Module 4

Natural Language Understanding

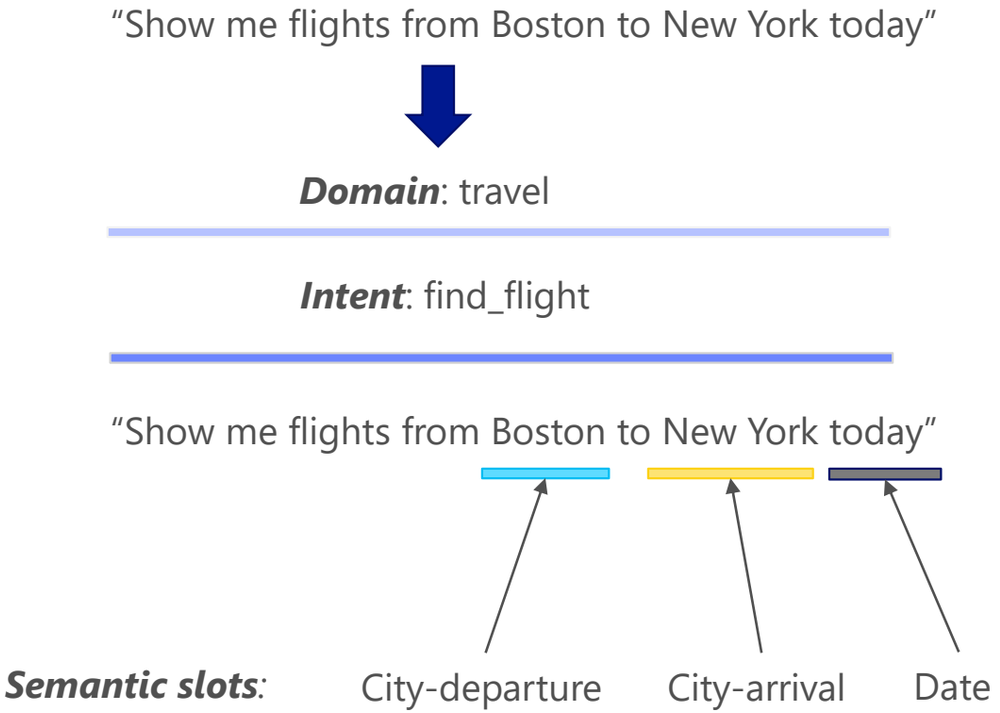
Natural Language Understanding

- Spoken Language Understanding
- Continuous Word Representations
  - Language is compositional
  - Word is the basic semantic unit
- Neural Knowledge Base Embedding
- KB-based Question Answering

Deep learning for spoken language processing

The scenarios

- Domain & intent classification
- Semantic slot filling



Semantic slot filling

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

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

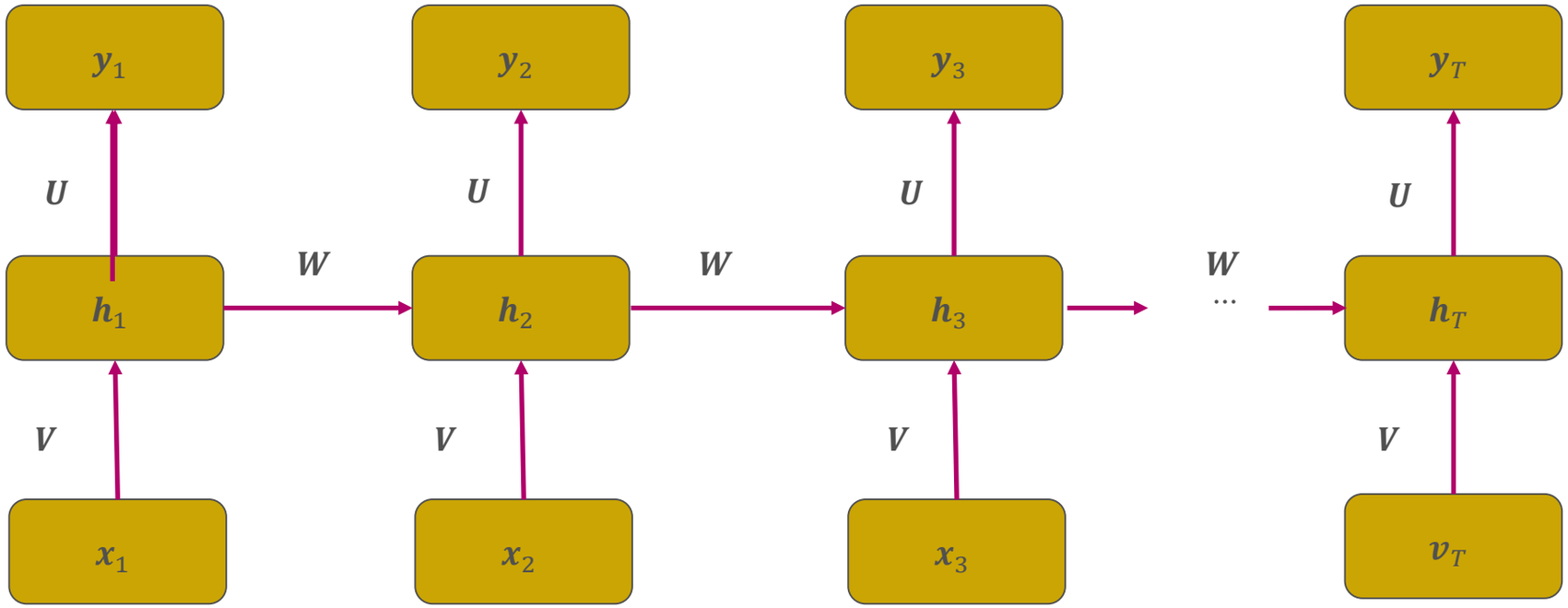
Slot filling can be viewed as a sequential tagging problem

Recurrent neural networks for slot filling

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

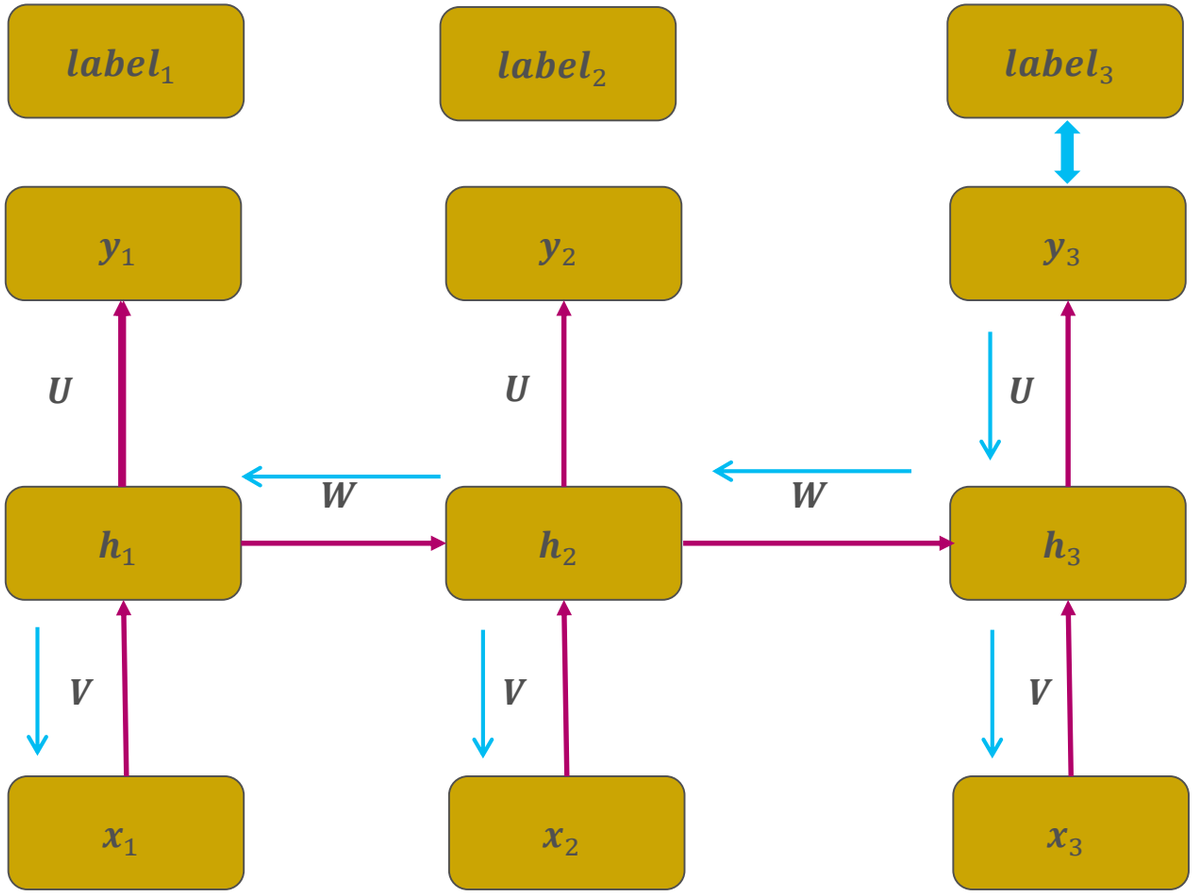
where x_t: $the\ input\ word$, y_t: $the\ output\ tag$

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



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

Back-propagation through time (BPTT)



at time $t = 3$

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

Results

- Evaluated on the ATIS corpus
  - 4978 utterances for training
  - 893 utterances for testing
  - Using word feature only
  - Baseline CRF: 92.94% in F1-measure

~25% error reduction!

