Understanding Short Texts
CCKS 2016 Tutorial

Zhongyuan Wang (Microsoft Research)

Haixun Wang (Facebook Inc.)

Tutorial Website:
http://www.wangzhongyuan.com/tutorial/ACL2016/Understanding-Short-Texts/

Outline

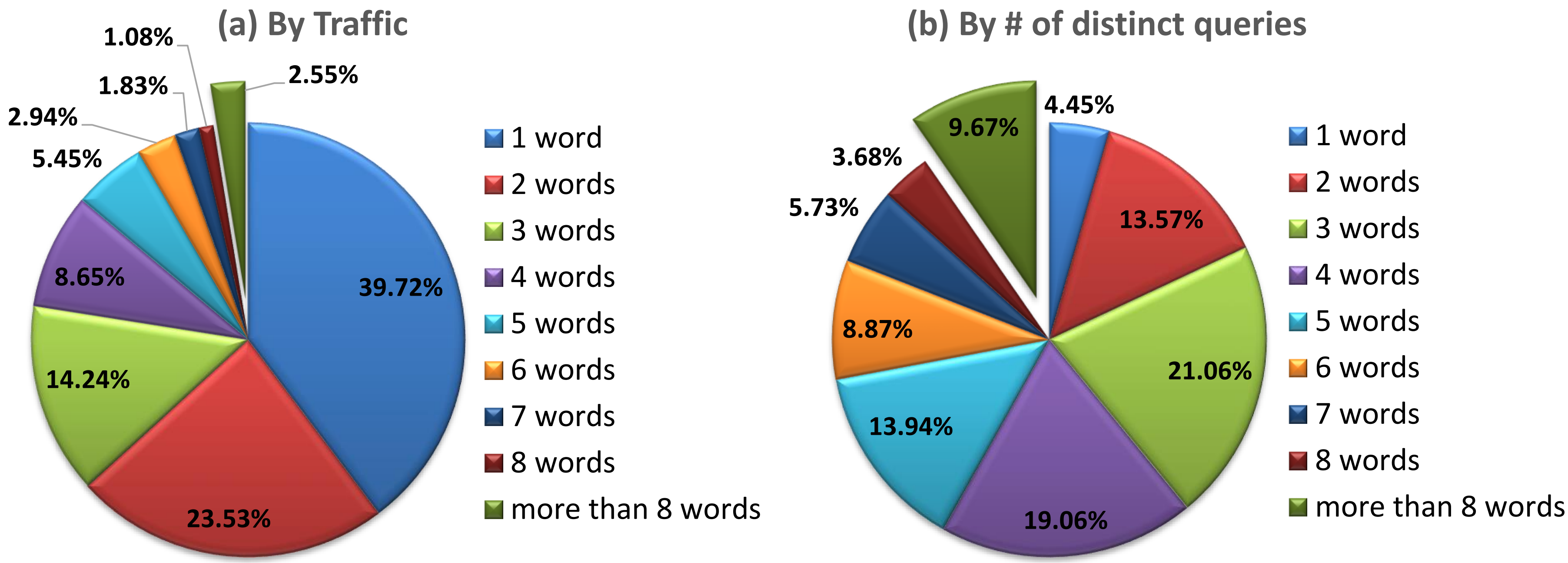
- Part 1: Challenges
- Part 2: Explicit representation
- Part 3. Implicit representation
- Part 4: Conclusion

Short Text

- Search Query
- Ad keyword
- Anchor text
- Image Tag
- Document Title
- Caption
- Question
- Tweet/Weibo

Challenges

- *First, short texts contain limited context*



Based on Bing query log between 06/01/2016 and 06/30/2016

Challenges

- *Second, "telegraphic": no word order, no function words, no capitalization, …*

*Query "*Distance between Sun and Earth*" can also be expressed as:*

- "how far" earth sun
- "how far" sun
- "how far" sun earth
- average distance earth sun
- average distance from earth to sun
- average distance from the earth to the sun
- distance between earth & sun
- distance between earth and sun
- distance between earth and the sun
- distance from earth to the sun
- distance from sun to earth
- distance from sun to the earth
- distance from the earth to the sun
- distance from the sun to earth
- distance from the sun to the earth
- distance of earth from sun
- distance between earth sun
- how far away is the sun from earth
- how far away is the sun from the earth
- how far earth from sun
- how far earth is from the sun
- how far from earth is the sun
- how far from earth to sun
- how far from the earth to the sun
- distance between sun and earth

Challenges

- *Second, "telegraphic": no word order, no function words, no capitalization, …*

Short Text 1	Short Text 2	Term Match	Semantic Match
china kong *(actor)*	china hong kong	partial	no
hot dog	dog hot	yes	no
the big apple tour	new york tour	almost no	yes
Berlin	Germany capital	no	Yes
DNN tool	deep neural network tool	almost no	Yes
wedding band	band for wedding	partial	no
why are windows so expensive	why are macs so expensive	partial	no

Challenges

- *Sparse, noisy, ambiguous*

watch for kids

i) 

ii) 

iii) It’s not a fair trade!!

Short Text Understanding

- Many applications
  - Search engines
  - Automatic question answering
  - Online advertising
  - Recommendation systems
  - Conversational bot
  - …

- Traditional NLP approaches not sufficient

The big question

- Humans are much powerful than machines in understanding short texts.
- Our minds build rich models of the world and make strong generalizations from input data that is *sparse, noisy, and ambiguous* – in many ways far too limited to support the inferences we make.
- How do we do it?

If the mind goes beyond the data given, *another source of information* must make up the difference.

Science **331**, 1279 (2011);

How to Grow a Mind: Statistics, Structure, and Abstraction

Joshua B. Tenenbaum,[1*] Charles Kemp,[2] Thomas L. Griffiths,[3] Noah D. Goodman[4]

Explicit
(Logic)
Representation

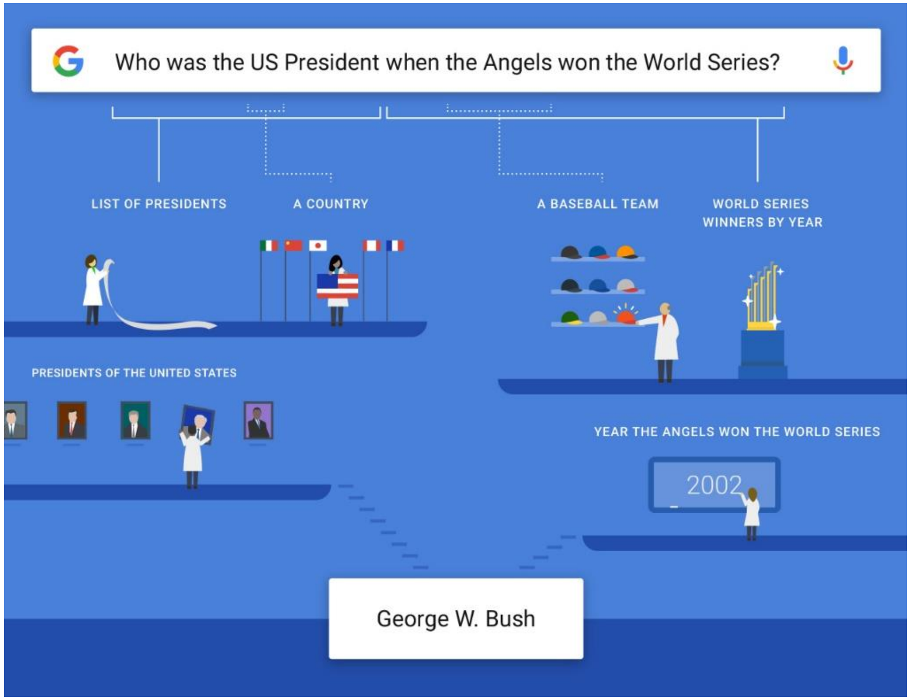
How?

Implicit
(Embedding)
Representation

Symbolic knowledge
(**Explicit**)

Distributional semantics
(**Implicit**)

Explicit Knowledge Representation

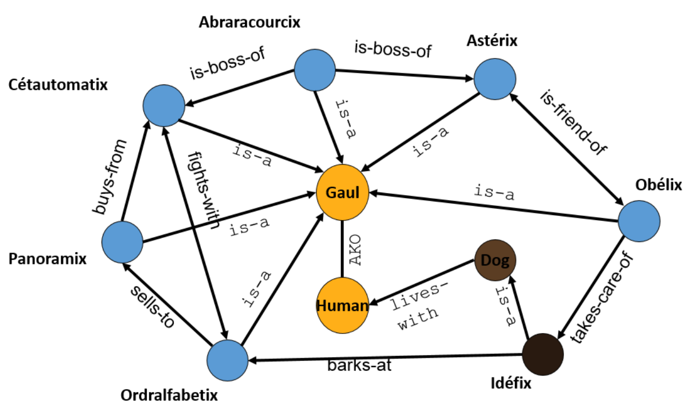


- First, understand **superlatives**—"tallest," "largest," etc.—and ordered items. So you can ask:
  "Who are the tallest Mavericks players? "
  "What are the largest cities in Texas? "
  "What are the largest cities in Iowa by area? "
- Second, have you ever wondered about a **particular point in time**? Google now do a much better job of understanding questions with dates in them. So you can ask:
  "What was the population of Singapore in 1965?"
  "What songs did Taylor Swift record in 2014?"
  "What was the Royals roster in 2013?"
- Finally, Google starts to understand some **complex combinations**. So Google can now respond to questions like:
  "What are some of Seth Gabel's father-in-law's movies?"
  "What was the U.S. population when Bernie Sanders was born?"
  "Who was the U.S. President when the Angels won the World Series?"

http://insidesearch.blogspot.com/2015/11/the-google-app-now-understands-you.html

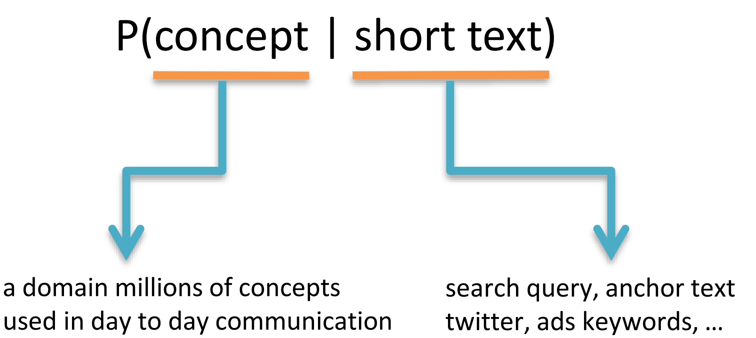
Explicit Knowledge Representation

- Logic Representation (First-order-logic)
  - Freebase, Google knowledge Graph…



True or False

- Vector Representation
  - ESA: Mapping text to Wikipedia article titles
  - Conceptualization: Mapping text to concept space



Probabilistic Model

Explicit Knowledge Representation

- Logic Representation (First-order-logic)
  - Freeb… G… l… kno…

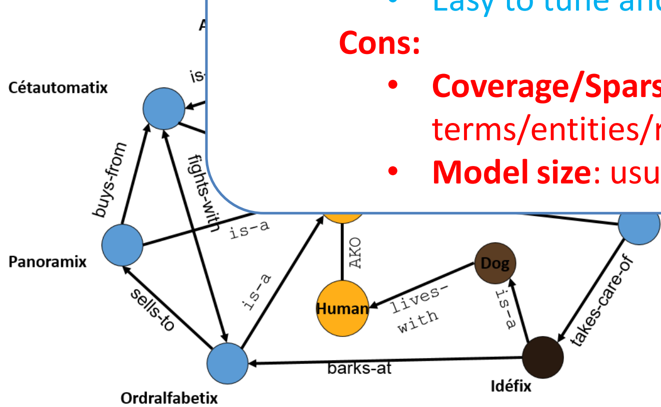
- Vector Representation
  - ESA: Mapping text to Wikipedia article titles
  - …apping text
  - …)

Pros:

- The results are easy to understand for human beings
- Easy to tune and customize

Cons:

- **Coverage/Sparse model**: can't handle unseen terms/entities/relations
- **Model size**: usually very large



a domain millions of concepts used in day to day communication

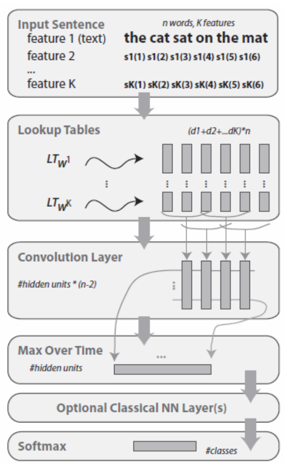
search query, anchor text twitter, ads keywords, …

True or False

Probabilistic Model

Implicit Knowledge Representation: Embedding

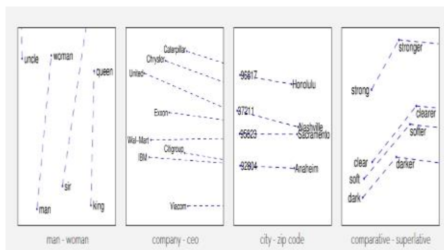
CW08



Input units: word
Vocabulary: 130k

Collobert, Ronan, et al. "Natural language processing (almost) from scratch." *The Journal of Machine Learning Research* 12 (2011): 2493-2537.

GloVe



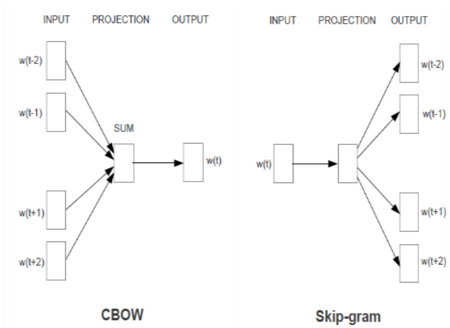
Input units: word
Training size: > 42B tokens
Vocabulary: > 400K

J Pennington, R Socher, CD Manning "Glove: Global Vectors for Word Representation." EMNLP 2014.

