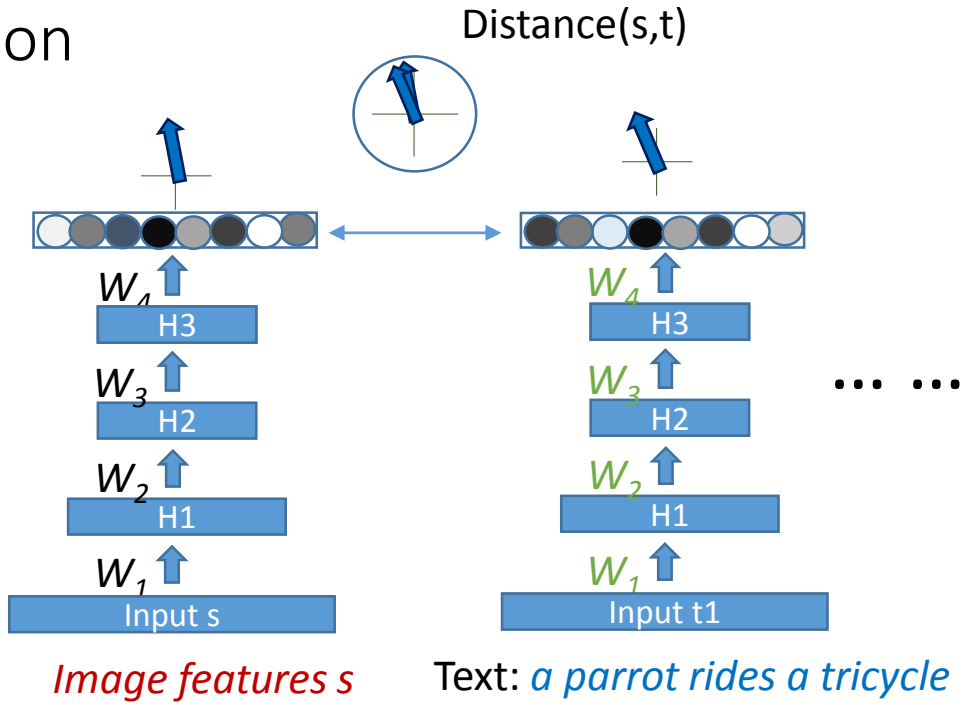


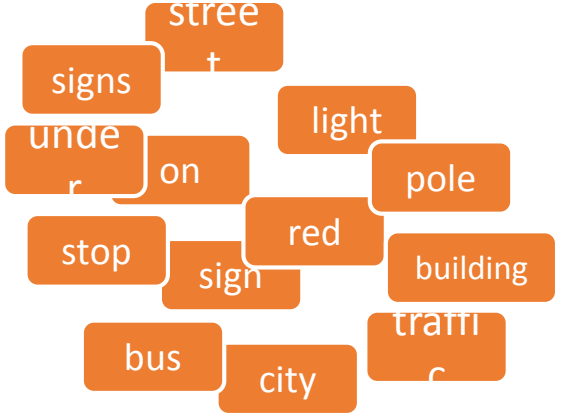
Image features s

Text: *a parrot rides a tricycle*

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

Detector Models, Deep Neural Net Features

Computer Vision System



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

a stop sign at an intersection on a city street

DSSM Model

Semantic Ranking System

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

DSSM in recommendation

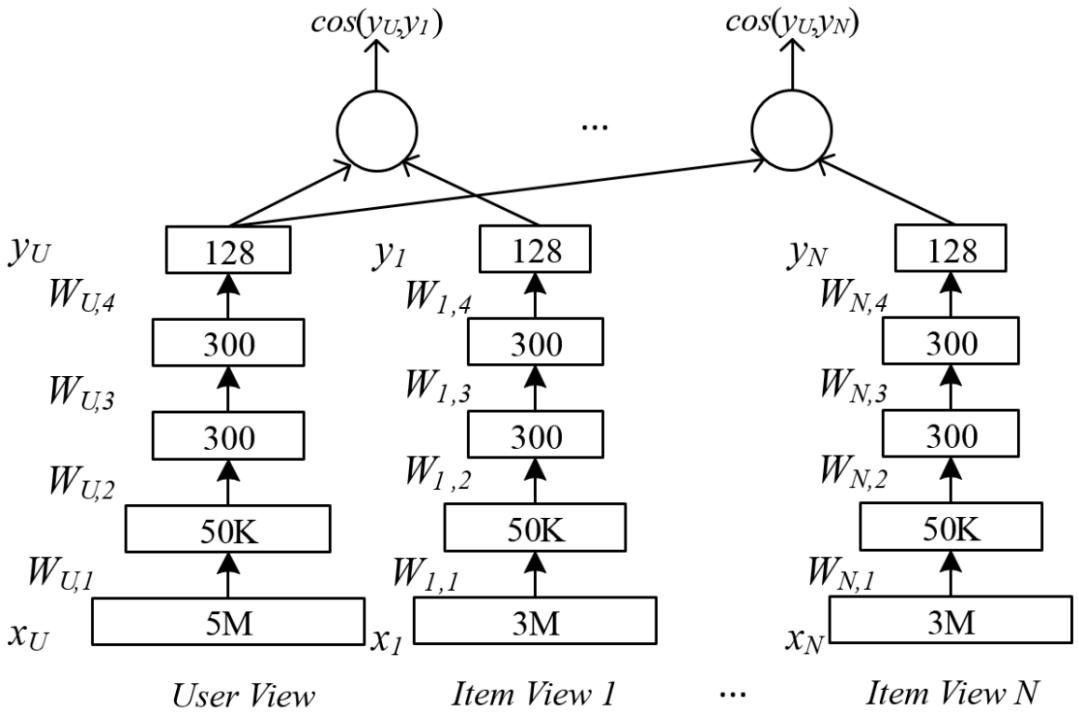
- Modeling Interestingness with Deep Neural Networks
- A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems/Multi-Rate Deep Learning for Temporal Recommendation

Modeling Interestingness with Deep Neural Networks

- 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
- Features: document text

- We can leverage this idea to make video -> video recommendation w/ w/o video content.