SGD vs. minibatch training

With local context window

Model	Elman	Jordan	Hybrid
Stochastic GD	94.55 ±0.51	94.66 ±0.23	94.75 ±0.31
Sentence-minibatch	94.54 ±0.23	94.33 ±0.19	94.25 ±0.28

Left-to-right vs. bi-directional RNN

With local context window

Model	Elman	Jordan
Left-to-right	94.54	94.33
bi-direction	94.73	94.03

Without local context window

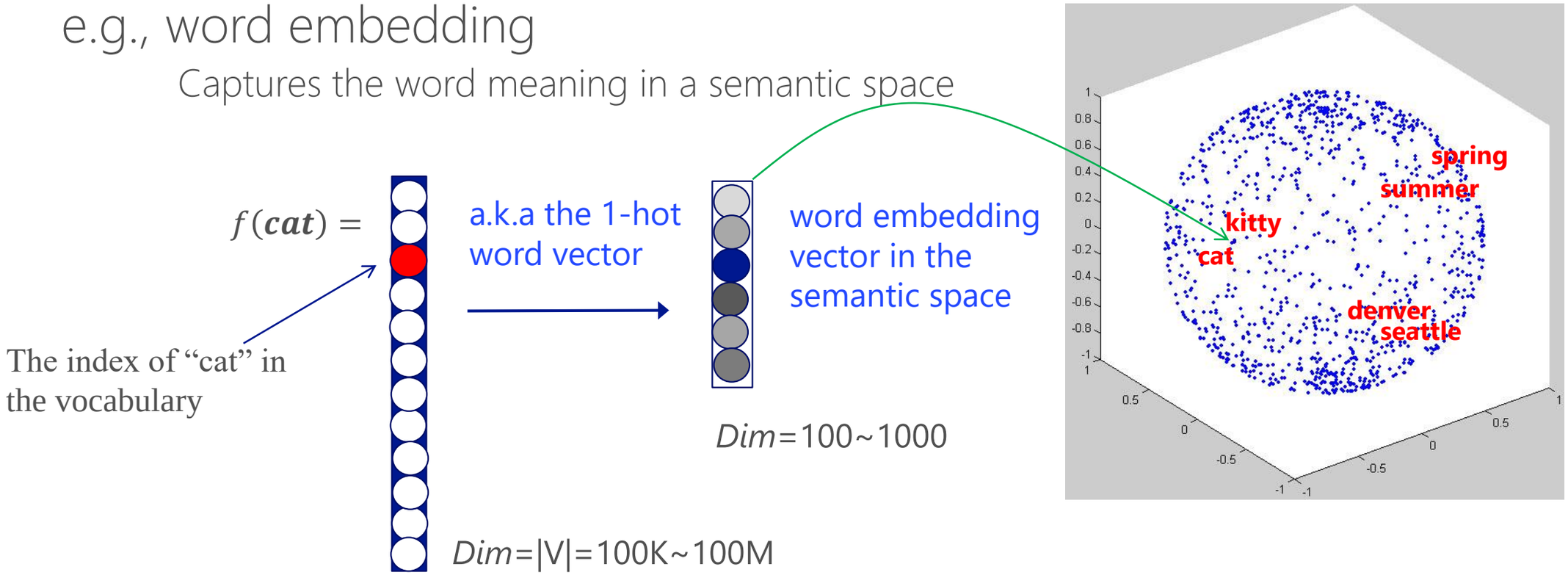
Model	Elman	Jordan
Left-to-right	93.15	65.23
bi-direction	93.46	90.31

Continuous Word Representations

Project a word into a continuous space

e.g., word embedding

Captures the word meaning in a semantic space



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

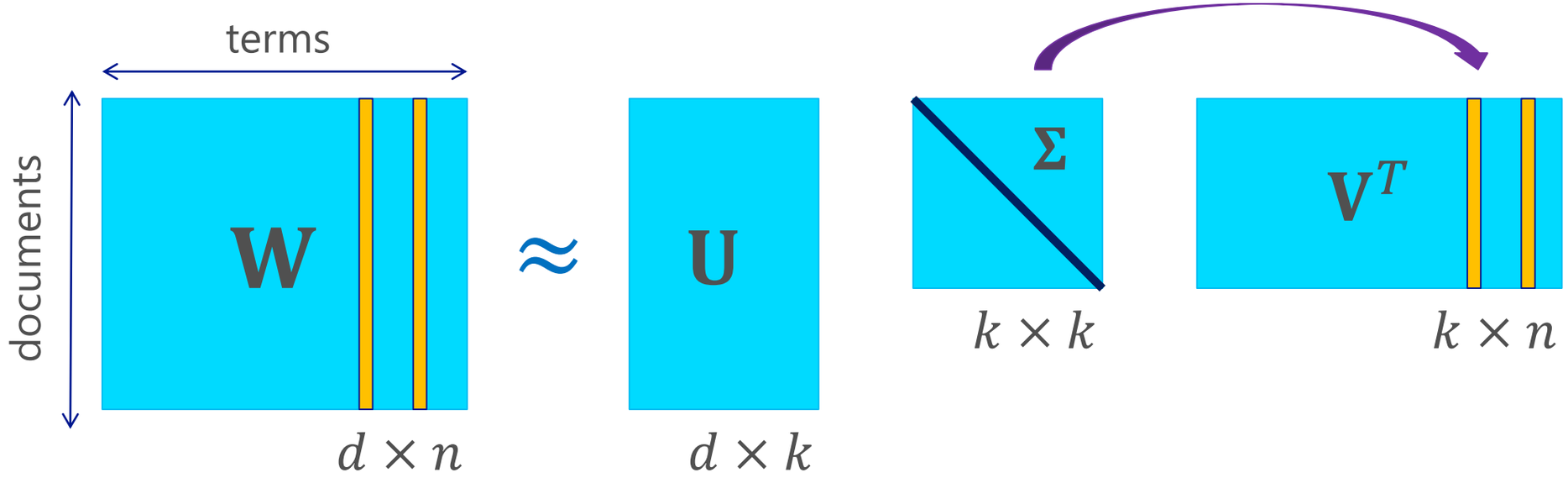
Continuous Word Representations

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

Roadmap – Continuous Word Representations

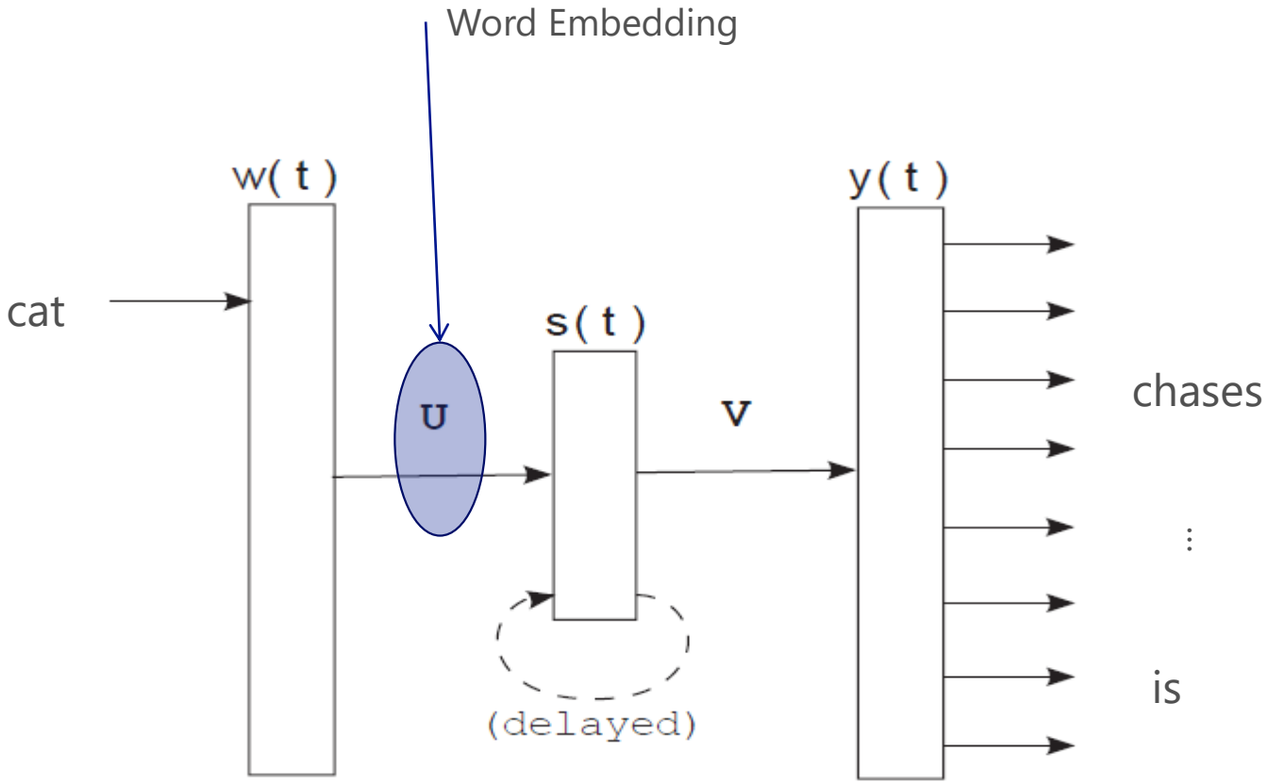
- Samples of word embedding models
  - Latent Semantic Analysis (LSA), Recurrent Neural Networks
  - SENNA, CBOW/Skip-gram, DSSM, GloVe
- Evaluation
  - Semantic word similarity
  - Relational similarity (word analogy)
- Related work
  - Model different word relations
  - Other word embedding models