Google word2vec

Tool for computing continuous distributed representations of words.

https://code.google.com/p/word2vec/

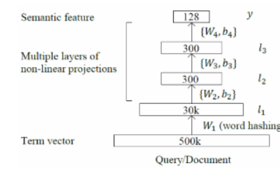


Input units: word
Training size: > 100B sequence (Freebase)
Vocabulary: > 2M

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.

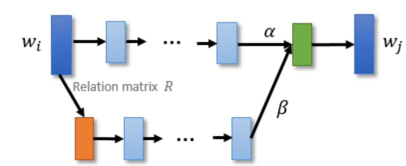
Microsoft

Deep Structured Semantic Model (DSSM)



Input units: Tri-letter
Training size: ~20B clicks (Bing + IE log)
Vocabulary: 30K **Parameter:** ~10M

KNET



$$\max \Pr(w_j|w_i) = \frac{\exp\left(c_1(w_i) v_{w_j}'^{T} v_{w_i} + c_2(w_i) v_{w_j}'^{T} v_{R_i}\right)}{\sum_{w \in V} \exp\left(c_1(w_i) v_w'^{T} v_{w_i} + c_2(w_i) v_w'^{T} v_{R_i}\right)}$$

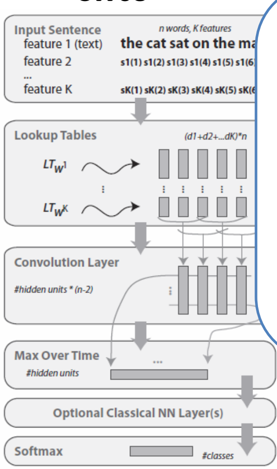
Huang, Po-Sen, et al. "Learning deep structured semantic models for web search using clickthrough data." in CIKM. ACM, 2013.

Count + Predict

Predict

Implicit Knowledge Representation: Embedding

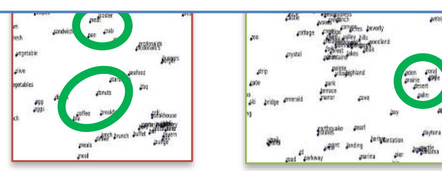
CW08



Input units: word
Vocabulary: 130k

Collobert, Ronan, et al. "Natural language processing (almost) from scratch." *The Journal of Machine Learning Research* 12 (2011): 2493-2537.

GloVe



Input units: word
Training size: > 42B tokens
Vocabulary: > 400K

J Pennington, R Socher, CD Manning "Glove: Global Vectors for Word Representation." EMNLP 2014.

Count + Predict

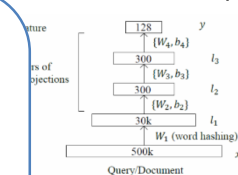
word2vec

Input units: word
Training size: > 100B sequence (Freebase)
Vocabulary: > 2M

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.

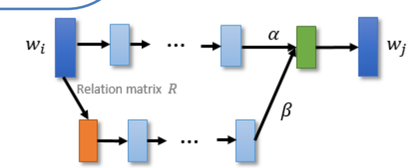
Predict

Deep Structured Semantic Model (DSSM)



: Tri-letter
e: ~20B clicks (Bing + IE log)
: 30K **Parameter:** ~10M

KNET



$$\max \Pr(w_j|w_i) = \frac{\exp\left(c_1(w_i){v'_{w_j}}^T v_{w_i} + c_2(w_i){v'_{w_j}}^T v_{R_i}\right)}{\sum_{w \in V} \exp\left(c_1(w_i){v'_{w}}^T v_{w_i} + c_2(w_i){v'_{w}}^T v_{R_i}\right)}$$

Huang, Po-Sen, et al. "Learning deep structured semantic models for web search using clickthrough data." in CIKM. ACM, 2013.

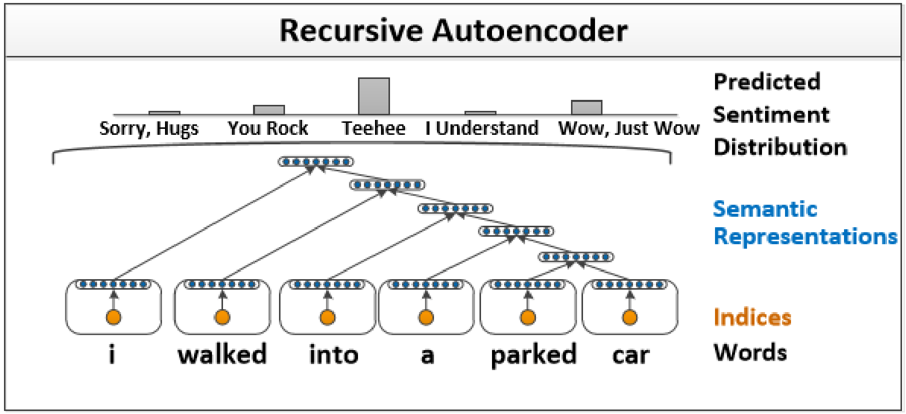
Pros:

- Dense semantic encoding
- A good representation framework
- Facilitates computation (similarity measure)

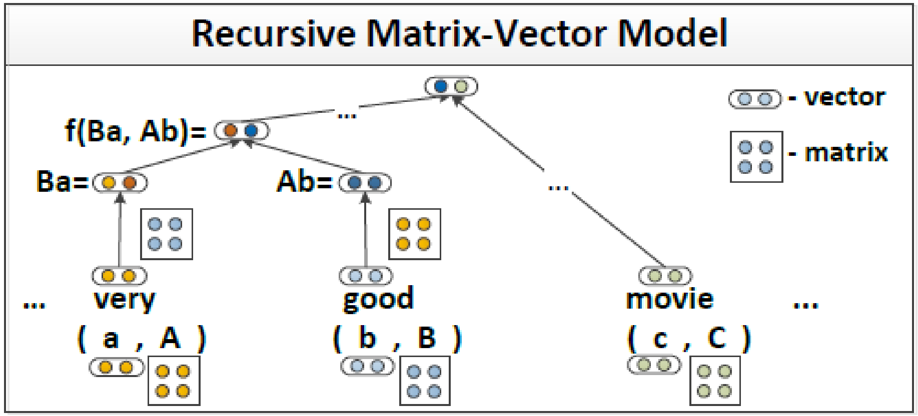
Cons:

- Perform poorly for rare words and new words
- Missing relations (e.g, isA, isPropertyOf)
- Hard to tune since it's not nature for human beings

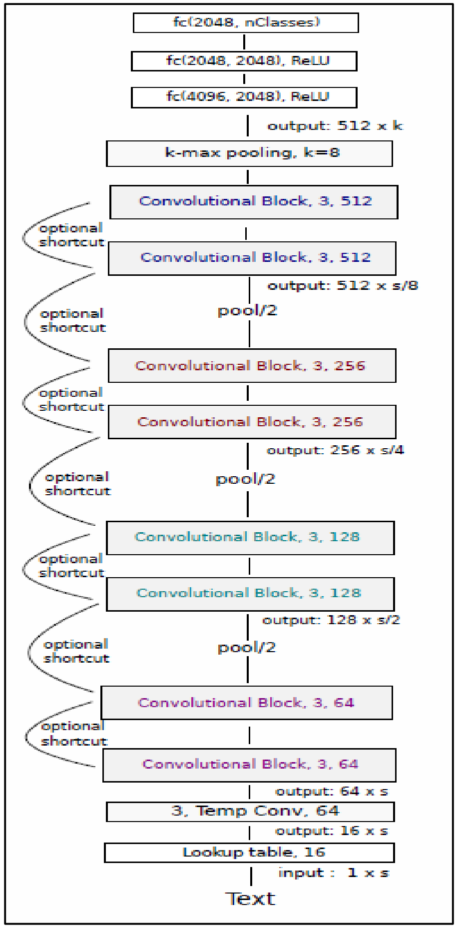
Implicit Knowledge Representation: DNN



Stanford Deep Autoencoder for Paraphrase Detection [Soucher et al. 2011]



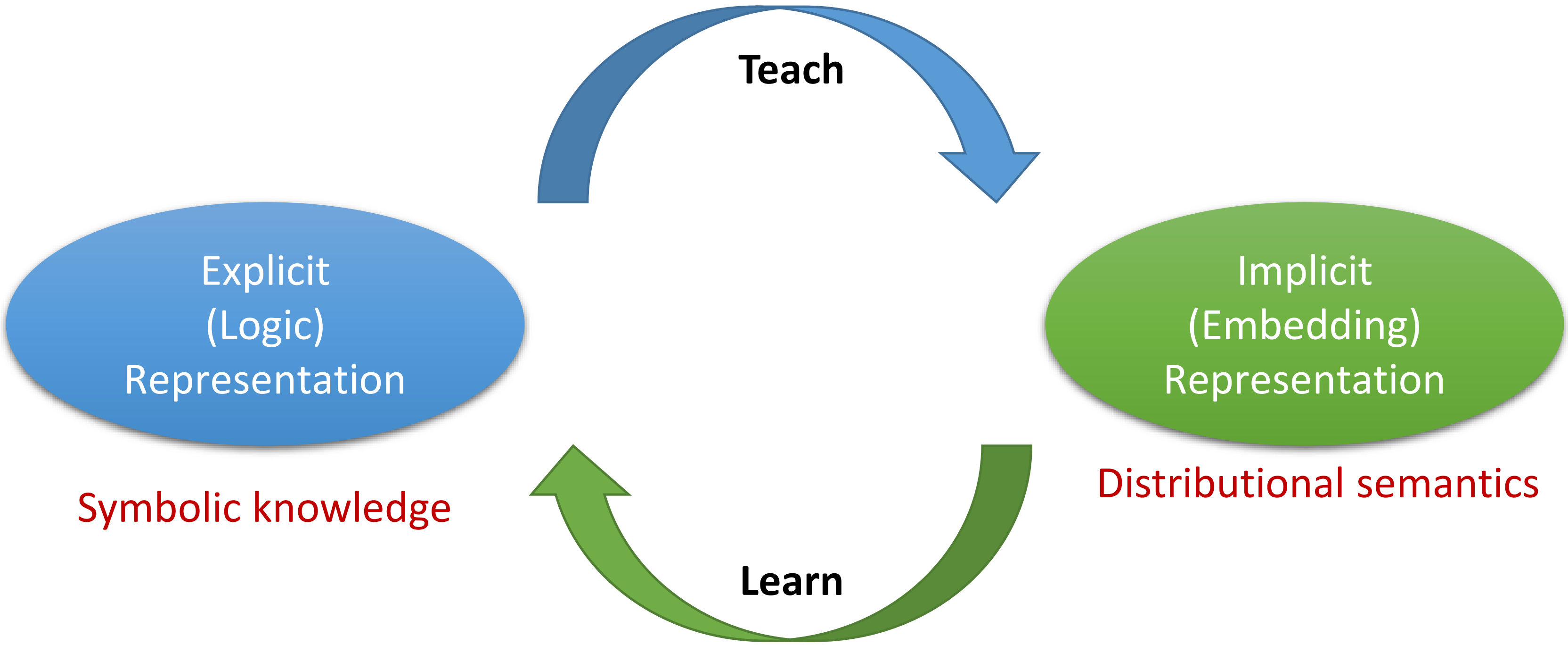
Stanford MV-RNN for Sentiment Analysis [Soucher et al. 2012]



Facebook DeepText classifier [Zhang et al. 2015]

New Trend: Fusion of Explicit and Implicit knowledge

- Relationship
- Rules of inference



Symbolic knowledge

Distributional semantics

- Learn more similar rules, enrich logic representation

Part II: Explicit Representation for Short Text Understanding

Zhongyuan Wang (Microsoft Research)

Haixun Wang (Facebook Inc.)

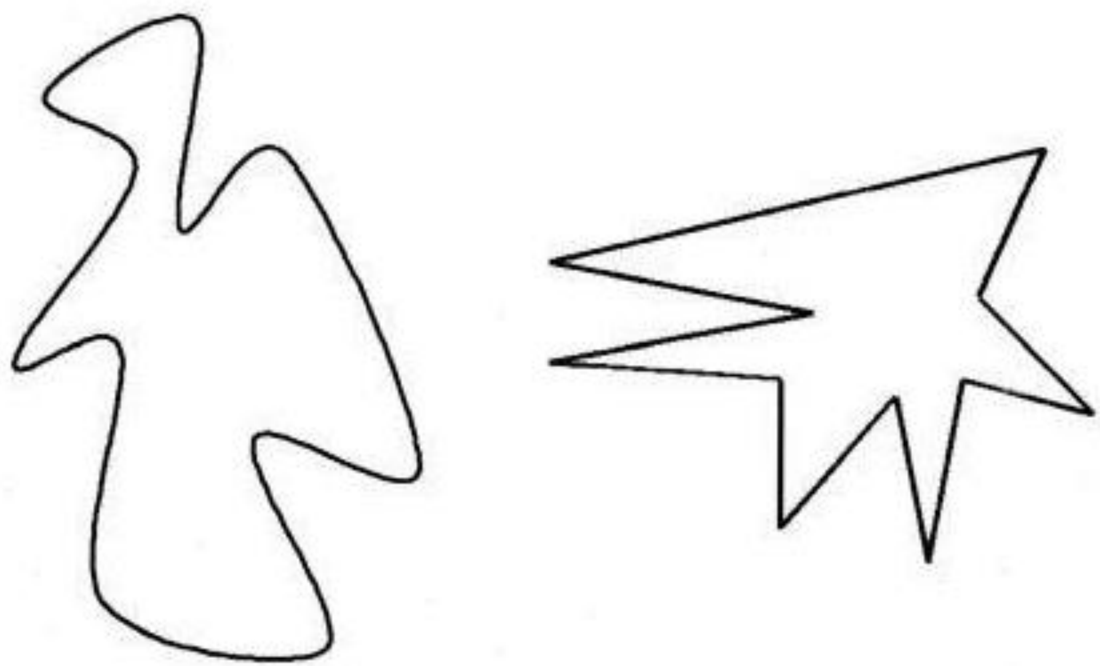
Tutorial Website:
http://www.wangzhongyuan.com/tutorial/ACL2016/Understanding-Short-Texts/

What is explicit understanding?