A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems/Multi-Rate

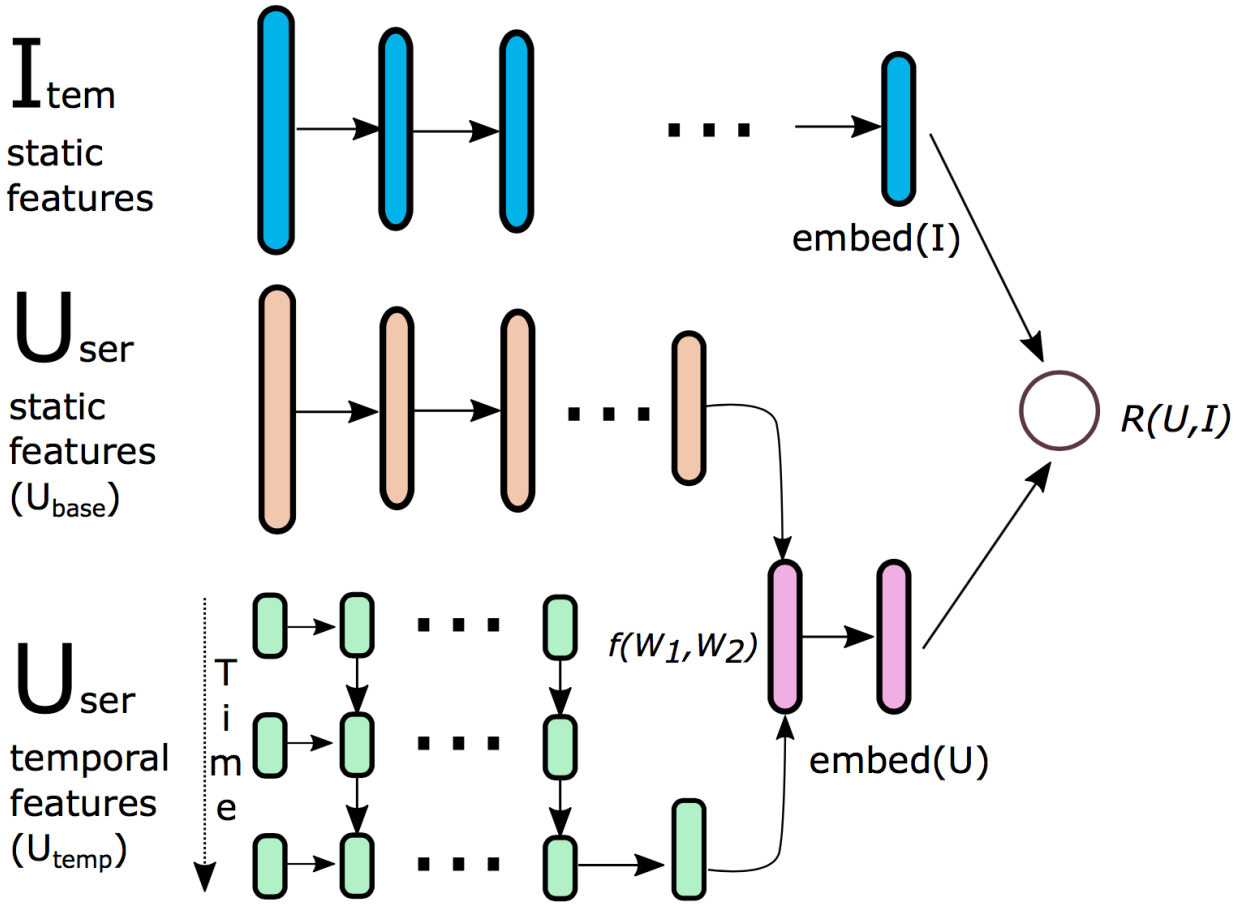


Type	DataSet	UserCnt	Feature Size	Joint Users
User View	Search	20M	3.5M	/
Item View	News	5M	100K	1.5M
	Apps	1M	50K	210K
	Movie/TV	60K	50K	16K

We can leverage the framework in real time recommendation

- User features: search history (unigram, bigram), clicked hosts(one hot)
- Item features: title/description letter-trigram
- Items: News, App, Movie/TV (xbox)

Multi-Rate Deep Learning for Temporal Recommendation



We can leverage the framework in real time recommendation

- User features: search history (unigram, bigram), clicked hosts(one hot)
- Item features: title/description letter-trigram
- User temporal features
- Items: News, App, Movie/TV (xbox)

Thanks!