Latent Semantic Analysis



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

RNN-LM Word Embedding



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

SENNA Word Embedding

Scoring:

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

Training:

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

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

Where

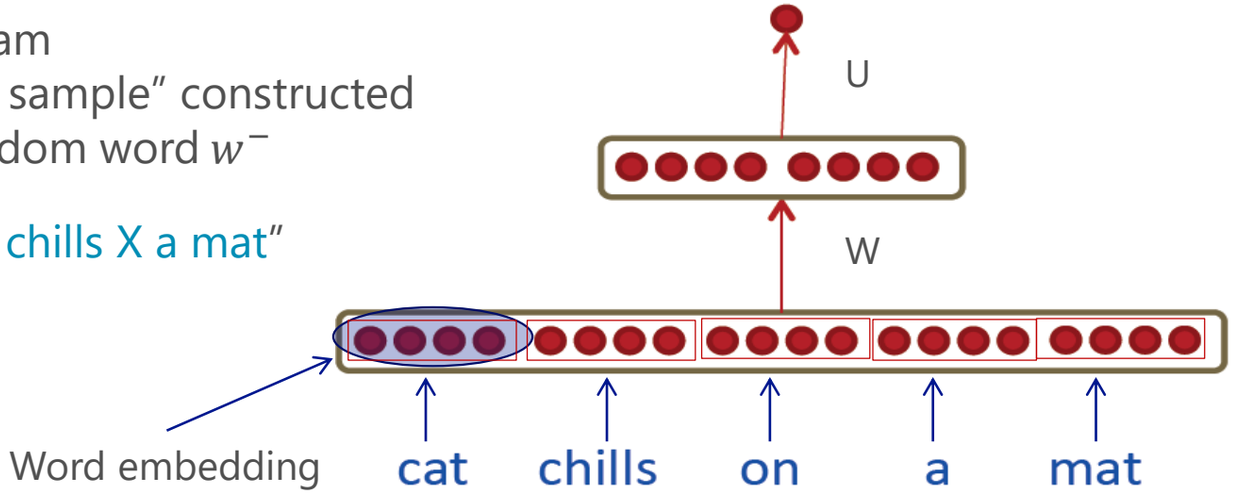
$$S^+ = Score(w_1, w_2, w_3, w_4, w_5)$$
$$S^- = Score(w_1, w_2, w^-, w_4, w_5)$$

And

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

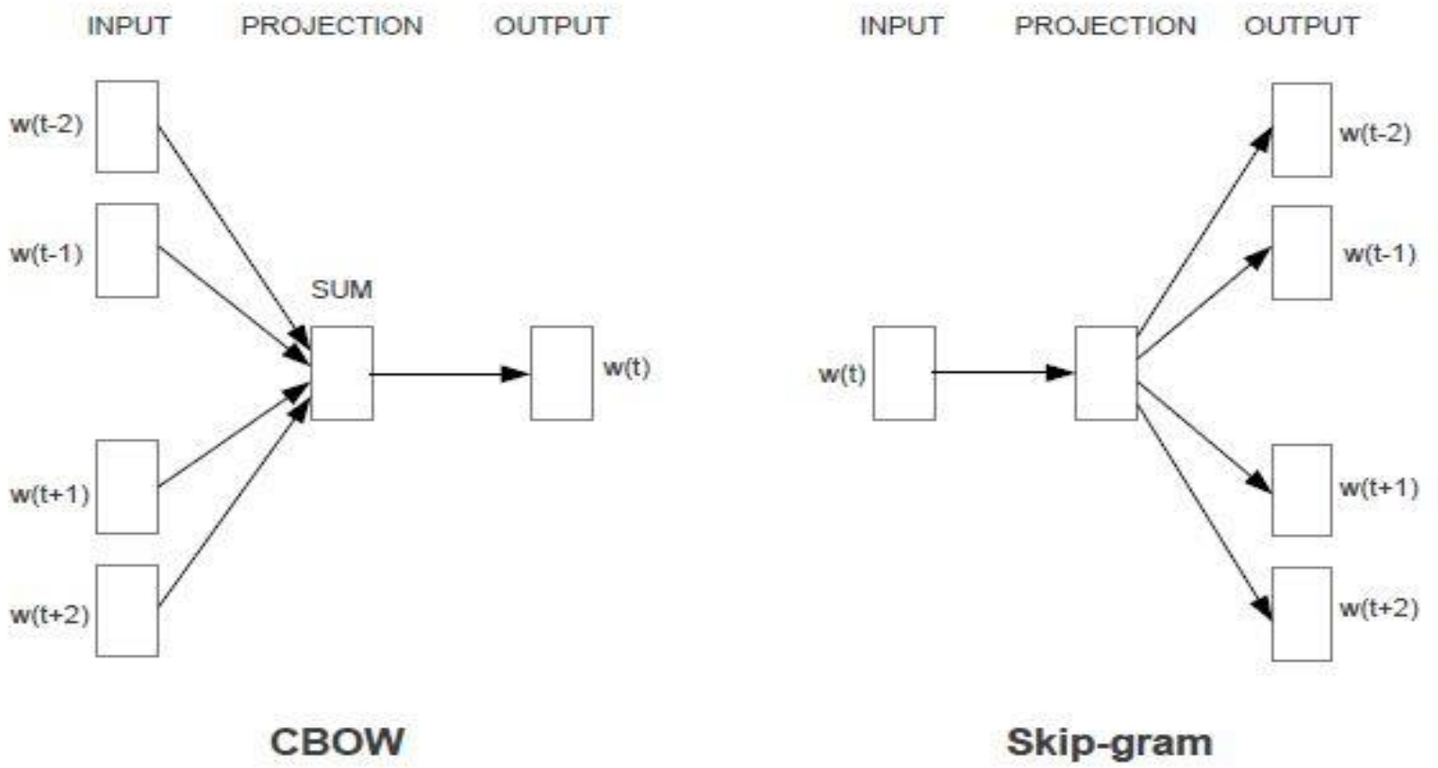
$< w_1, w_2, w^-, w_4, w_5 >$ is a "negative sample" constructed by replacing the word w_3 with a random word w^-

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



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

CBOW/Skip-gram Word Embeddings



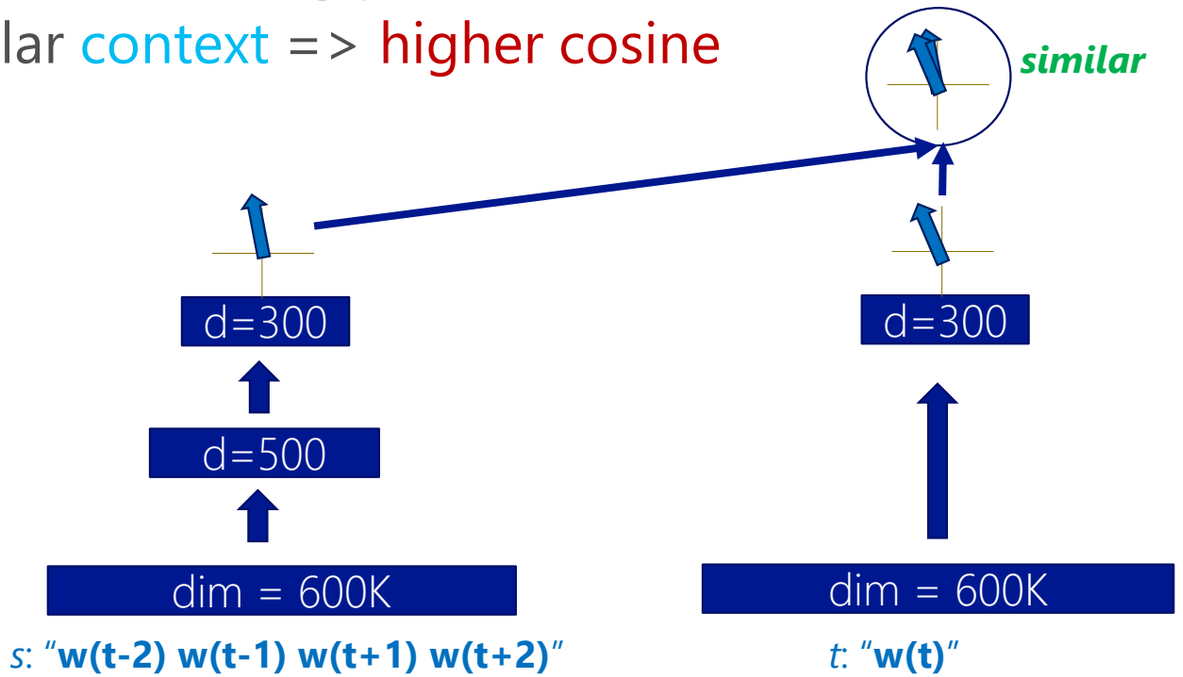
Continuous Bag-of-Words

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

DSSM: Learning Word Meaning

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

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



[Song, He, Gao, Deng, 2014]

GloVe: Global Vectors for Word Representation [Pennington+ EMNLP-14]

- Semantic relatedness can be observed from word co-occurrence counts and ratios

Context words

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

"solid" is more related to "ice"

GloVe: Global Vectors for Word Representation [Pennington+ EMNLP-14]

- Semantic relatedness can be observed from word co-occurrence counts and ratios

Context words

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

"gas" is more related to "steam"

GloVe: Global Vectors for Word Representation [Pennington+ EMNLP-14]