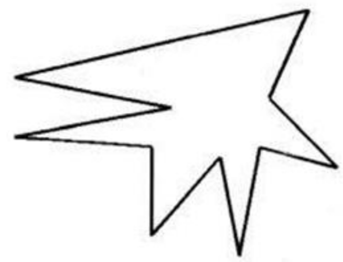
Add Common Sense to Computing

Pablo Picasso	**25 Oct 1881**	**Spanish**

Which is “kiki” and which is “bouba”?



$\backslash'\boldsymbol{k\bar{e}k\bar{e}}$



sound **shape**

zigzaggedness

China Brazil India

emerging market

country

The engineer is eating an apple

IT company

fruit

body smell taste

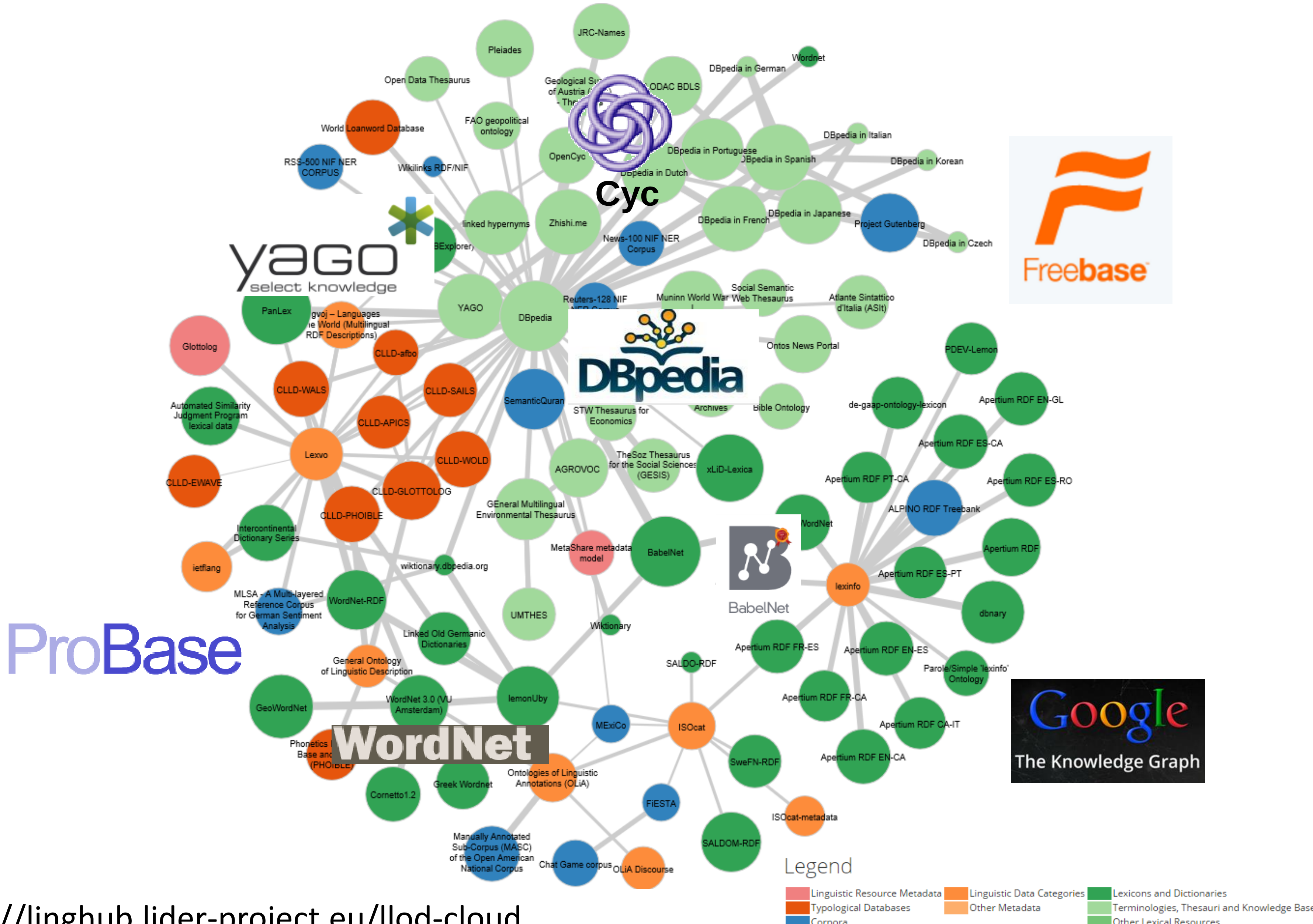
$\underbrace{\hspace{6em}}_{\textit{wine}}$

wine

Outline

- Knowledge Bases
- Explicit Representation Models
- Applications

Linguistic Linked Open Data Cloud



http://linghub.lider-project.eu/llod-cloud

1. *"Python Tutorial"*
2. *"Who was the U.S. President when the Angels won the World Series?"*

≠

linguistic, common sense knowledge knowledge

knowledge **Encyclopedia knowledge**

short text → understanding (internal representation) → answer

Common Sense Knowledge *vs.* Encyclopedia Knowledge

Common Sense Knowledge Base	Encyclopedia Knowledge Base
Common sense/linguistic knowledge among terms	Entities Facts
isA isPropertyOf co-occurrence …	DayOfBirth LocatedIn SpouseOf …
Typicality, basic level of categorization	Black or White Precision
WordNet*, KnowItAll, NELL, Probase, …	Freebase, Yago, DBPedia, Google knowledge graph, …

* Special cases

WordNet
A lexical database for English

WordNet [Stark et al. 1998]

Brief Introduction:

- WordNet® is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.

Statistics:

POS	*Unique Strings*	*Synsets*	*Total Word-Sense Pairs*
Noun	117798	82115	146312
Verb	11529	13767	25047
Adjective	21479	18156	30002
Adverb	4481	3621	5580
Totals	155287	117659	206941

Sample:

- S: (n) China, People's Republic of China, mainland China, Communist China, Red China, PRC, Cathay (a communist nation that covers a vast territory in eastern Asia; the most populous country in the world)

Authors:

- The project began in the Princeton University Department of Psychology, and is currently housed in the Department of Computer Science.

URLs:

- Homepage: http://wordnet.princeton.edu/wordnet/about-wordnet/
- Download: http://wordnet.princeton.edu/wordnet/download/

KnowItAll: Extract high-quality knowledge from the Web [Banko et al. 2007, Etzioni et al. 2011]

Brief Introduction:

- OpenIE distills semantic relations from Web-scale natural language texts.
- TextRunner -> ReVerb -> Open IE, part of KnowItAll

Statistics:

- Yielding over **5 billion extraction** from over a billion web pages.

Sample:

- **From** “U.S. president Barack Obama gave his inaugural address on January 20, 2013.”
  To: (Barack Obama; is president of; U.S.)
  (Barack Obama; gave; [his inaugural address, on January 20, 2013])

News:

- OpenIE v4.1.3 has been released.

Authors:

- Turing Center at the University of Washington

URLs:

- http://openie.allenai.org/
- http://reverb.cs.washington.edu/

NELL: Never-Ending Language Learning [Carlson et al. 2010]

Carnegie Mellon University

Brief Introduction:

- NELL is a research project that attempts to create a computer system that learns over time to read the web. Since January 2010.

Statistics:

- Over **50 million candidate beliefs** by reading the web. They are considered at different levels of confidence.
- Out of 50 million, **high confidence** in **2,817,156 beliefs**.

Sample:

Recently-Learned Facts twitter   Refresh

instance	iteration	date learned	confidence
melville_castle_hotel is a park	1003	06-jul-2016	100.0
bright_river is a river	1003	06-jul-2016	99.9
komo is a radio station	1003	06-jul-2016	100.0
spicy_yogurt is a visualizable thing	1004	11-jul-2016	99.6
alfred_tate is a monarch	1007	24-jul-2016	91.6
states is a state or province located in the geopolitical location cameroon	1004	11-jul-2016	93.8
the companies digg and google compete with eachother	1003	06-jul-2016	100.0
bernard_l__madoff is the CEO of new_york	1007	24-jul-2016	100.0
major_sports is a sport taught in the country __america	1006	18-jul-2016	99.8
eggs is an agricultural product coming from leatherback_turtle	1004	11-jul-2016	96.9

NELL: Never-Ending Language Learning

Carnegie Mellon University

News:

- It is continually learning facts on web. Resources is publicly available.

Authors:

- NELL research team at CMU.

URLs:

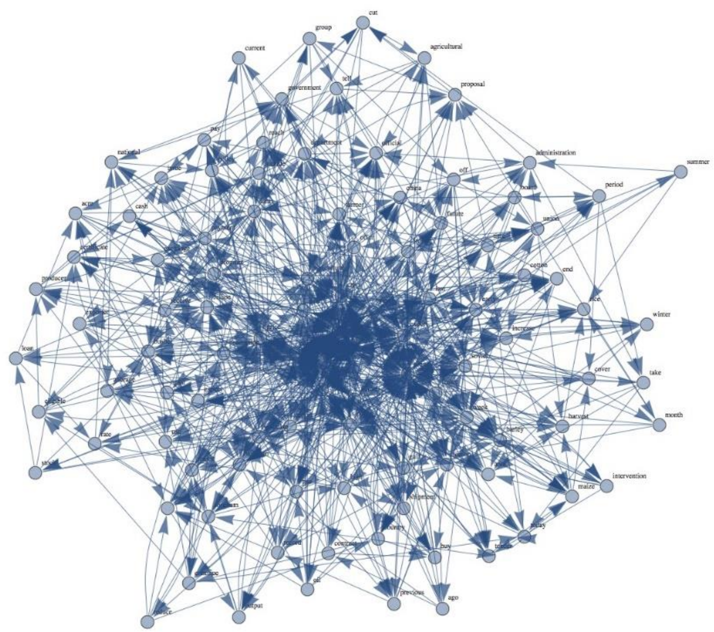
- Homepage: http://rtw.ml.cmu.edu/rtw/
- Download: http://rtw.ml.cmu.edu/rtw/resources

Probase [Wu et al. 2012]

Brief Introduction:

- Probase is a semantic network to make machines “aware” of the mental world of human beings, so that machines can better understand human communication.

Probase network:

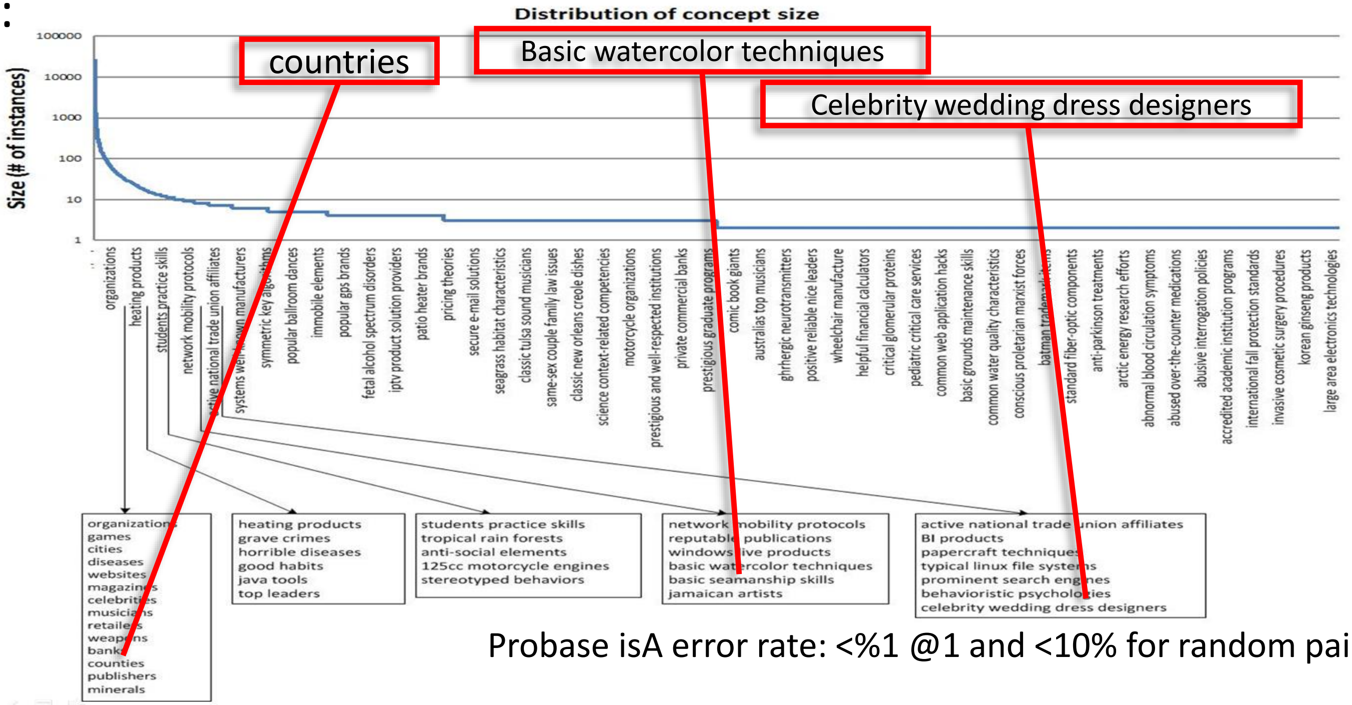


- **5,401,933** unique concepts
- **12,551,613** unique instances
- **87,603,947** IsA relations

Nodes:	Concepts (“Spanish Artists”)	Entities (“Pablo Picaso”)	Attributes (“Birthday”)	Verbs/Adjectives (“Eat”, “Sweet”)
Edges:	isA (concept, entities)	isPropertyOf (attributes)	Co-occurrence (isCEOof, LocatedIn, etc)	

Probase

Concepts:



Probase isA error rate: <%1 @1 and <10% for random pair

Authors:

- Microsoft Research

URLs:

- Public release: [**https://concept.research.microsoft.com/**](https://concept.research.microsoft.com/)
- Project homepage: http://research.microsoft.com/probase/

Freebase [Bollacker et al. 2008]

Freebase

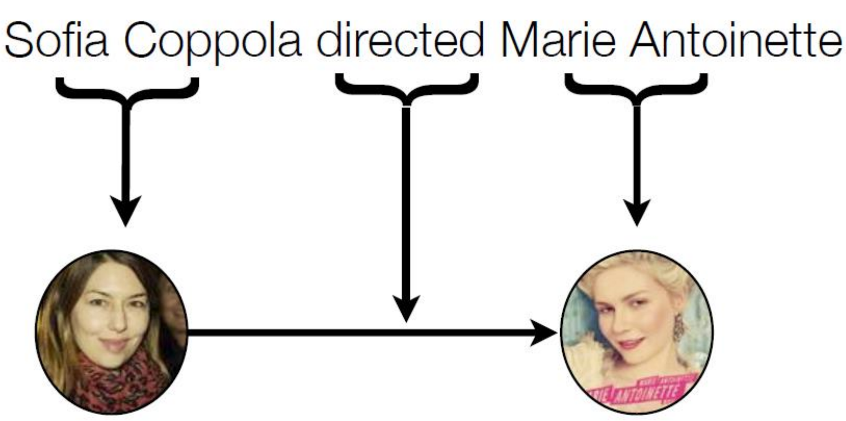
Brief Introduction:

- Freebase is a well-known collaborative knowledge base consisting of data composed mainly by its community

Statistics:

- Freebase contains more than **23 million entities**.
- Freebase contains **1.9 billion triples.**
- Each triple is organized as form of

  `<subject>    <predicate>    <object>`



- Freebase is a collection of facts.
- Freebase only contains nodes and links.
- Freebase is a labeled graph.