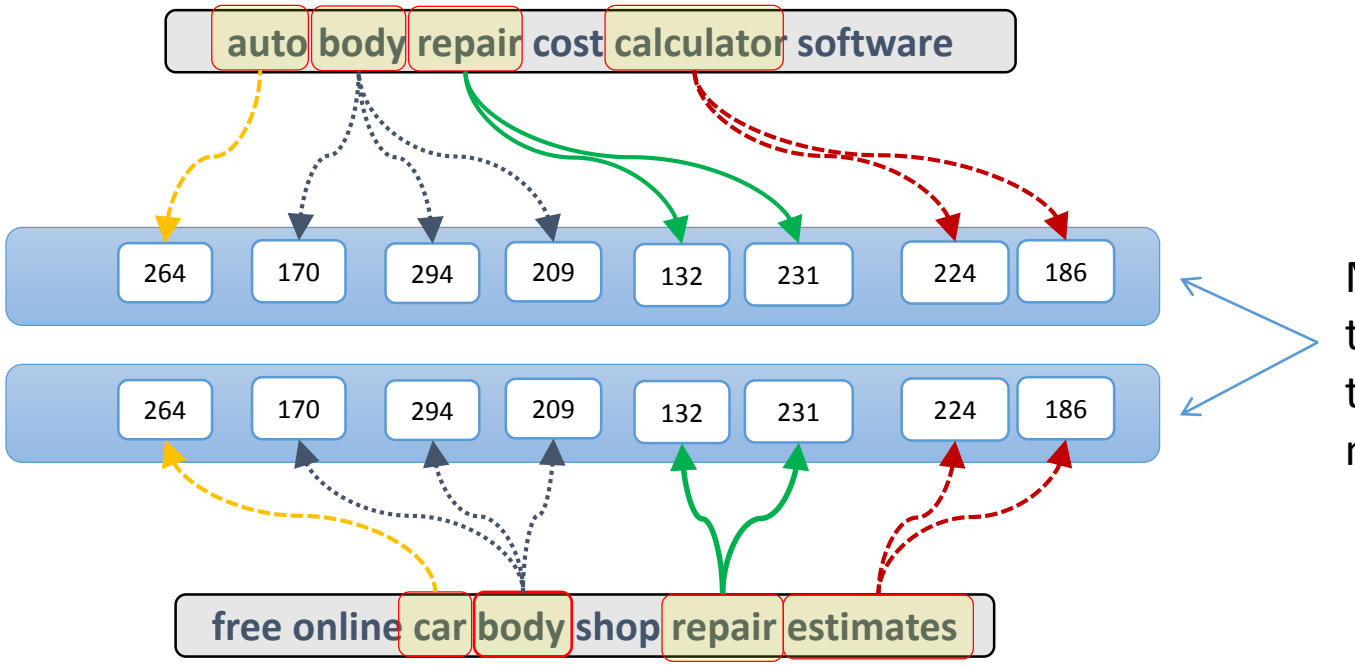
References

On word embeddings - Part 1 link
On word embeddings - Part 2: Approximating the Softmax link
Word embeddings benchmarks link
DSSM link
Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. Link
The WSDM 2017 Tutorial on Neural Text Embeddings for Information Retrieval **link**
Distributed representations of words and phrases and their compositionality (2013), T. Mikolov et al. *(Google)*[pdf]
Efficient estimation of word representations in vector space (2013), T. Mikolov et al. *(Google)* [pdf]
Evaluation methods for unsupervised word embeddings. (2015), Tobias Schnabel, Igor Labutov, David Mimno, Thorsten Joachims.
How to evaluate word embeddings?On importance of data efficiency and simple supervised tasks, Stanisław Jastrzebski, et al.
Deep Learning for Natural Language Processing and Related Applications (2014), Xiaodong He, Jianfeng Gao, and Li Deng
Improving Word Embeddings with Convolutional Feature Learning and Subword Information, Shaosheng Cao and Wei Lu.
Unsupervised Learning of Word Semantic Embedding using the Deep Structured Semantic Model (2014), Xinying Song, et al.
Deep Learning for Selected Natural Language Applications, Xiaodong He
Vectorland: Brief Notes from Using Text Embeddings for Search (2015), Bhaskar Mitra
Deep Learning for Web Search and Natural Language Processing (2015), Jianfeng Gao