- Semantic relatedness can be observed from word co-occurrence counts and ratios

Context words

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Equally related or unrelated

GloVe: Global Vectors for Word Representation [Pennington+ EMNLP-14]

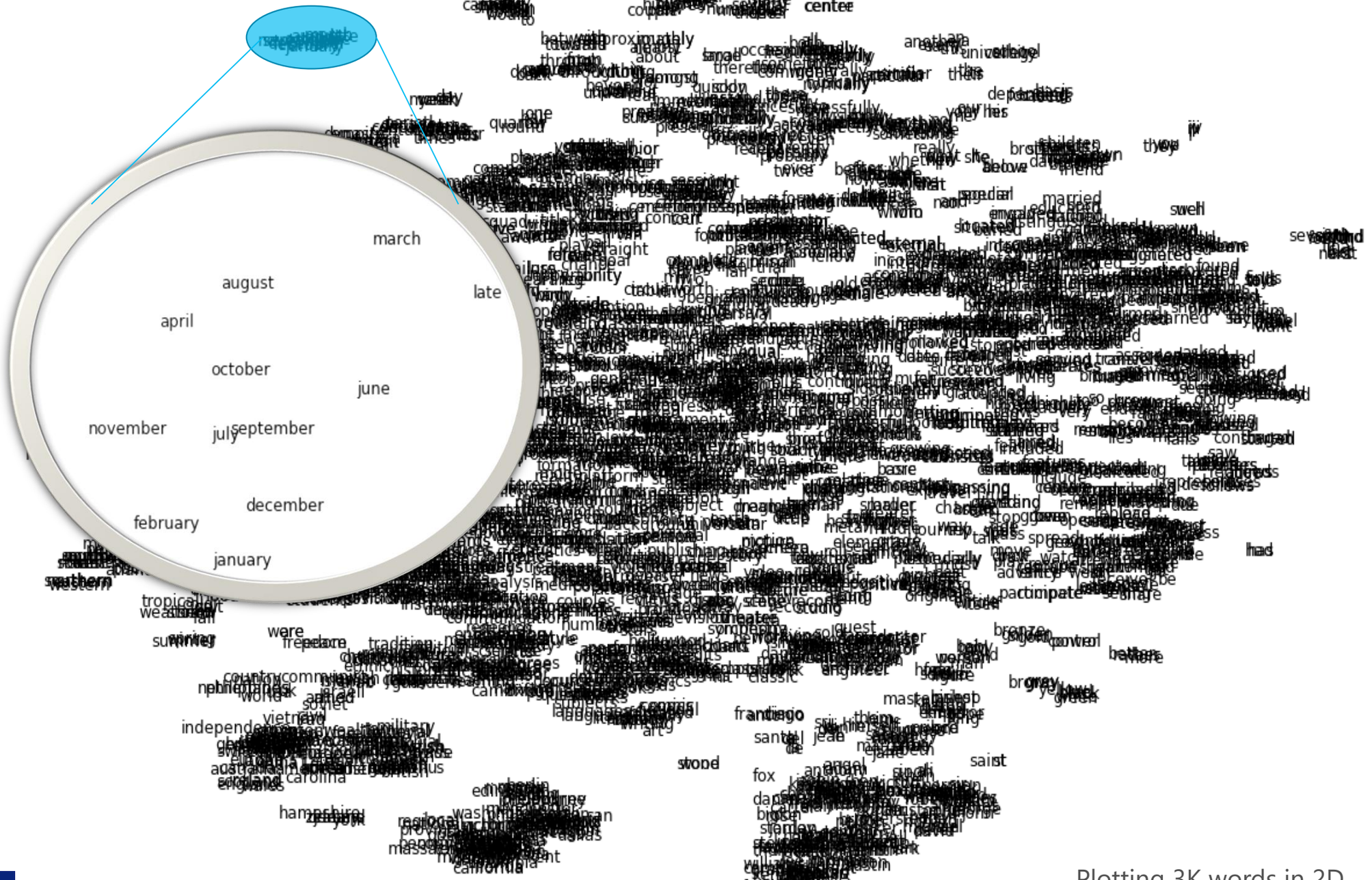
- Word embedding model design principle:

$F(w_i, w_j, \tilde{w}_k) = \frac{P(k|i)}{P(k|j)}$ (e.g., i = ice, j = steam, k = solid/gas)

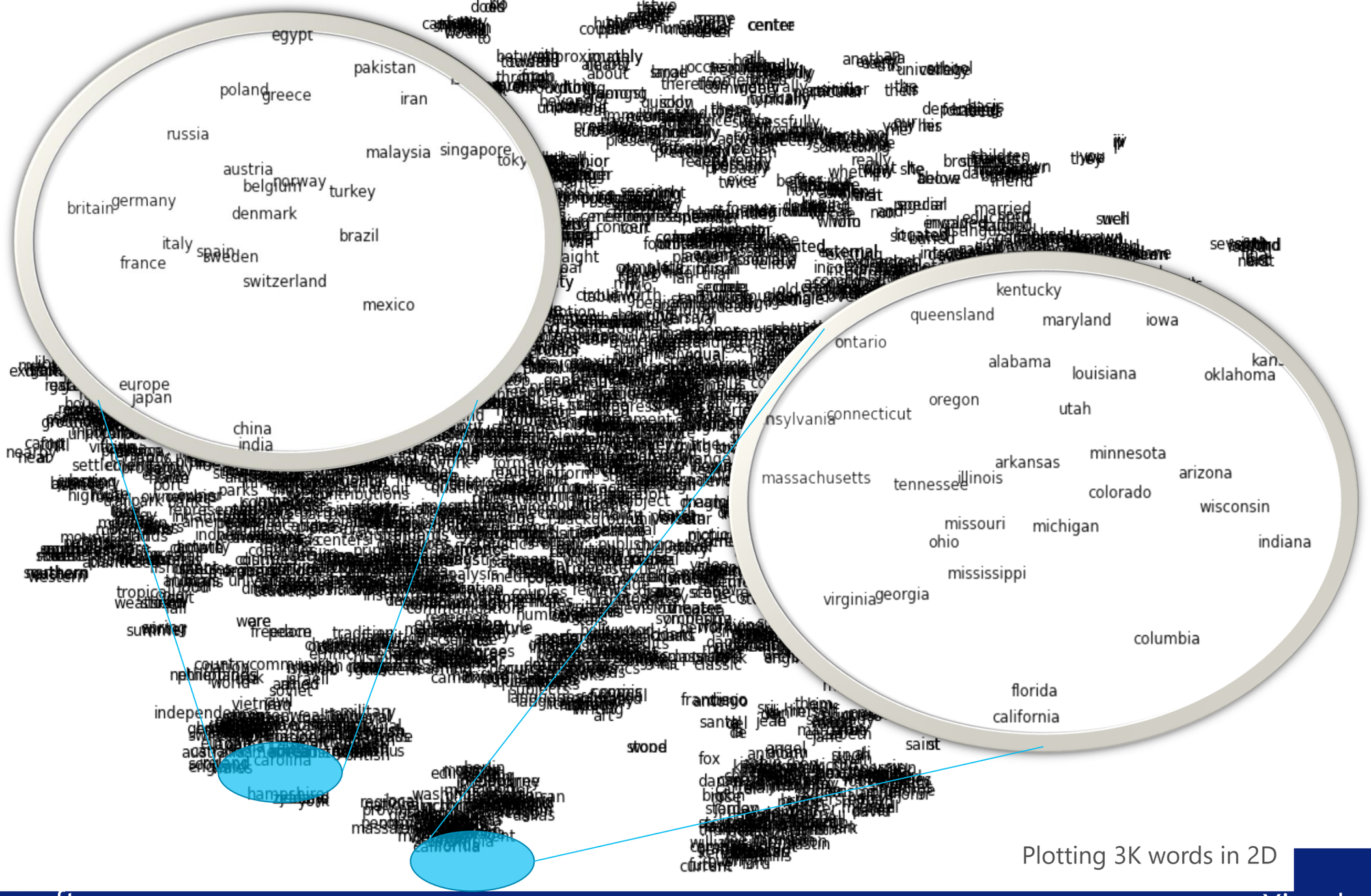
- Objective: $J = \sum_{i,j=1}^{V} f(X_{ij}) \left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij} \right)^2$