Freebase -> Wiki Data

News:

- Freebase data was integrated into Wikidata
- The Freebase API will be completely shut-down on Aug 31 2016, replaced by Google Knowledge Graph API

Authors:

- Freebase Community

URLs:

- Homepage: http://wiki.freebase.com/wiki/Main_Page
- Download: https://developers.google.com/freebase/
- Wikidata: https://www.wikidata.org/

Google Knowledge Graph

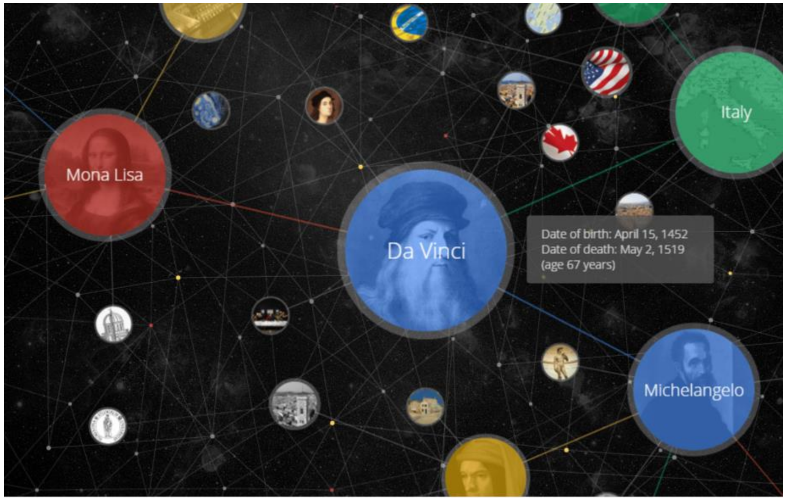
Brief Introduction:

- Knowledge Graph is a knowledge base used by Google to enhance its search engine's search results with semantic-search information gathered from a wide variety of sources.

Statistics:

- 570 million objects and more than 18 billion facts about relationships between different objects.

Sample:



Authors:

- Google Inc.

URLs:

- Homepage: https://www.google.com/intl/es419/insidesearch/features/search/knowledge.html

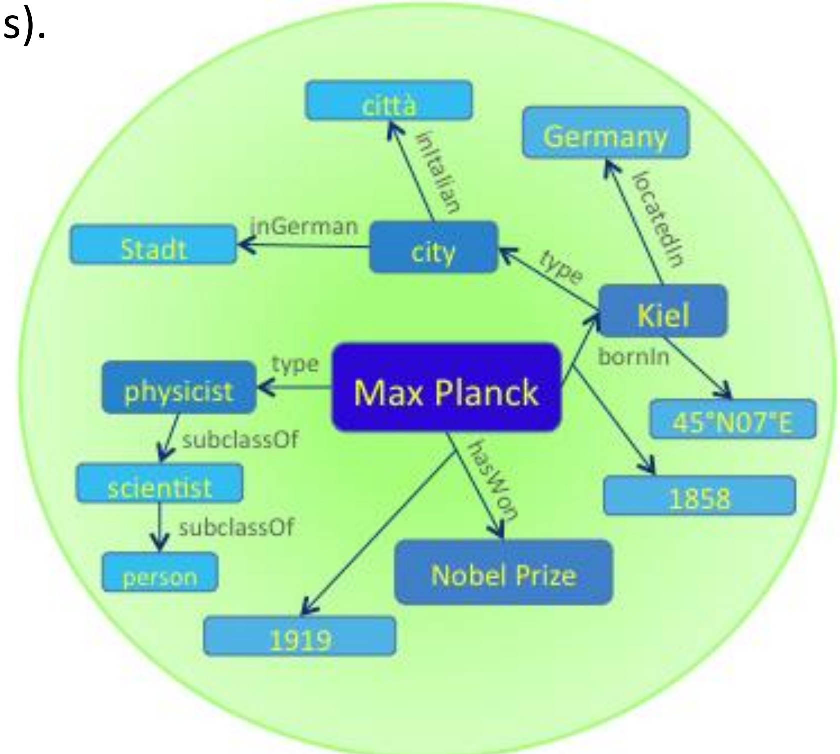
YAGO [Suchanek et al. 2007]

Brief Introduction:

- YAGO is a huge semantic knowledge base, derived from GeoNames WordNet and Wikipedia (10 Wikipedias in different languages).

Statistics:

- More than **10 million entities**.(persons, organizations, cities, etc)
- More than **120 million facts** about entities.
- More than **35,000 classes** assigned to entities.
- Many of its facts and entities are attached a **temporal** dimension and a **spatial** dimension.



Sample:

<Albert_Einstein> <isMarriedTo> <Elsa_Einstein>

YAGO

News:

- An evaluated version of YAGO3 (Combining information from Wikipedia from different languages) is released. [15 Sep. 2015]

Authors:

- Max Planck Institute for Informatics in Saarbrücken/Germany and DBWeb group at Télécom ParisTech University.

URLs:

- Homepage: http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago
- Download: http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads/

Outline

- Knowledge Bases
- Explicit Representation Models
- Applications

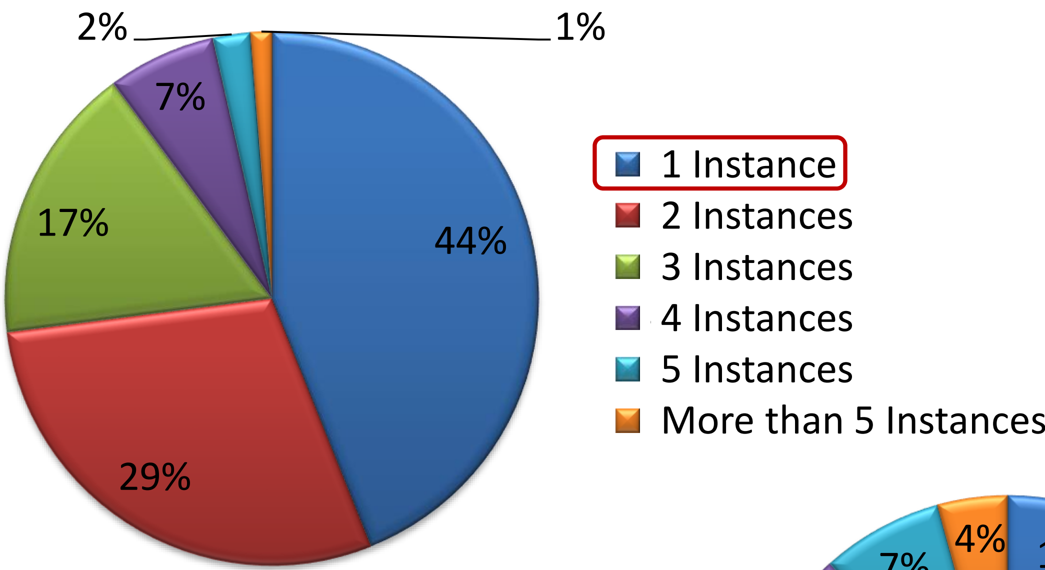
Statistics of Search Queries

Pokémon Go

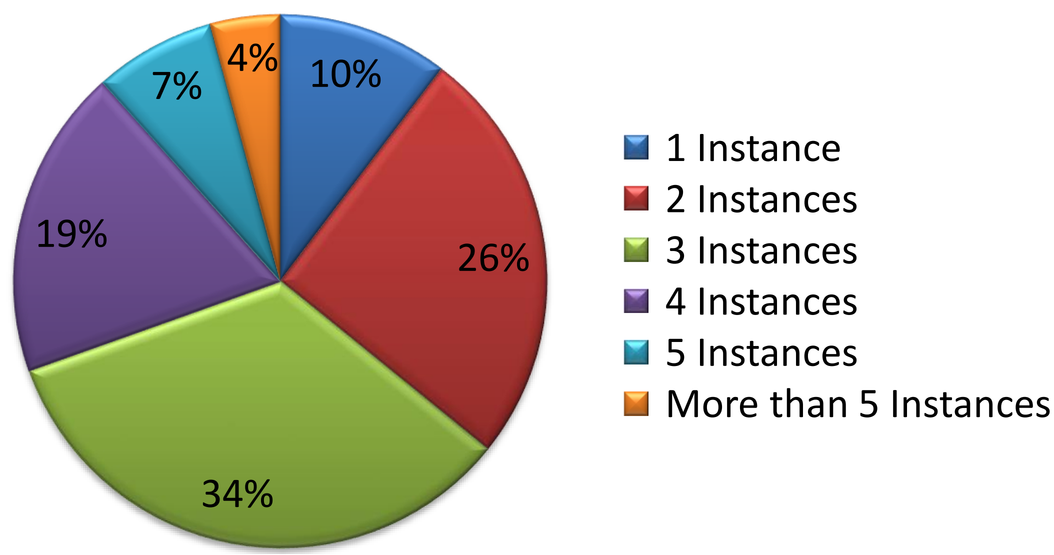
Instance 1

Microsoft HoloLens

Instance 2



(a) By traffic



(b) By # of distinct queries

If the short text is a single instance…

- Python
- Microsoft
- Apple
- …

Single Instance Understanding

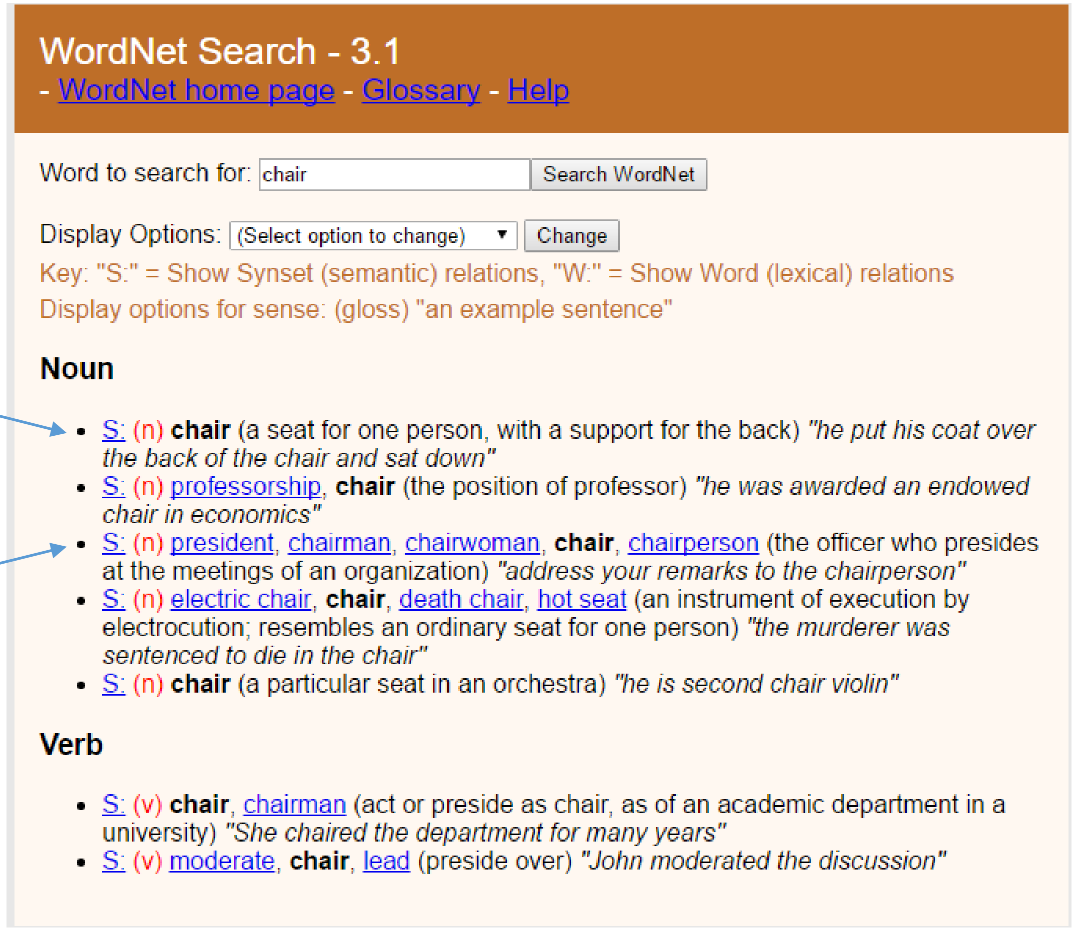
- Is this instance ambiguous?
- What are its basic-level concepts?
- What are its similar instances?