Modeling Interestingness with Deep Neural Networks, Jianfeng Gao, Patrick Pantel, Michael Gamon, Xiaodong He, Li Deng
A multi-view deep learning approach for cross domain user modeling in recommendation systems, A. M. Elkahky, Y. Song, and X. He.
Multi-Rate Deep Learning for Temporal Recommendation, Yang Song, Ali Mamdouh Elkahk and Xiaodong He.

backup

Intent matching via convolutional-pooling

- Semantic matching of query and document

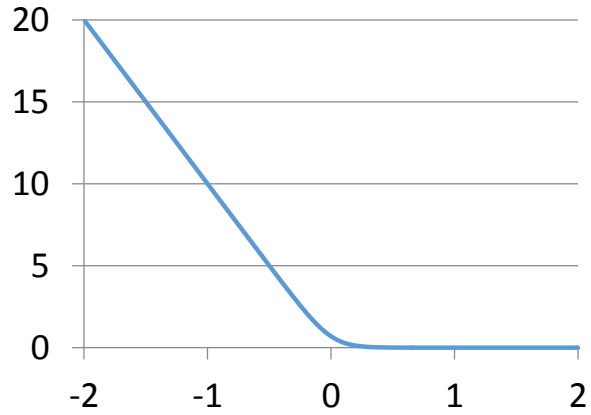


Most active neurons at the **max-pooling layers** of the query and document nets, respectively

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
- $\text{sim}_{\boldsymbol{\theta}}(X, Y)$ is the cosine similarity of X and Y in semantic space, mapped by DSSM parameterized by $\boldsymbol{\theta}$

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



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.

Related work on semantic modeling for IR

- Document retrieval based on semantic content
  - Deal with lexicon mismatch between search queries and web documents
- Early approaches
  - Latent Semantic Analysis (LSA) and its varieties (Deerwester et al., 1990)
    - LSA extracts abstract semantic content using SVD
  - Many extensions exist: PLSA, LDA, etc.
- Recent improvements:
  - Go deeper: e.g., semantic hashing (Hinton and Salakhutdinov 2011)
  - Go beyond documents: e.g., using click signals (Gao et al. 2010; Gao et al. 2011)