Down weight low co-occurrences

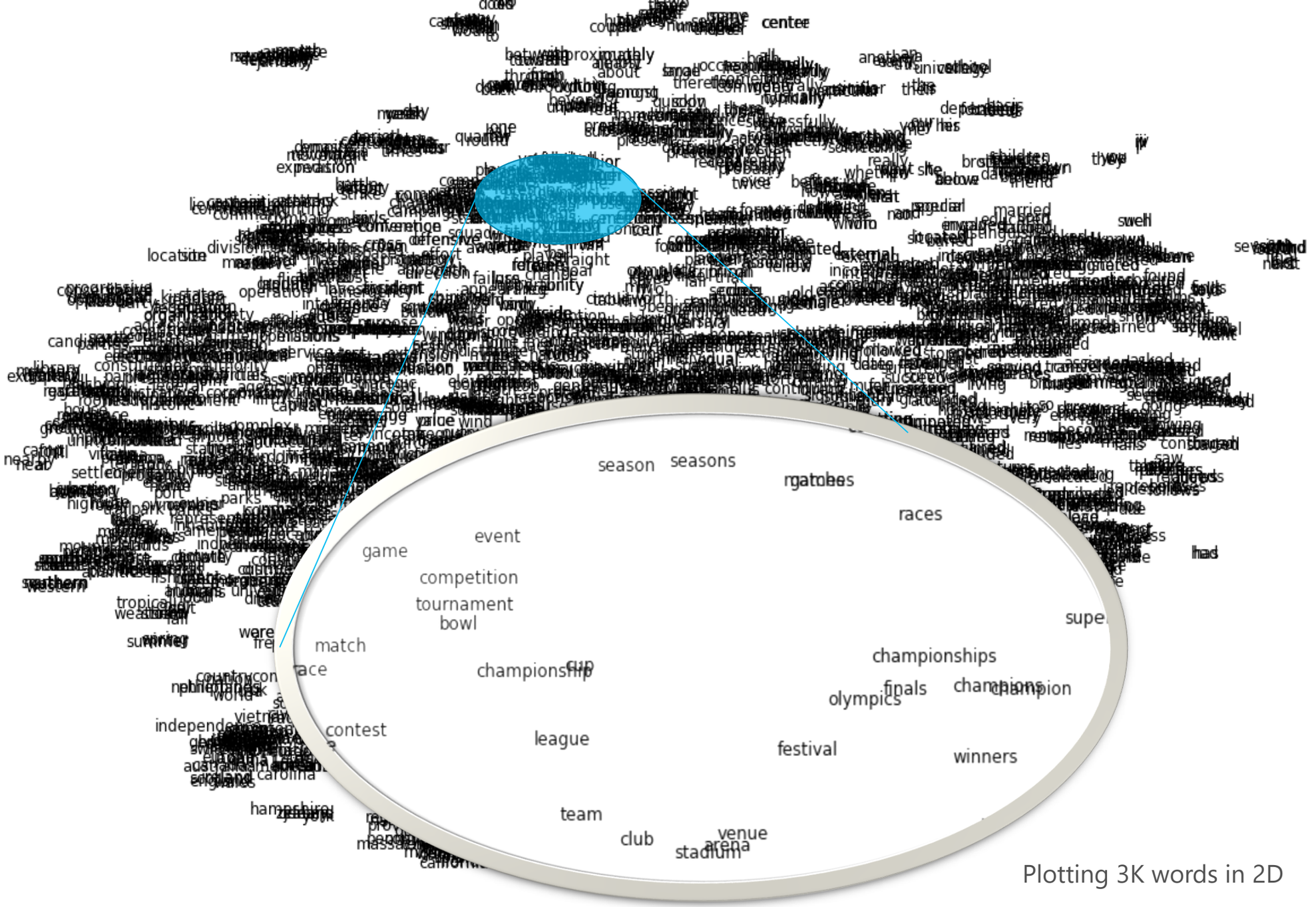
co-occurrence counts



Plotting 3K words in 2D



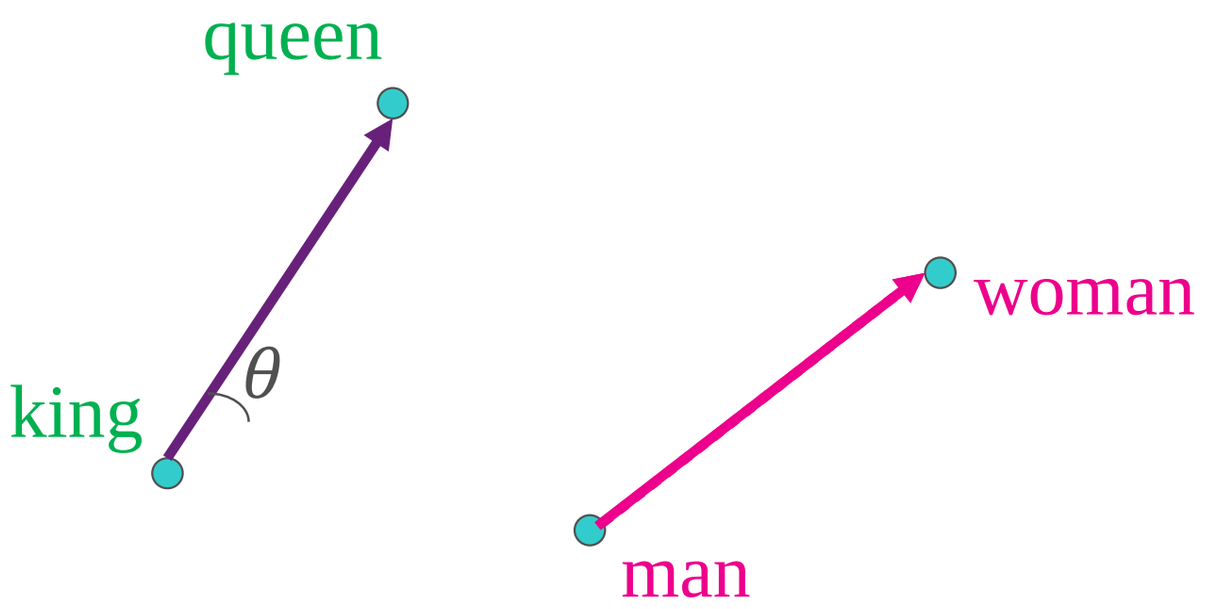
Plotting 3K words in 2D



Plotting 3K words in 2D

Unexpected Finding: Directional Similarity

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



- Relational similarity is derived by the cosine score

Similar Results Observed on Other Datasets

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

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

Evaluation on Word Analogy

The dataset contains 19,544 word analogy questions:

- Semantic questions, e.g.,: “Athens is to Greece as Berlin is to ?”
- Syntactic questions, e.g.,: “dance is to dancing as fly is to ?”

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

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

Related Work – Model Different Word Relations

Tomorrow will be rainy.

Tomorrow will be sunny.

$similar$(r[illegible], sunny)?

$antonym$(rainy, sunny)?

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

f_{rel}(●, ●) 

Related Work – Word Embedding Models

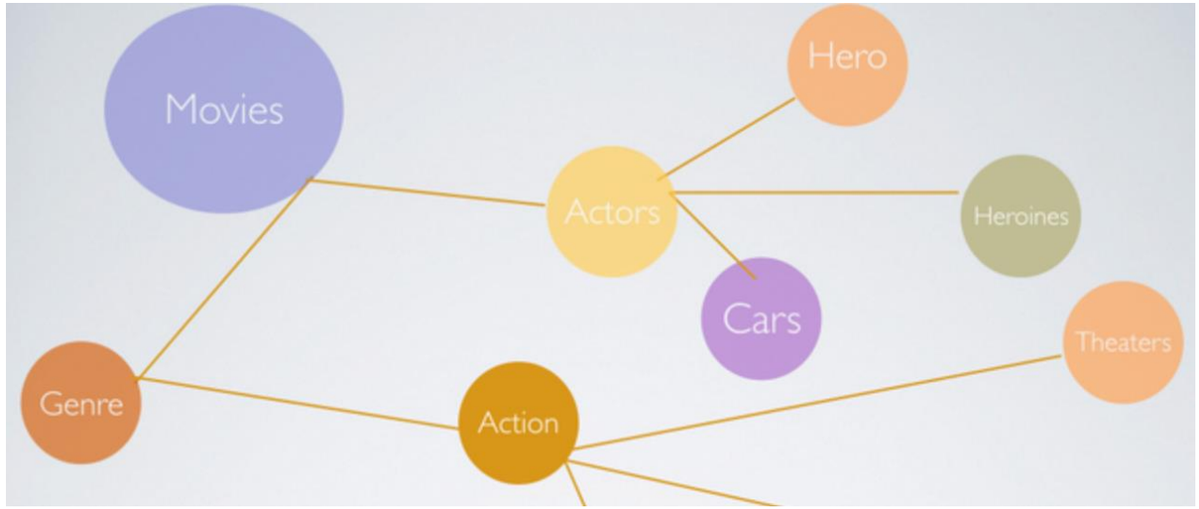
- Other word embedding models
  - [Wang+ EMNLP-14], [Bian+ ECML/PKDD-14], [Xu+, CIKM-14], [Faruqui+ NAACL-15], [Yogatama+ ICML-15], [Faruqui+ ACL-15]
- Analysis of Word2Vec and Directional Similarity
  - Linguistic Regularities in Sparse and Explicit Word Representations [Levy & Goldberg CoNLL-14]
  - Neural Word Embedding as Implicit Matrix Factorization [Levy & Goldberg NIPS-14]
- Theoretical justification and unification
  - Word Embeddings as Metric Recovery in Semantic Spaces [Hashimoto+ TACL-16]
- New Evaluation: RelEval@ACL-16 – Evaluating Vector Space Representations for NLP

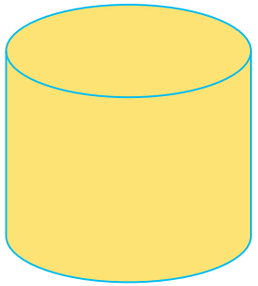
Natural Language Understanding

- Knowledge Base Embedding
  - Nickel et al., “A Review of Relational Machine Learning for Knowledge Graphs”

Knowledge Base / Knowledge Graph

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





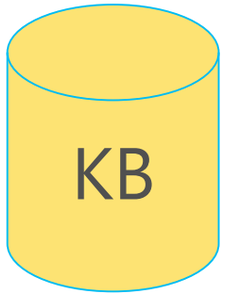
Freebase
DBpedia
YAGO
NELL
OpenIE/ReVerb

Current KB Applications in NLP & IR

- Question Answering

  "*What are the names of Mike's daughters?*"