Word Ambiguity

- Word sense disambiguation: rely on dictionaries (WordNet)

Take a seat on this chair

The chair of the Math Department

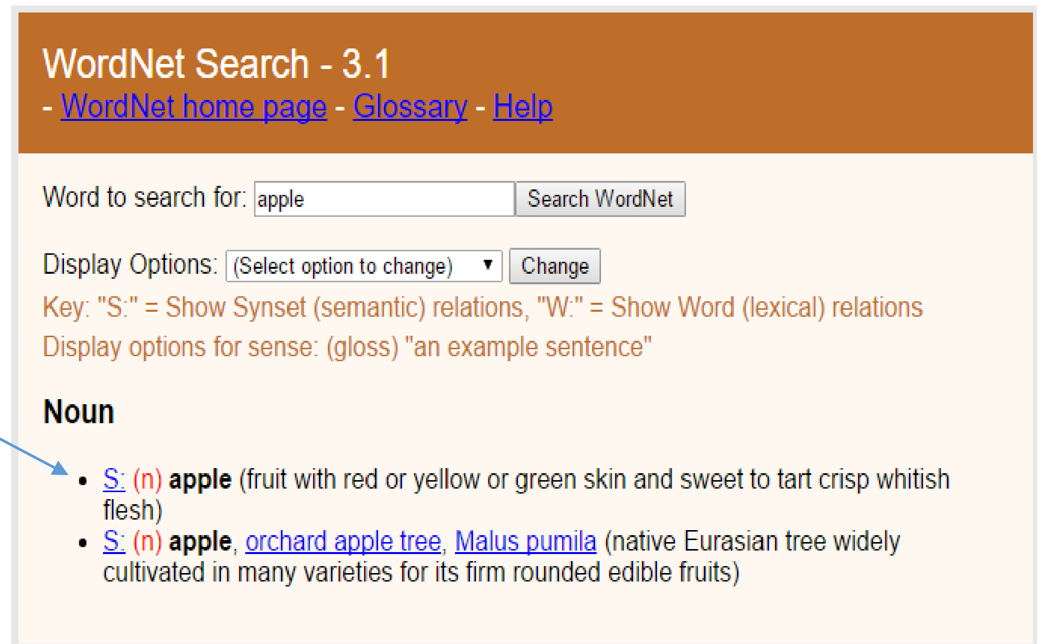


Instance Ambiguity

- Instance sense disambiguation: extra knowledge needed

I have an apple pie for lunch

He bought an apple ipad ?



Here "apple" is a proper noun!

Ambiguity [Hua et al. 2016]

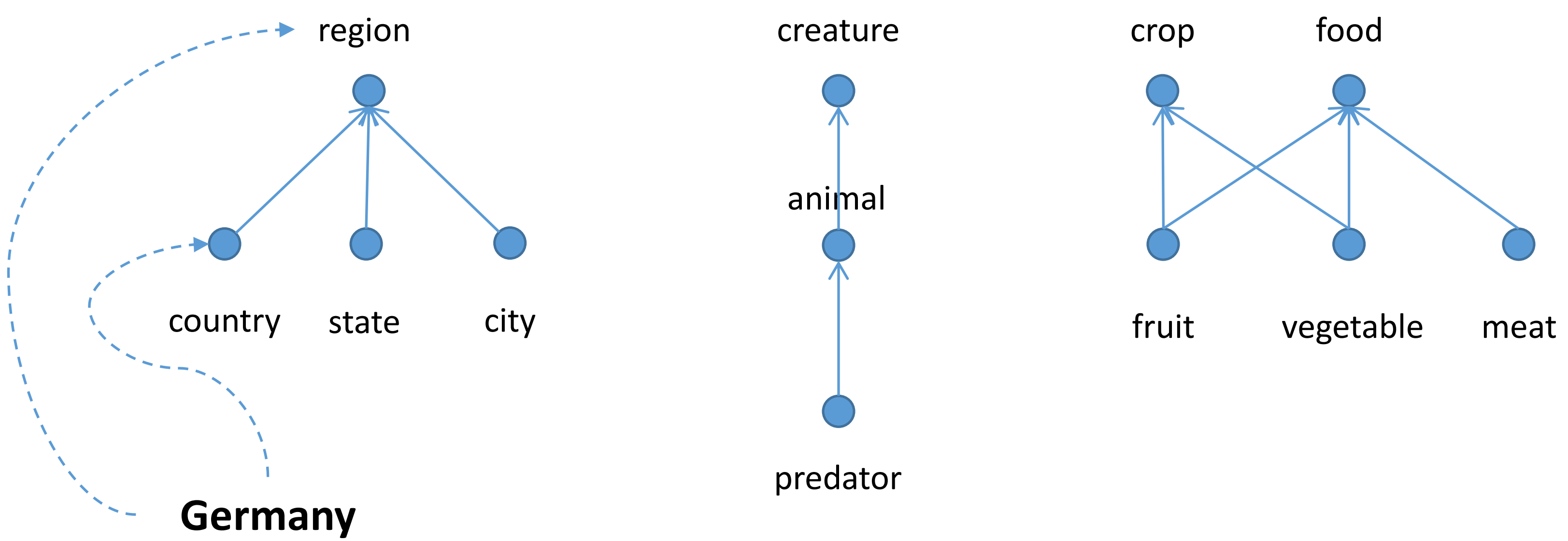
- Many instances are ambiguous

short text	instance	sense
population china	*china*	country
glass vs. china	*china*	fragile item
pear apple	*apple*	fruit
microsoft apple	*apple*	company
read harry potter	*harry potter*	book
watch harry potter	*harry potter*	movie
age of harry potter	*harry potter*	character

- Intuition: ambiguous instances have multiple senses

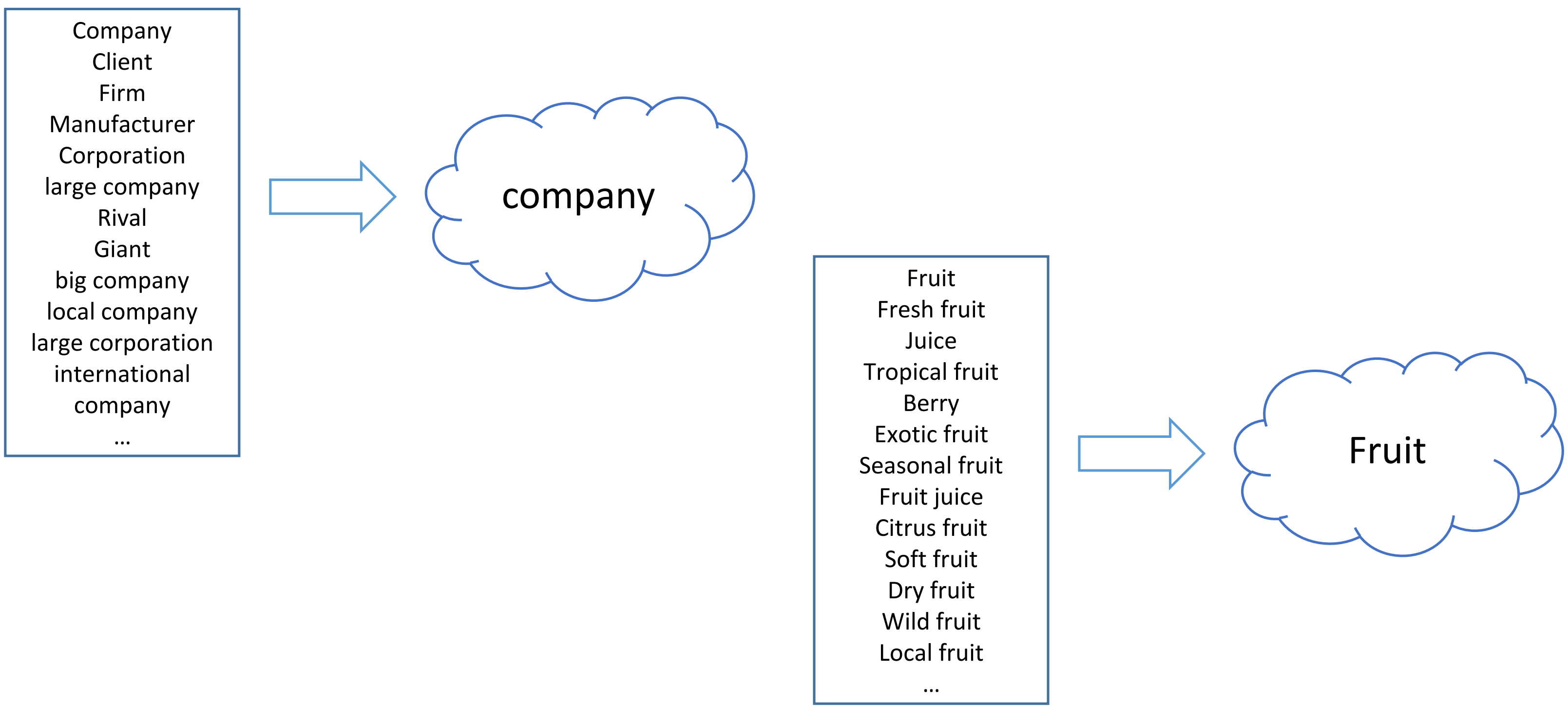
Pre-definition for Ambiguity (1): Sense [Hua et al. 2016]

- What is a Sense in semantic networks?
  - A sense as a hierarchy of *concept clusters*



Pre-definition for Ambiguity (2): Concept Cluster [Li et al. 2013, Li et al. 2015]

- What is a Concept Cluster (CL)?
  - Cluster similar concepts into a concept cluster using K-Means like approach (k-Medoids)



Definitions of Instance Ambiguity
[Hua et al. 2016]

- 3 levels of instance ambiguity
  - **Level 0: unambiguous**
    - Contains only 1 sense
    - E.g., *dog (animal), beijing (city), potato (vegetable)*
  - **Level 1: unambiguous and ambiguous both make sense**
    - Contains 2 or more senses, but these senses are related
    - E.g., *google (company & search engine), french (language & country), truck(vehicle & public transport service)*
  - **Level 2: ambiguous**
    - Contains 2 or more senses, and the senses are very different from each other
    - E.g., *apple (fruit & company), jaguar(animal & company), python (animal & language)*

Ambiguity Score

- Using top-2 senses to calculate the ambiguity score

$$score = \begin{cases} 0, & level = 0 \\ \dfrac{w(s_2|e)}{w(s_1|e)} * \big(1 - similarity(s_1, s_2)\big), & level = 1 \\ \text{score = } 1 + \dfrac{w(sc_2|e)}{w(sc_1|e)} * \big(1 - similarity(sc_1, sc_2)\big), & level = 2 \end{cases}$$

* Denote top-2 senses as s_1 and s_2, top-2 sense clusters as sc_1 and sc_2

* Denote similarity of two sense clusters as the maximum similarity of their senses:

$$similarity(sc_1, sc_2) = \boldsymbol{max}\{similarity(si \in sc_1, sj \in sc_2)\}$$

* For an entity e, denote the weight (popularity) of a sense s_i as the sum of weights of its concept clusters:

$$w(s_i|e) = w(H_i|e) = \sum\nolimits_{CL_j \in Hi} P(CLj|e)$$

* For an entity e, denote the weight (popularity) of a sense cluster sc_i as the sum of weights of its senses:

$$w(sc_i|e) = \sum\nolimits_{s_j \in sci} w(sj|e)$$

Examples

- Level 0:
  - california
    - country; state; city; region; institution: 0.943
  - fruit
    - food; product; snack; carbs; crop: 0.827
  - alcohol
    - substance; drug; solvent; food; addiction: 0.523
  - computer
    - device; product; electronics; technology; appliance: 0.537
  - coffee
    - beverage; product; food; crop; stimulant: 0.73
  - potato
    - vegetable; food; crop; carbs; product: 0.896
  - bean
    - food; vegetable; crop; legume; carbs: 0.801

Examples (cont.)

- Level 1:
  - nike, score = 0.034
    - company; store: 0.861
    - brand: 0.035
    - shoe; product: 0.033
  - twitter, score = 0.035
    - website; tool: 0.612
    - network: 0.165
    - application: 0.033
    - company: 0.031
  - facebook, score = 0.037
    - website; tool: 0.595
    - network: 0.17
    - company: 0.053
    - application: 0.029
  - yahoo, score = 0.38
    - search engine: 0.457
    - company; provider; account: 0.281
    - website: 0.0656
  - google, score = 0.507
    - search engine: 0.46
    - company; provider; organization: 0.377
    - website: 0.0449

Examples (cont.)

- Level 2:
    - jordan, score = 1.02
        - country; state; company; regime: 0.92
        - shoe: 0.02
    - fox, score = 1.09
        - animal; predator; species: 0.74
        - network: 0.064
        - company: 0.035
    - puma, score = 1.15
        - brand; company; shoe: 0.655
        - species; cat: 0.116
    - gold, score = 1.21
        - metal; material; mineral resource; mineral:0.62
        - color: 0.128

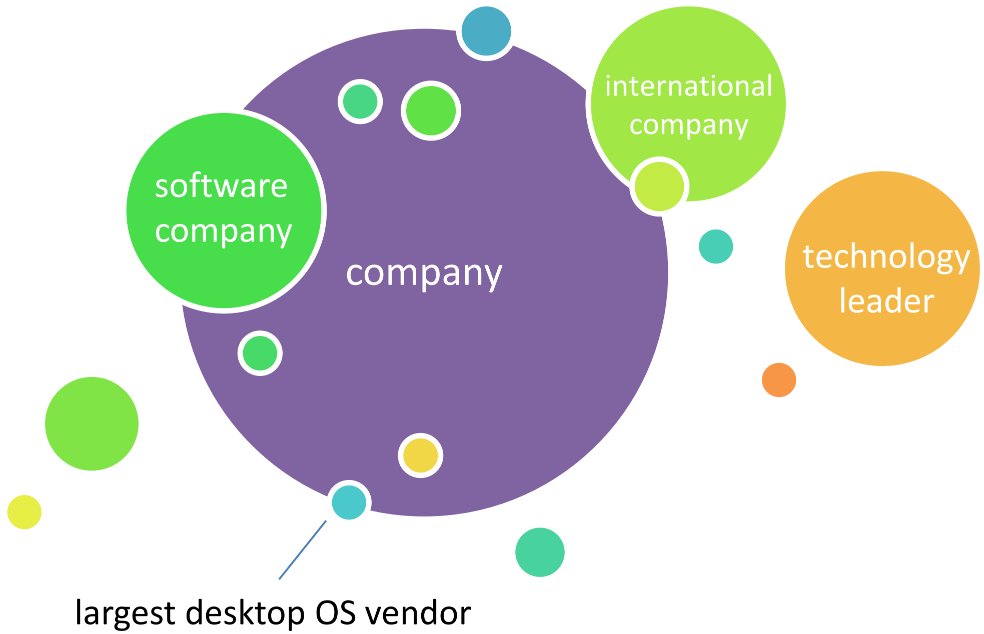
Examples (cont.)