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



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

  $bornIn(Henry, Brooklyn)$

  $contains(New\ York, Brooklyn)$

- Web Search
  - Identify entities and relationships in queries

Reasoning with Knowledge Base

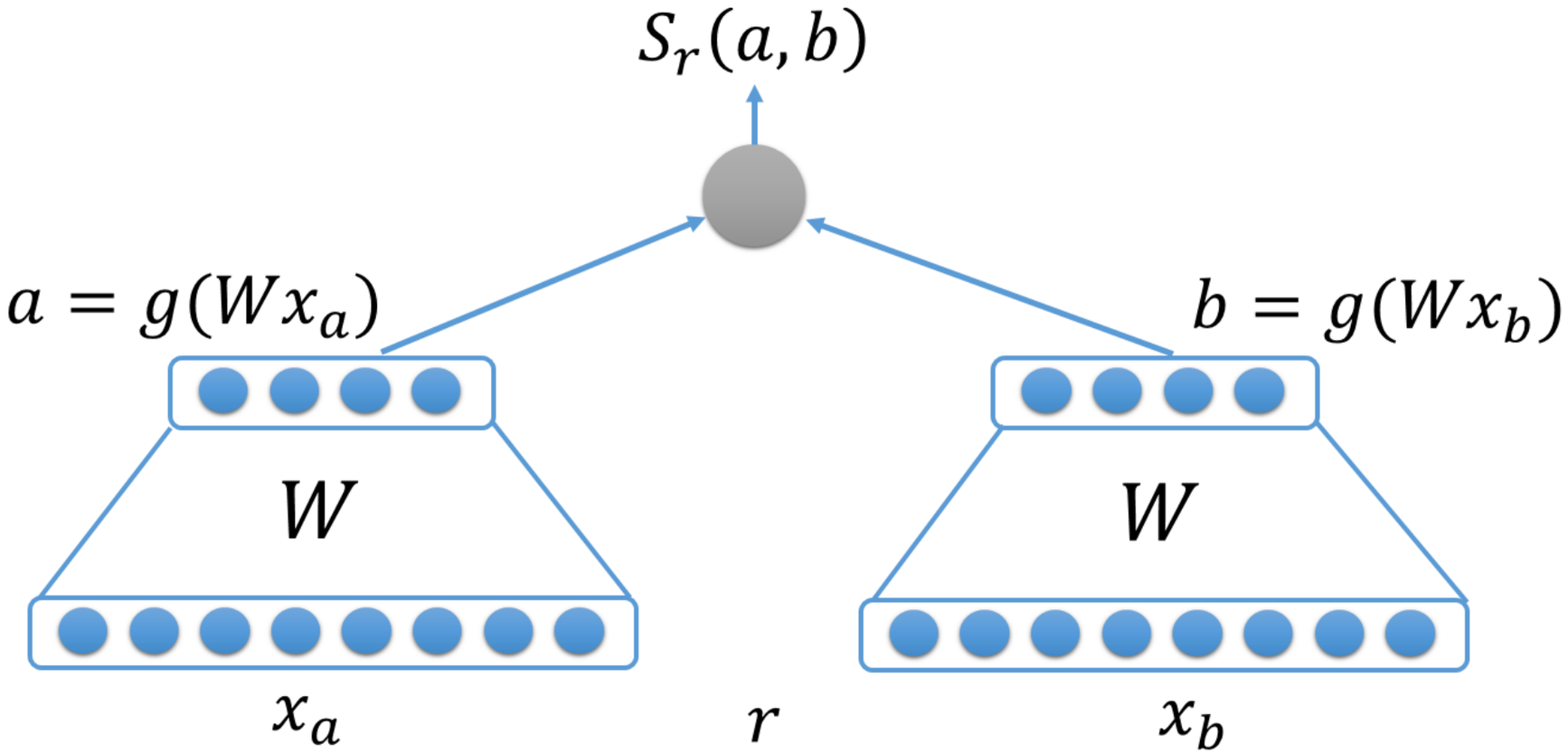
- Knowledge base is never complete!
  - Predict new facts: $Nationality(Firstname\ Lastname, ?)$
  - Mine rules: $BornInCity(a, b) \wedge CityInCountry(b, c) \Rightarrow Nationality(a, c)$
- Modeling multi-relational data
  - Statistical relational learning [Getoor & Taskar, 2007]
  - Path ranking methods (e.g., random walk) [e.g., Lao+ 2011]
  - Knowledge base embedding
    - Very efficient
    - Better prediction accuracy

Knowledge Base Embedding

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

- Recent neural network based KB embedding
  - SME [Bordes+, AISTATS-12], NTN [Socher+, NIPS-13], TransE [Bordes+, NIPS-13], Bilinear-Diag [Yang+, ICLR2015]

Neural Knowledge Base Embedding



Relation Operators

Relation representation	Scoring Function $S_r(a, b)$	# Parameters
Vector (TransE) (Bordes+ 2013)	$\|\|a - b + V_r\|\|_{1,2}$	$O(n_r \times k)$
Matrix (Bilinear) (Bordes+ 2012, Collobert & Weston 2008)	$a^T M_r b$ $u^T f(M_{r1}a + M_{r2}b)$	$O(n_r \times k^2)$
Tensor (NTN) (Socher+ 2013)	$u^T f(a^T T_r b + M_{r1}a + M_{r2}b)$	$O(n_r \times k^2 \times d)$
Diagonal Matrix (Bilinear-Diag) (Yang+ 2015)	$a^T diag(M_r) b$	$O(n_r \times k)$

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

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

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

Mining Horn-clause Rules [Yang+ ICLR-2015]

- Can relation embedding capture relation composition?
  $$BornInCity(a, b) \wedge CityInCountry(b, c) \Rightarrow Nationality(a, c)$$

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

Learning from Relational Paths [Guu+ EMNLP-15, Garcia-Duran+ EMNLP-15, Toutanova+ ACL-16]

- Single-edge path: $\mathbf{score}(s, r, t) = v_s^{\mathrm{T}} M_r v_t$
  - (Mike, Nationality, USA)
- Multi-edge path: $\mathbf{score}(s, r_1, \cdots, r_k, t) = v_s^{\mathrm{T}} M_{r_1} \cdots M_{r_k} v_t$
  - (Mike, BornInCity, CityInCountry, USA)

Natural Language Understanding

- Question Answering

Who is Justin Bieber's sister?

Jazmyn Bieber

semantic parsing

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

matching

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

query

Knowledge Base

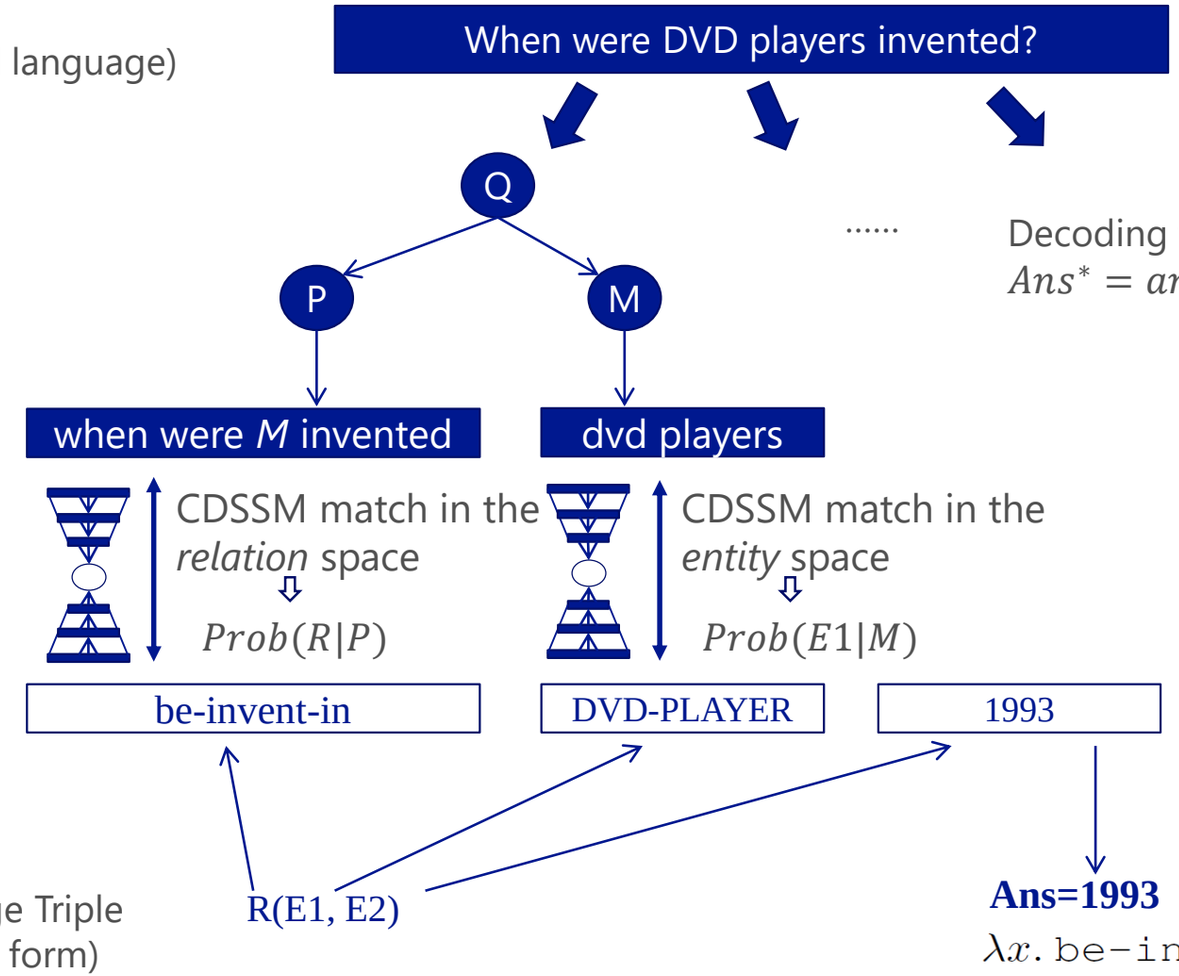
Key Challenge – Language Mismatch

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

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

DSSM in question answering

Question
(in natural language)



Knowledge Triple
(in logical form)

Decoding the best answer:

$$Ans^* = argmax_{Ans} P(Ans|KB, Q)$$

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

Yih, He, Meek, "Semantic parsing for single-relation question answering," ACL 2014

Experiments: Data

Paralex dataset [Fader et al., 2013]

- 1.8M (question, single-relation queries)

  When were DVD players invented?
  $\lambda x.\,\mathrm{be{-}invent{-}in}(\mathrm{dvd{-}player}, x)$

- 1.2M (relation pattern, relation)

  When were X invented?
  $\mathrm{be{-}invent{-}in}_2$

- 160k (mention, entity)

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

Experiments: Task – Question Answering

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

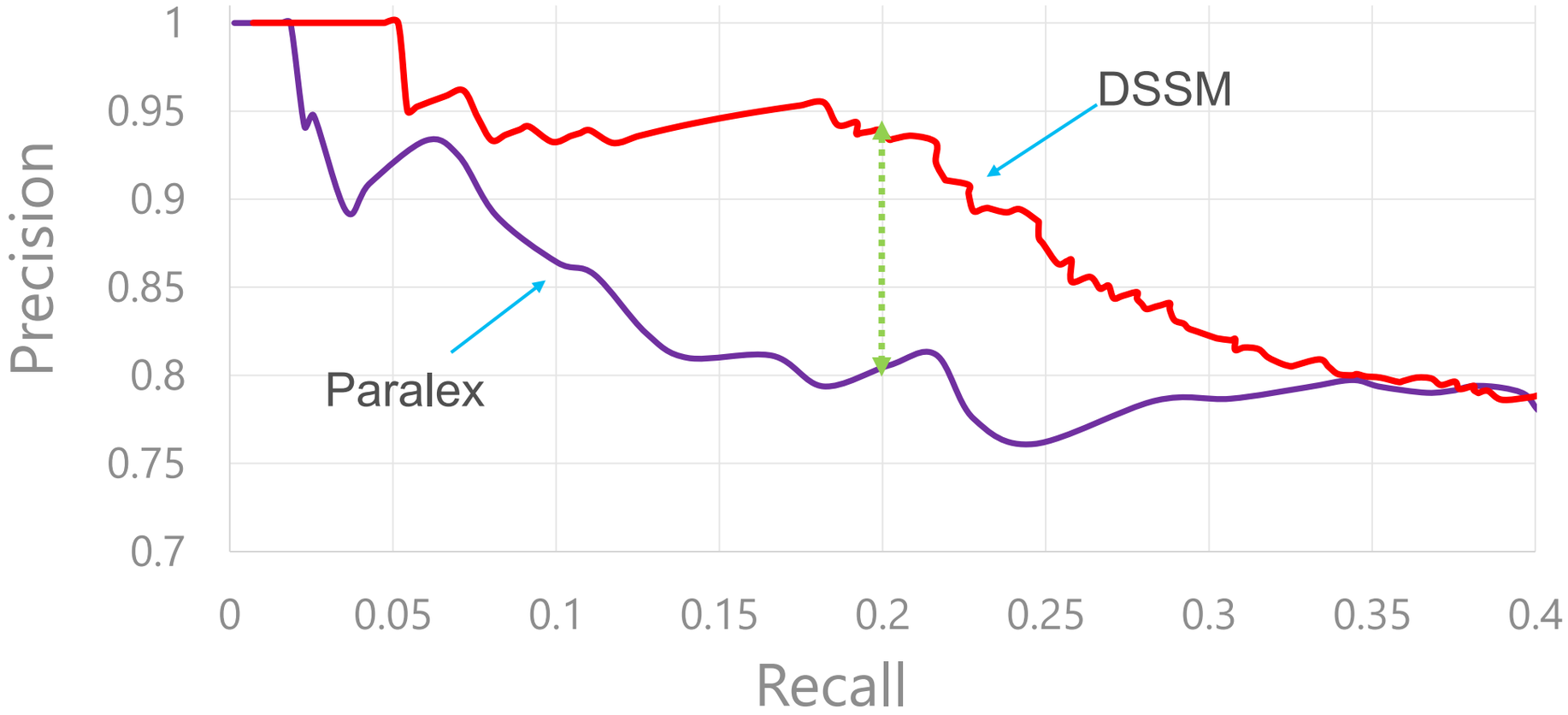
- *What language do people in Hong Kong use?*

  be–speak–in(english, hong–kong)
  be–predominant–language–in
  (cantonese, hong–kong)

- *Where do you find Mt Ararat?*

  be–highest–mountain–in(ararat, turkey)
  be–mountain–in(ararat, armenia)

Experiments: Results



Answering more complicated questions

WebQuestions Dataset [Berant+ EMNLP-2013]

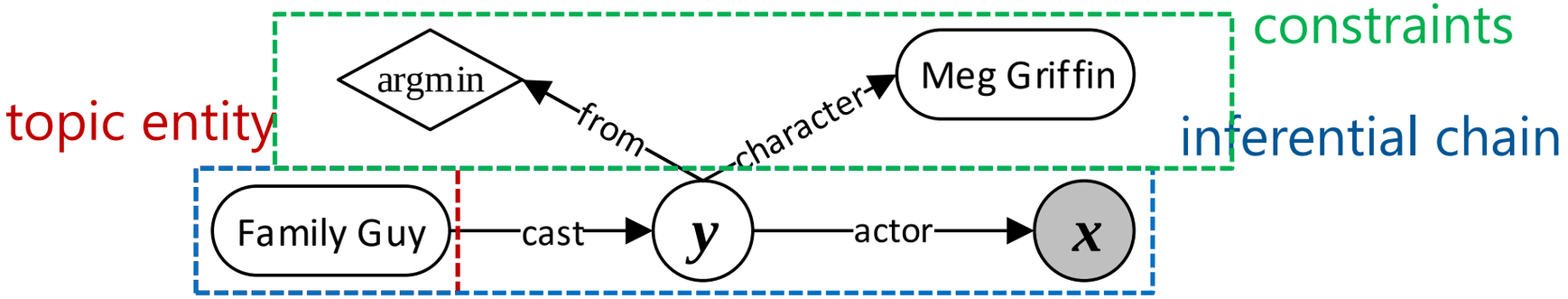
- *What character did Natalie Portman play in Star Wars?* ⇒ Padme Amidala
- *What kind of money to take to Bahamas?* ⇒ Bahamian dollar
- *What currency do you use in Costa Rica?* ⇒ Costa Rican colon
- *What did Obama study in school?* ⇒ political science
- *What do Michelle Obama do for a living?* ⇒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

[Examples from Berant]

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

Staged Query Graph Generation [Yih+ ACL-15]

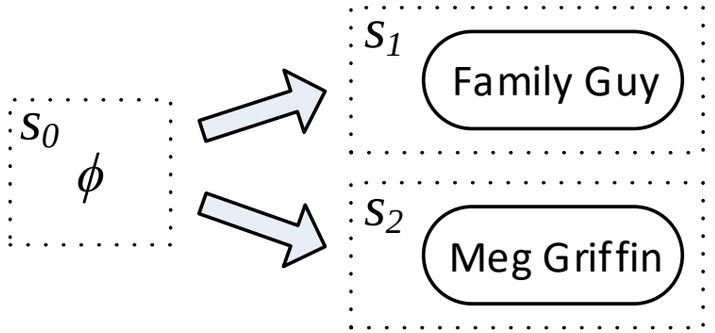
- Query graph
  - Resembles subgraphs of the knowledge base
  - Can be directly mapped to a logical form in λ-calculus
  - Semantic parsing: a search problem that *grows* the graph through actions
- Who first voiced Meg on Family Guy?
- $\lambda x.\exists y.\mathrm{cast}(\mathrm{FamilyGuy}, y) \wedge \mathrm{actor}(y, x) \wedge \mathrm{character}(y, \mathrm{MegGriffin})$