- Level 2:
  - soap, score = 1.22
    - product; toiletry; substance: 0.49
    - technology; industry standard: 0.11
  - silver, score = 1.24
    - metal; material; mineral resource; mineral: 0.638
    - color: 0.156
  - python, score = 1.29
    - language: 0.667
    - snake; animal; reptile; skin: 0.193
  - apple, score = 1.41
    - fruit; food; tree: 0.537
    - company; brand: 0.271

Single Instance

- Is this instance ambiguous?
- What are its basic-level concepts?
- What are its similar instances?

A Concept View of “Microsoft”



Basic-level Conceptualization (BLC)

[Rosch et al. 1976]

Category Level	Informative?	Distinctive?
Superordinate	No	Yes
Basic-level	Yes	Yes
Subordinate	Yes	No

Basic-level conceptualization

software company

company ... ... **largest desktop OS vendor**

KFC

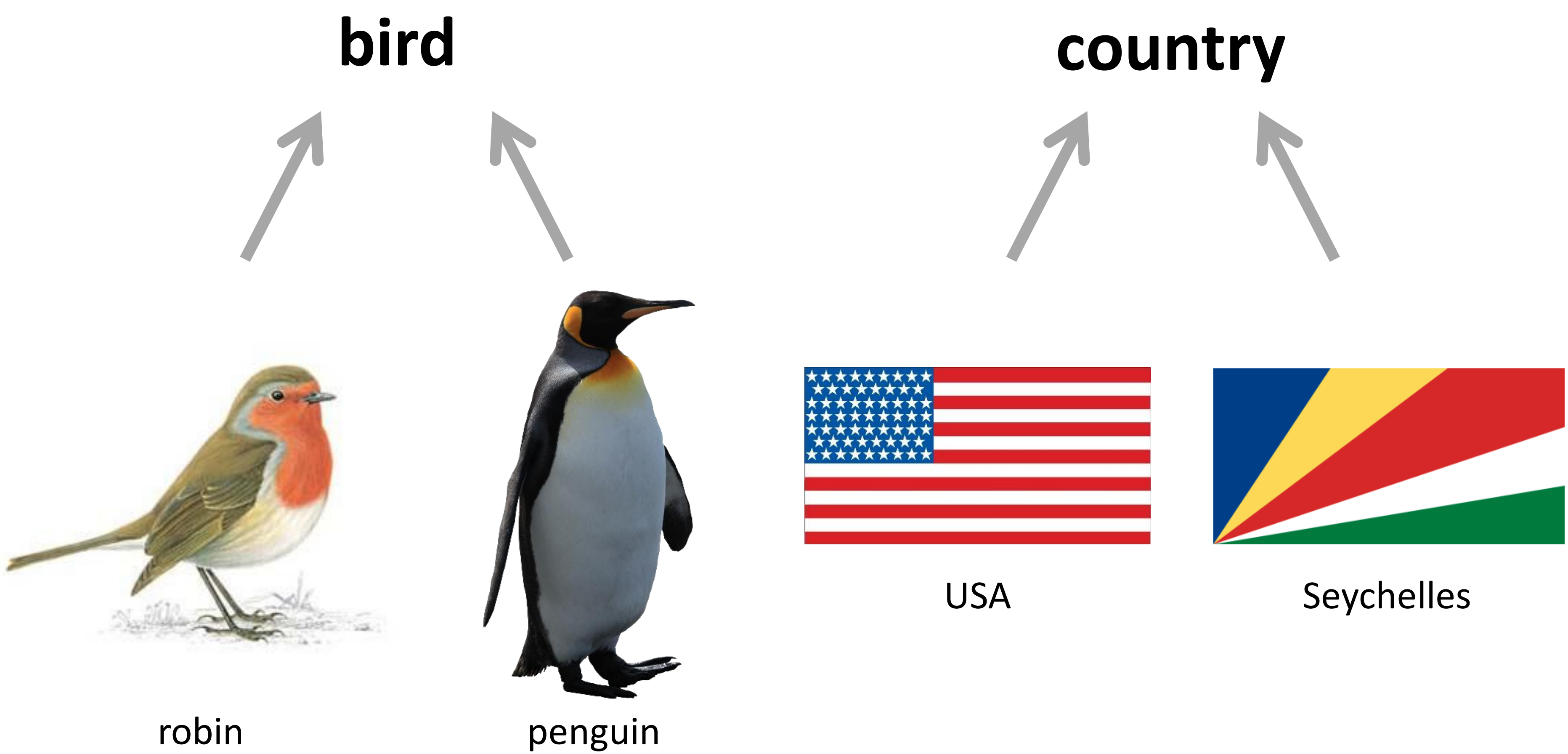
BMW

Microsoft

How to Make BLC?

- Naive approaches
  - **Typicality**: an important measure for understanding the relationship between an object and its concept
  - **Pointwise Mutual Information (PMI)**: a common measure of the strength of association between two terms

Naive Approach 1: Typicality



"robin" is a more *typical* bird than a "penguin" ⇨ *P(robin|bird) > P(penguin|bird)*

"USA" is a more *typical* country than "Seychelles" ⇨ *P(USA|country) > P(*Seychelles|country*)*

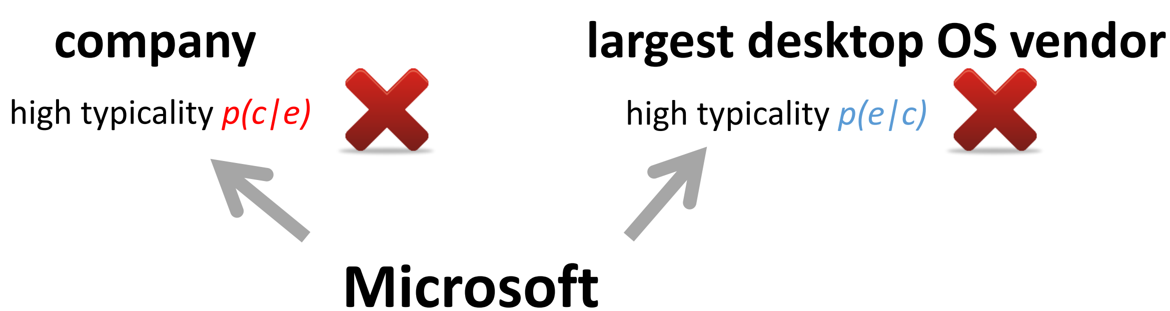
Using Typicality for BLC

- Associate each isA relationship (e is c) with typicality scores $P(e|c)$ and $P(c|e)$

$$P(e|c) = \frac{n(c,e)}{n(c)} \qquad P(c|e) = \frac{n(c,e)}{n(e)}$$

  - *P(e|c)* indicates how *typical (or popular) e* is in the given concept *c*
  - *P(c|e)* indicates how *typical (or popular)* the concept *c* is given *e*

- However,



Naive Approach 2: PMI

[Manning and Schutze 1999]

- **Pointwise mutual information** (**PMI**), is a measure of association used in information theory and statistics
- Consider using the *PMI* between concept *c* and instance *e* to find the basic-level concepts as follows:

$$PMI(e,c) = \log\frac{P(e,c)}{P(e)P(c)} = \log P(e|c) - \log P(e)$$

- However,
  - In basic level of categorization, we are interested in finding a concept for a given e, which means P(e) is a constant
  - Thus, ranking by PMI(e, c) is the same as ranking by P(e|c)

Using *Rep*(*e, c*) for BLC [Wang et al. 2015b]

- **The measure** $Rep(e,c) = P(c|e) * P(e|c)$ means:

Given *e*, the *c* should be its typical concept (shortest distance)

Given *c*, the *e* should be its typical instance (shortest distance)

A process of finding *concept nodes* having shortest expected distance with *e*

- **(With PMI)** If we take the logarithm of our scoring function, we get:

$$\log Rep(e,c) = \log P(c|e) * P(e|c) = \log \frac{P(e,c)}{P(e)} * \frac{P(e,c)}{P(c)} = \log \frac{P(e,c)^2}{P(e)P(c)} = PMI(e,c) + \log P(e,c)$$
$$= PMI^2$$

- **(With Commute Time)** The commute time between an instance *e* and a concept *c* is:

$$Time(e,c) = \sum_{k=1}^{\infty} (2k) * P_k(e,c) = \sum_{k=1}^{T} (2k) * P_k(e,c) + \sum_{k=T+1}^{\infty} (2k) * P_k(e,c)$$
$$\geq \sum_{k=1}^{T} (2k) * P_k(e,c) + 2(T+1) * (1 - \sum_{k=1}^{T} P_k(e,c)) = 4 - 2 * Rep(e,c)$$

Evaluations on Different Measures for BLC

NDCG

No Smoothing	1	2	3	5	10	15	20
MI(e)	0.516	0.531	0.519	0.531	0.562	0.574	0.594
PMI3(e)	0.725	0.664	0.652	0.660	0.628	0.631	0.646
NPMI(e)	0.599	0.597	0.579	0.554	0.540	0.539	0.549
Typicality P(c|e)	0.297	0.380	0.409	0.422	0.438	0.446	0.461
Typicality P(e|c)	0.401	0.386	0.396	0.398	0.401	0.410	0.428
Rep(e)	**0.758**	**0.771**	**0.745**	**0.723**	**0.656**	**0.647**	**0.661**
Smoothing=1e-3							
MI(e)	0.374	0.414	0.441	0.448	0.473	0.481	0.495
PMI3(e)	0.484	0.511	0.509	0.502	0.519	0.525	0.533
NPMI(e)	0.692	0.652	0.607	0.603	0.585	0.585	0.592
Typicality P(c|e)	0.297	0.380	0.409	0.422	0.438	0.446	0.460
Typicality P(e|c)	**0.703**	**0.697**	**0.704**	**0.681**	**0.637**	**0.628**	**0.626**
Rep(e)	0.621	0.580	0.554	0.561	0.554	0.555	0.559
Smoothing=1e-4							
MI(e)	0.407	0.430	0.458	0.462	0.492	0.503	0.512
PMI3(e)	0.648	0.604	0.579	0.575	0.578	0.576	0.590
NPMI(e)	0.747	0.777	0.761	0.737	0.700	0.685	0.688
Typicality P(c|e)	0.297	0.380	0.409	0.422	0.438	0.446	0.461
Typicality P(e|c)	**0.791**	**0.795**	**0.802**	**0.767**	**0.738**	**0.729**	**0.724**
Rep(e)	0.758	0.714	0.711	0.689	0.653	0.636	0.653
Smoothing=1e-5							
MI(e)	0.429	0.465	0.478	0.501	0.517	0.528	0.545
PMI3(e)	0.725	0.647	0.642	0.642	0.627	0.624	0.638
NPMI(e)	**0.813**	0.779	**0.778**	**0.765**	**0.730**	**0.723**	**0.729**
Typicality P(c|e)	0.297	0.380	0.409	0.422	0.438	0.446	0.461
Typicality P(e|c)	0.709	0.728	0.735	0.722	0.702	0.696	0.703
Rep(e)	0.791	**0.787**	0.762	0.739	0.707	0.703	0.706
Smoothing=1e-6							
MI(e)	0.516	0.510	0.515	0.526	0.546	0.563	0.579
PMI3(e)	0.725	0.655	0.651	0.654	0.641	0.631	0.649
NPMI(e)	**0.791**	0.766	0.732	0.728	0.673	0.659	0.668
Typicality P(c|e)	0.297	0.380	0.409	0.422	0.438	0.446	0.461
Typicality P(e|c)	0.495	0.516	0.520	0.508	0.512	0.521	0.540
Rep(e)	0.758	**0.784**	**0.767**	**0.755**	**0.691**	**0.686**	**0.694**
Smoothing=1e-7							
MI(e)	0.516	0.531	0.519	0.530	0.562	0.571	0.592
PMI3(e)	0.725	0.664	0.652	0.658	0.630	0.631	0.647
NPMI(e)	0.670	0.655	0.633	0.604	0.575	0.570	0.581
Typicality P(c|e)	0.297	0.380	0.409	0.422	0.438	0.446	0.461
Typicality P(e|c)	0.423	0.421	0.415	0.407	0.414	0.424	0.438
Rep(e)	**0.758**	**0.771**	**0.745**	**0.725**	**0.663**	**0.661**	**0.668**