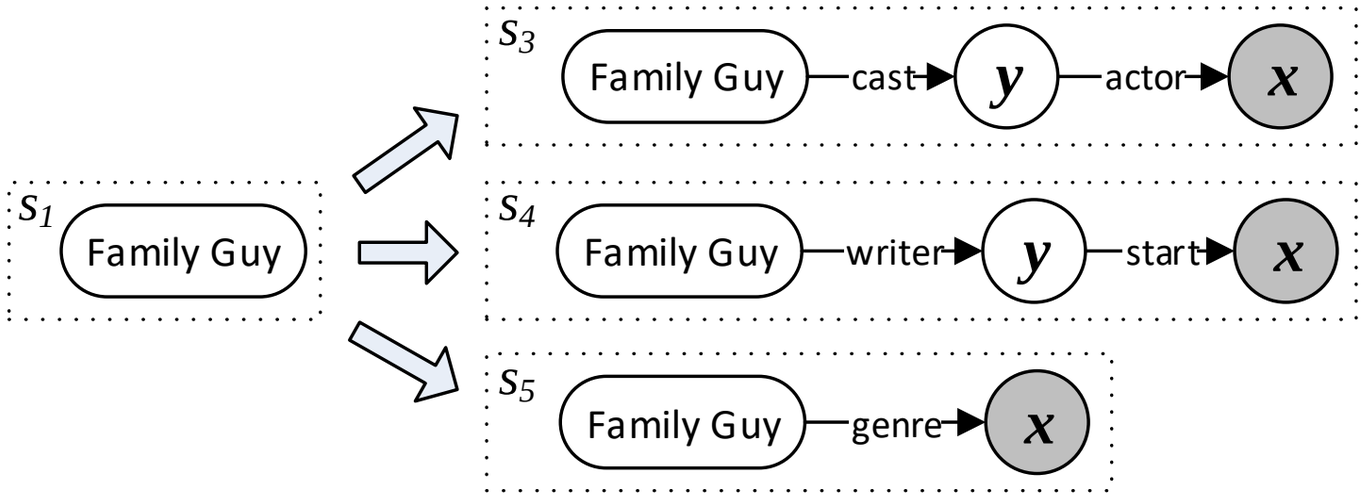


Graph Generation Stages

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

1. Topic Entity Linking [Yang&Chang ACL-15]
2. Identify the core inferential chain

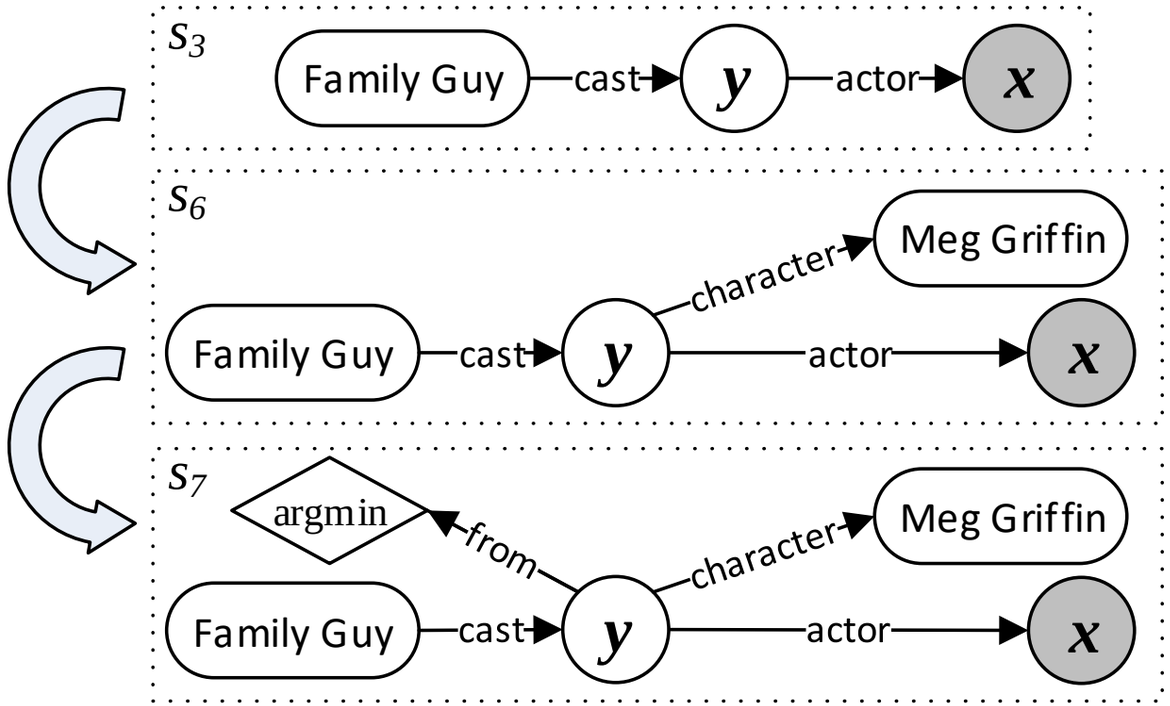




Graph Generation Stages (cont'd)

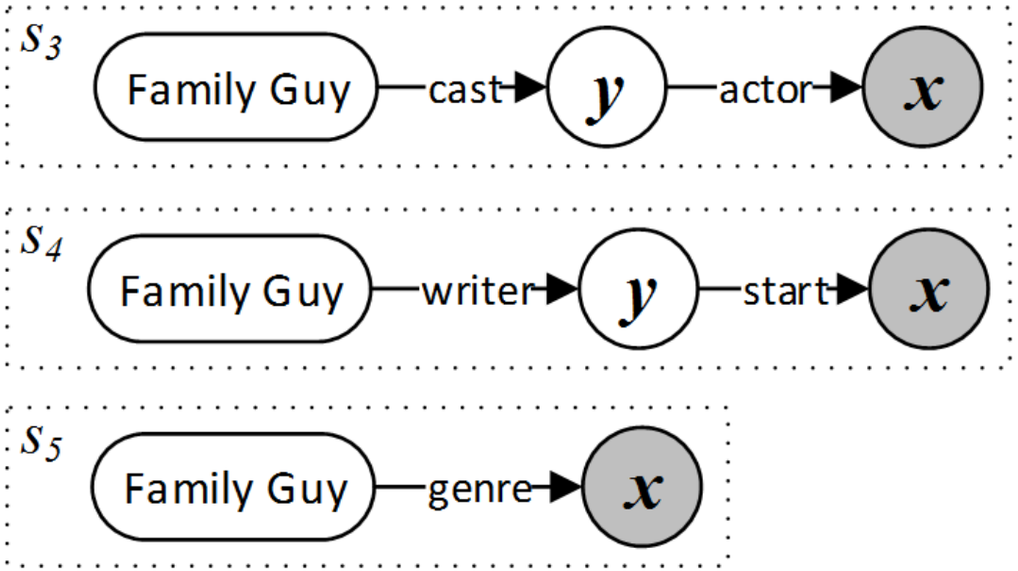
- Who first voiced Meg on Family Guy?

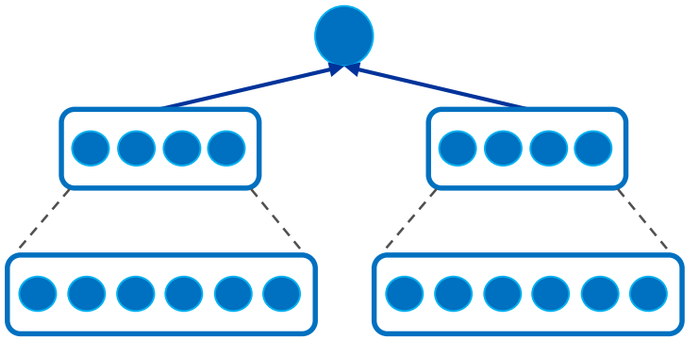
3. Augment constraints



Identify Inferential Chain using DSSM

- Who first voiced Meg on Family Guy?

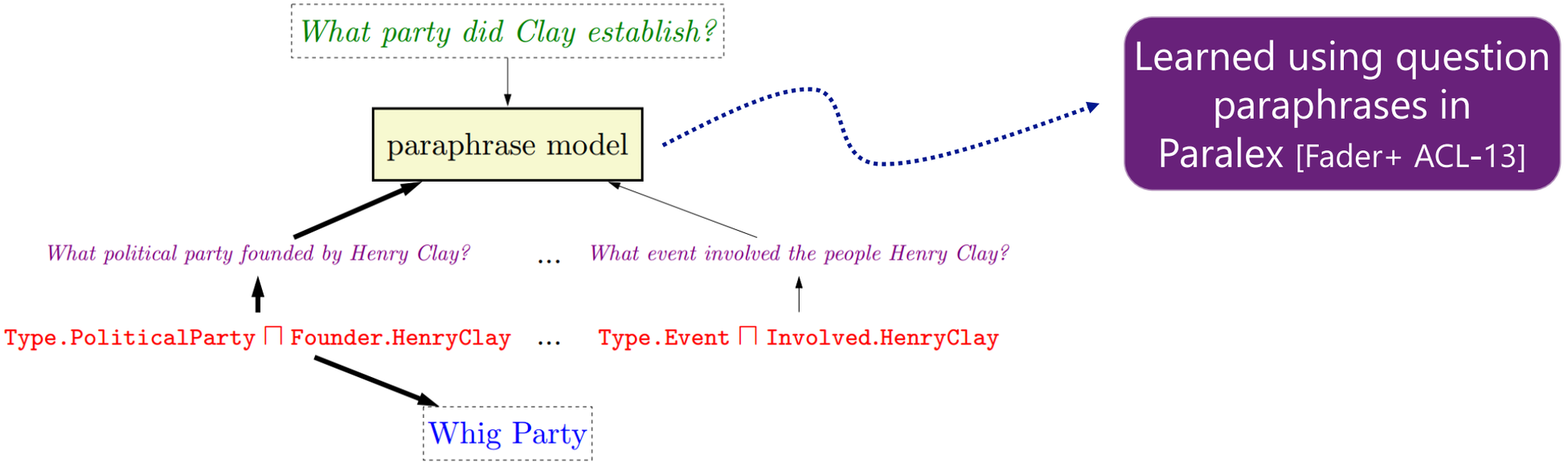




- Semantic match (“Who first voiced Meg on ⟨e⟩”, “cast-actor”)
- Single pattern/relation matching model: 49.6% F_1 (vs. 52.5% F_1 Full)

Matching Questions

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



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

Subgraph Embedding [Bordes+ EMNLP-2014]

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

Avg. F1 (Accuracy) on WebQuestions Test Set