Precision

No smoothing	1	2	3	5	10	15	20
MI(e)	0.769	0.692	0.705	0.685	0.719	0.705	0.690
PMI3(e)	**0.885**	0.769	0.756	0.800	0.754	**0.733**	**0.721**
NPMI(e)	0.692	0.692	0.667	0.638	0.627	0.610	0.610
Typicality P(c|e)	0.462	0.577	0.603	0.577	0.569	0.564	0.556
Typicality P(e|c)	0.500	0.462	0.526	0.523	0.523	0.510	0.521
Rep(e)	0.846	**0.865**	**0.872**	**0.862**	**0.758**	0.731	0.719
Smoothing=0.001							
MI(e)	0.577	0.615	0.628	0.600	0.612	0.605	0.592
PMI3(e)	0.731	0.673	0.692	0.654	0.669	0.644	0.623
NPMI(e)	**0.923**	0.827	0.769	0.746	0.731	0.695	0.671
Typicality P(c|e)	0.462	0.577	0.603	0.577	0.569	0.564	0.554
Typicality P(e|c)	0.885	**0.865**	**0.872**	**0.831**	**0.785**	**0.741**	**0.704**
Rep(e)	0.846	0.731	0.718	0.723	0.700	0.669	0.638
Smoothing=0.0001							
MI(e)	0.615	0.615	0.654	0.608	0.635	0.628	0.612
PMI3(e)	0.846	0.731	0.731	0.715	0.723	0.685	0.677
NPMI(e)	0.885	**0.904**	0.885	0.869	0.823	0.777	0.752
Typicality P(c|e)	0.462	0.577	0.603	0.577	0.569	0.564	0.556
Typicality P(e|c)	0.885	**0.904**	**0.910**	**0.877**	**0.831**	**0.813**	**0.777**
Rep(e)	**0.923**	0.846	0.833	0.815	0.781	0.736	0.719
Smoothing=1e-5							
MI(e)	0.615	0.635	0.667	0.662	0.677	0.656	0.646
PMI3(e)	**0.885**	0.769	0.744	0.777	0.758	0.731	0.710
NPMI(e)	**0.885**	0.846	**0.872**	**0.869**	**0.831**	**0.810**	**0.787**
Typicality P(c|e)	0.462	0.577	0.603	0.577	0.569	0.564	0.556
Typicality P(e|c)	0.769	0.808	0.846	0.823	0.808	0.782	0.765
Rep(e)	**0.885**	**0.904**	**0.872**	0.862	0.812	0.800	0.767
Smoothing=1e-6							
MI(e)	0.769	0.673	0.705	0.677	0.700	0.692	0.679
PMI3(e)	**0.885**	0.769	0.756	0.785	0.773	0.726	0.723
NPMI(e)	**0.885**	0.846	0.821	0.815	0.750	0.726	0.719
Typicality P(c|e)	0.462	0.577	0.603	0.577	0.569	0.564	0.556
Typicality P(e|c)	0.538	0.615	0.615	0.615	0.608	0.613	0.615
Rep(e)	0.846	**0.885**	**0.897**	**0.877**	**0.788**	**0.777**	**0.765**
Smoothing=1e-7							
MI(e)	0.769	0.692	0.705	0.685	0.719	0.703	0.688
PMI3(e)	**0.885**	0.769	0.756	0.792	0.758	0.736	0.725
NPMI(e)	0.769	0.750	0.718	0.700	0.650	0.641	0.633
Typicality P(c|e)	0.462	0.577	0.603	0.577	0.569	0.564	0.556
Typicality P(e|c)	0.500	0.481	0.526	0.523	0.531	0.523	0.523
Rep(e)	0.846	**0.865**	**0.872**	**0.854**	**0.765**	**0.749**	**0.733**

Single Instance

- Is this instance ambiguous?
- What are its basic-level concepts?
- What are its similar instances?

What is the Semantic Similarity?

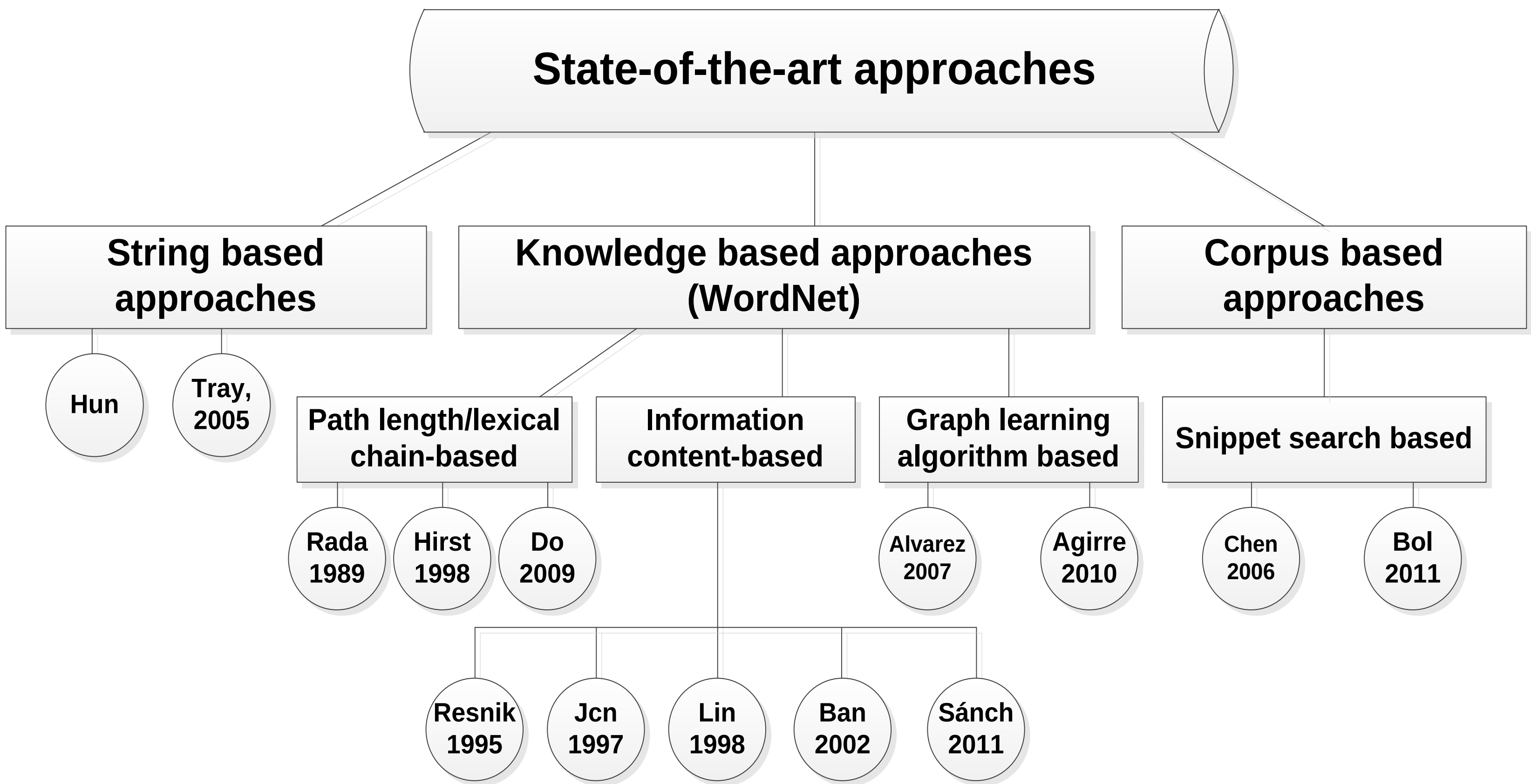
- Are the following instance pairs similar?
  - **<apple, microsoft>**
  - **<apple, pear>**
  - **<apple, fruit>**
  - **<apple, food>**
  - **<apple, ipad>**
  - **<car, journey>**

Approaches on Term Similarity

- Categories of approaches for semantic similarity
  - String based approach
  - Knowledge based approach
    - Use preexisting thesauri, taxonomy or encyclopedia, such as WordNet;
  - Corpus based approach
    - Use contexts of terms extracted from web pages, web search snippets or other text repositories;
  - Embedding based approach
    - *Will introduce in detail in "Part 3: Implicit Understanding"*

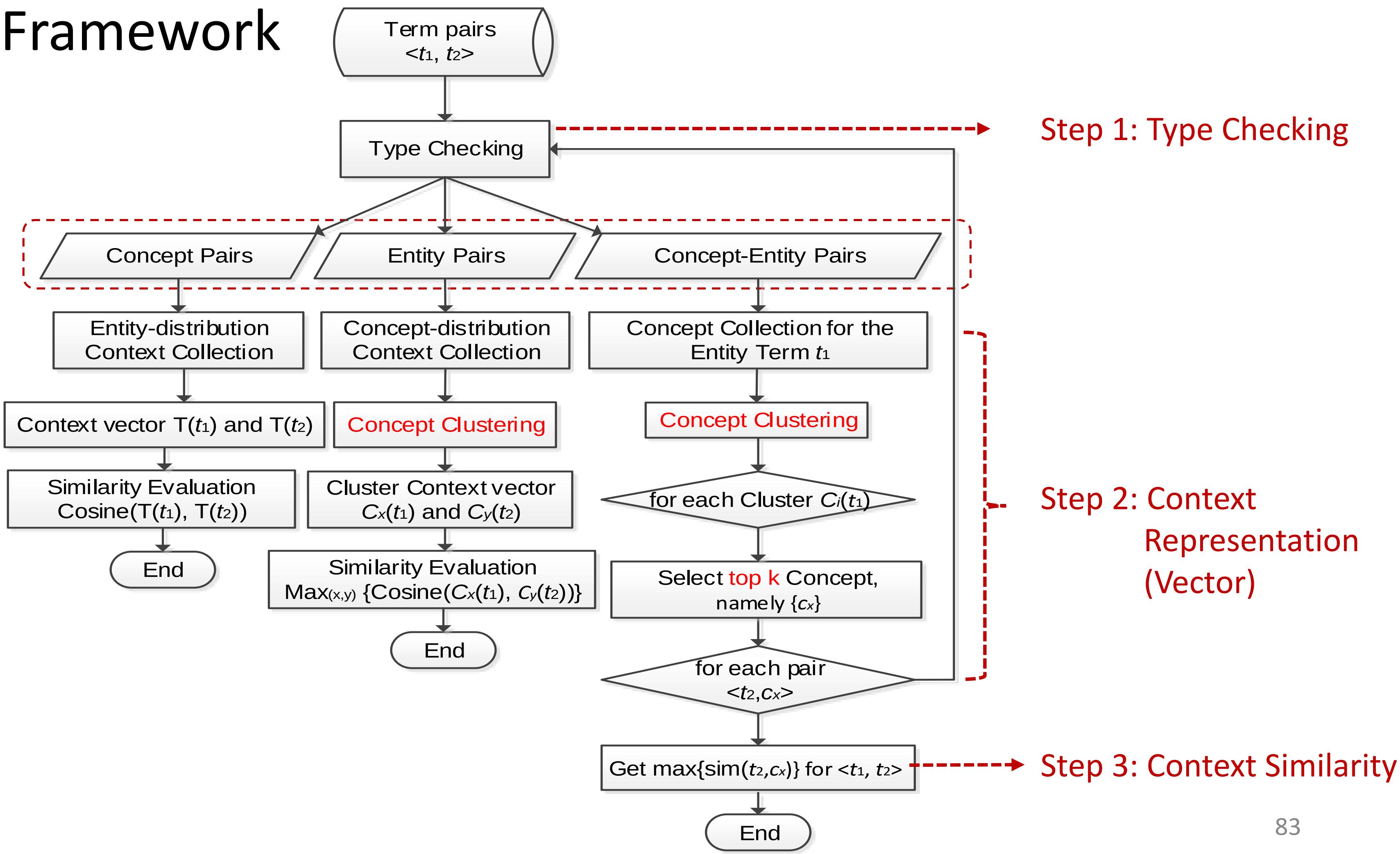
Approaches on Term Similarity (2)

- Categories



Term Similarity Using Semantic Networks [Li et al. 2013, Li et al. 2015]

- Framework



An example [Li et al. 2013, Li et al. 2015]

For example:

<banana, pear>

<banana, pear>

Type Checking

Entity Pairs

Step 1: Type Checking

Concept Context Collection

concept	entity	frequency
fruit	banana	1319
food	banana	637
crop	banana	238
fresh fruit	banana	144
tropical fruit	banana	134
potassium-rich food	banana	95
item	banana	83
product	banana	83
flavor	banana	82
plant	banana	68
bland food	banana	57
thing	banana	48
potassium rich food	banana	46
soft fruit	banana	43
fruit tree	banana	39
healthy snack	banana	39
soft food	banana	34
cash crop	banana	30
ingredient	banana	30
agricultural product	banana	29

concept	entity	frequency
fruit	pear	883
food	pear	126
fruit tree	pear	109
fresh fruit	pear	66
crop	pear	42
tree	pear	32
product	pear	19
tree fruit	pear	18
flavor	pear	18
item	pear	18
fruit crop	pear	17
others	pear	15
dried fruit	pear	15
juicy fruit	pear	15
climacteric fruit	pear	12
soft fruit	pear	12
pomes	pear	12
juice	pear	12
cut	pear	11
large fruit	pear	10

Step 2: Context Representation (Vector)

Similarity Evaluation, Cosine(T(t_1), T(t_2)): 0.916

Step 3: Context Similarity

Examples

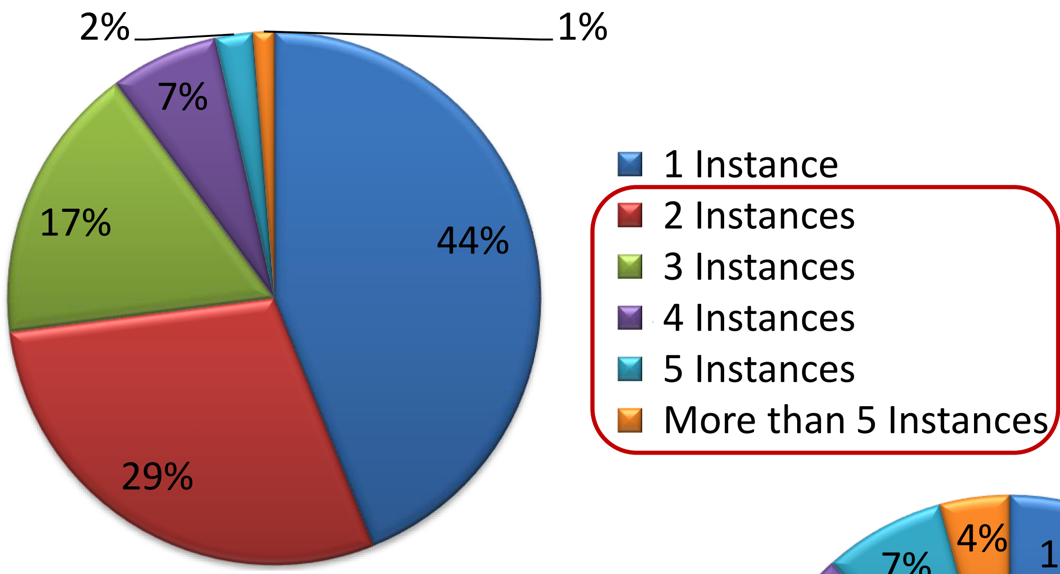
Term 1	Term 2	Similarity
lunch	dinner	0.9987
tiger	jaguar	0.9792
car	plane	0.9711
television	radio	0.9465
technology company	microsoft	0.8208
high impact sport	competitive sport	0.8155
employer	large corporation	0.5353
fruit	green pepper	0.2949
travel	meal	0.0426
music	lunch	0.0116
alcoholic beverage	sports equipment	0.0314
company	table tennis	0.0003

http://adapt.seiee.sjtu.edu.cn/similarity/SimCompleteResults.pdf

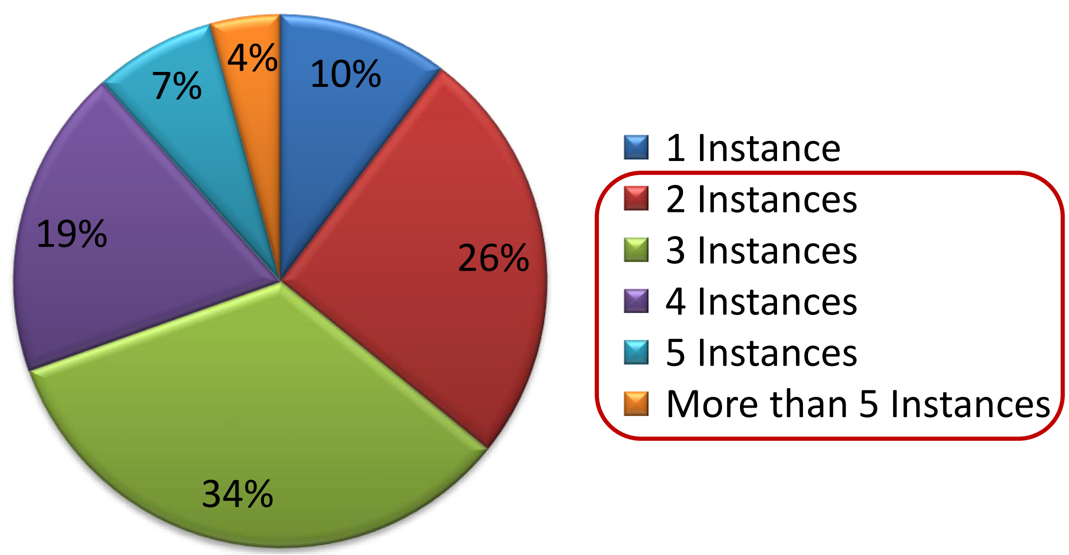
Statistics of Search Queries

Pokémon Go *Microsoft HoloLens*

Instance 1 *Instance 2*



(a) By traffic



(b) By # of distinct queries

If the short text has context for the instance…

- python tutorial
- dangerous python
- moon earth distance
- …

Short Text Understanding

- How to segment this short text?
- What does this short text mean (its intent, senses, or concepts)?
- What are the relations among terms in the short text?
- How to calculate the similarity between short texts?

Supervised Segmentation [Bergsma et al. 2007]

- Problem: divide query into semantic units

Example Query:

Two man power saw

?

[two man] [power saw]
[two] [man] [power saw]
[two] [man power] [saw]

- Approach: turn segmentation into position-based binary classification

Input: a query and its positions

Output: the decision for making segmentation at each position

Supervised Segmentation

- Features
  - **Decision boundary features**
  - **Statistical features**

e.g. Indicators: the
POS tags in query: is
Position features: forward/backward

$$\mathrm{MI}(x_{L0}, x_{R0}) = \log \frac{\Pr(x_{L0}x_{R0})}{\Pr(x_{L0})\Pr(x_{R0})}$$

Mutual information between left and right parts

  - **Context features**
  - **Dependency features**

Context information

Bank loan amortization schedule

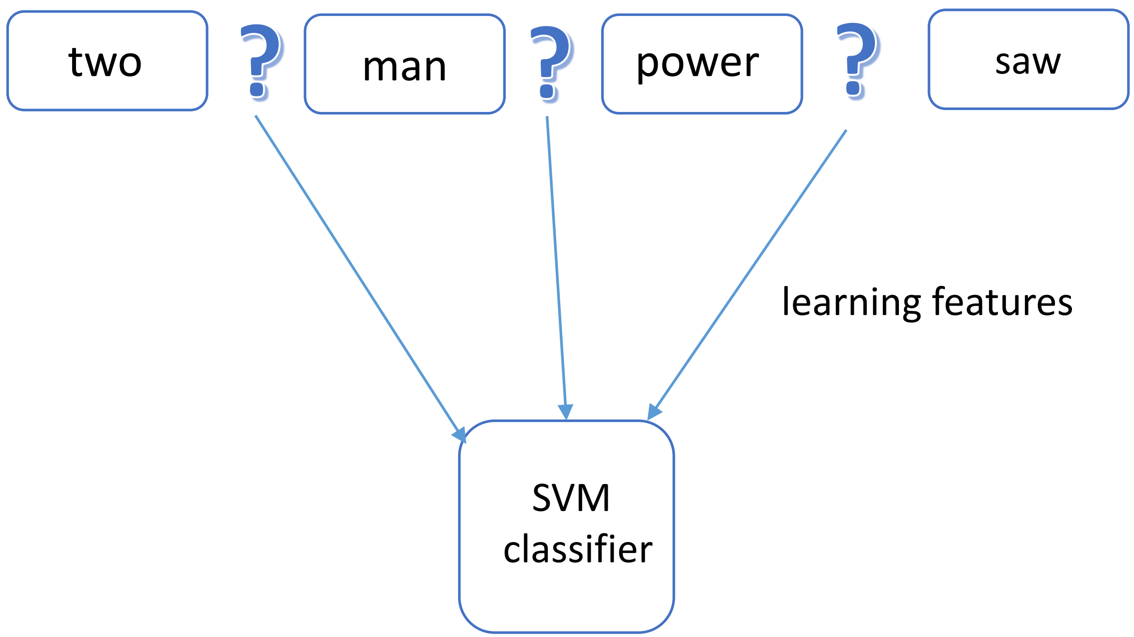
depend

female bus driver

Supervised Segmentation

- Segmentation Overview

Input query: two man power saw



Output: segmentation decision for each position (yes/no)

Unsupervised Segmentation [Tan et al. 2008]

- Unsupervised learning for query segmentation

Probability of generated segmentation S for query Q

$$P(S^Q) = P(s_1)\mathrm{P}(s_2|s_1)\dots\mathrm{P}(s_m|s_1 s_2 \dots s_{m-1})$$

$$\approx \prod_{s_i \in S} P(s_i)$$

Unigram model

segments

Valid segment boundary: *<u>if and only if</u>* the pointwise mutual information between the two segments resulting from the split is negative:

$$MI(s_k, s_{k+1}) = \log \frac{P_c([s_k, s_{k+1}])}{P_c(s_k) \cdot P_c\ (s_{k+1})} < 0$$

Example: $\log \frac{P_c(["new", "york"])}{P_c("\ new") \cdot P_c\ ("york")} > 0$

$\underbrace{\text{new york times}}_{s_1}$ / subscription s_2

no segment boundary here

Unsupervised Segmentation

- Find top *k* segmentations: dynamic programming

Words in a query: $\underbrace{w_1, \ldots, w_{j-1}}_{\text{Recurse}}, \underbrace{w_j, \ldots, w_i}_{\text{Last segment}}$

Input: query $w_1 w_2 \ldots w_n$, concept probability distribution

Output: top k segmentations with highest likehood

- Using EM optimization on the fly

Exploit Click-through [Li et al. 2011]

- Motivation
  - Probabilistic query segmentation

    ***Input Query**: bank of america online banking*

    Output top-3 segmentation:

    {[bank of america] [online banking], 0.502},

    {bank of america online banking], 0.428},

    {[bank of] [america] [online banking], 0.001}

  - Use click-through data

click data

query

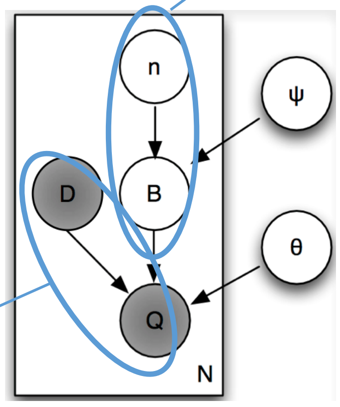
Q -> {URL} -> D

document

Exploit Click-through

- Segmentation Model

global info



Click-through info

An interpolated model:

$$P(Q|B,\theta,\theta_D) \propto \prod_{m=1}^{M} P(S_m|\theta)P(S_m|\theta_D)$$

global info

Click-through info

Clicked html documents

Query

[credit card] [bank of America]

1. bank of america credit cards contact us overview
2. secured visa credit card from bank of america
3. credit cards overview find the right bank of america credit card for you

Short Text Understanding

- How to segment this short text?
- What does this short text mean (its intent, senses, or concepts)?
- What are the relations among terms in the short text?
- How to calculate the similarity between short texts?

Sense Changes with Different Context

watch harry potter

Movie

read harry potter

Book

age harry potter

Character

harry potter walkthrough

Game

Entity Recognition in Query [Guo et al. 2009]

- Motivation

Detect named entity in a short text and categorize it

Single-named-entity query ⇨ triple <e, t, c>

ambiguous term

context terms

class of entity

Example:

harry potter walkthrough → (“harry potter”, “# walkthrough”, “game”)

term

context

class

Entity Recognition in Query

- Probabilistic Generative Model

Goal: Given a query q, find triple <e, t, c> maximize the probability

Objective: given query q, find:

$$\begin{aligned}(e,t,c)^* &= \arg\max_{(e,t,c)} \Pr(q,e,t,c) \\ &= \arg\max_{(e,t,c)} \Pr(q|e,t,c)\Pr(e,t,c) \\ &= \arg\max_{(e,t,c)\in G(q)} \Pr(e,t,c)\end{aligned}$$

Probability to generate triple:

$$\begin{aligned}\Pr(e,t,c) &= \Pr(e)\Pr(c|e)\boxed{\Pr(t|e,c)} \\ &= \underbrace{\Pr(e)\Pr(c|e)\boxed{\Pr(t|c)}}\end{aligned}$$

assume context only depends on class

E.g. "***walkthrough***" only depends on ***game***, instead of ***happy potter***

The problem then becomes how to estimate Pr(*e*), Pr(*c*/*e*) and Pr(*t*/*c*)

Entity Recognition in Query

- Probability Estimation by Learning

learning objective: $\max \prod_{i=1}^{N} P(e_i, t_i, c_i)$

Challenge: difficult as well as time consuming to manually assign class labels to named entities in queries

Build training set $T = \{(e_i, t_i)\}$, view c_i as a hidden variable

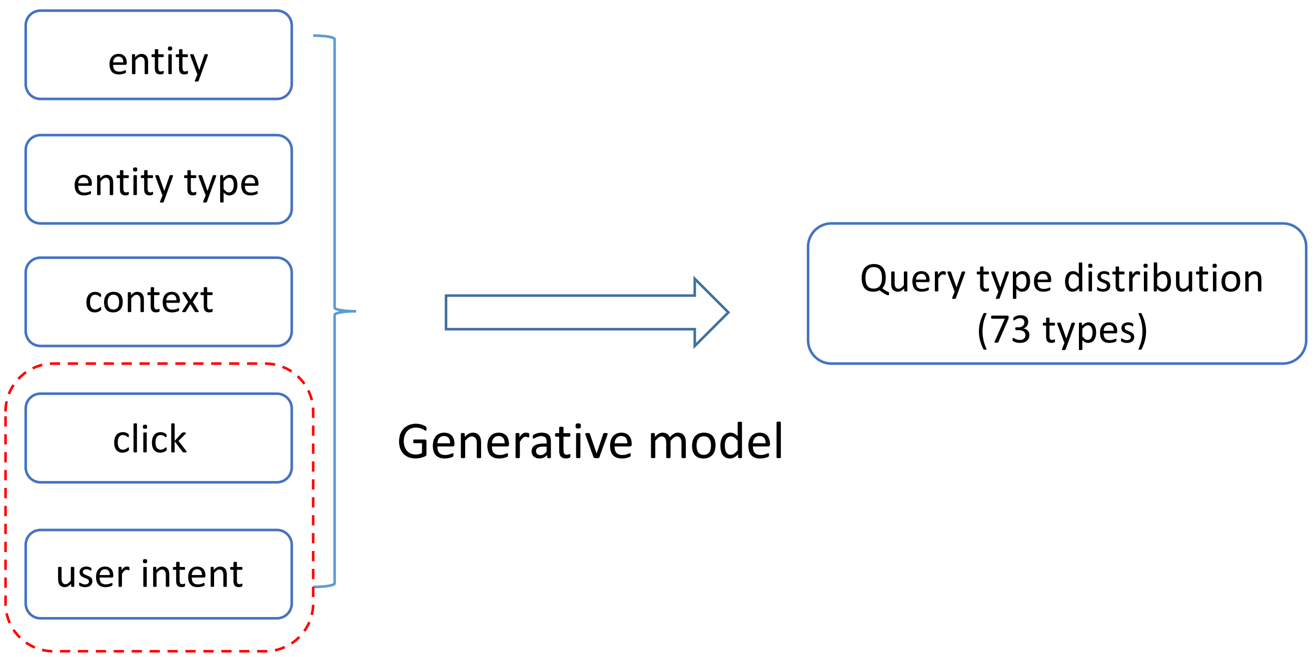
New Learning problem: $\max \prod_{i=1}^{N} P(e_i, t_i) = \max \prod_{i=1}^{N} P(e_i) \prod_{i=1}^{N} (\sum_{c} P(c \mid e_i) P(t_i \mid c))$

solved with topic model WS-LDA

Signal from Click [Pantel et al. 2012]

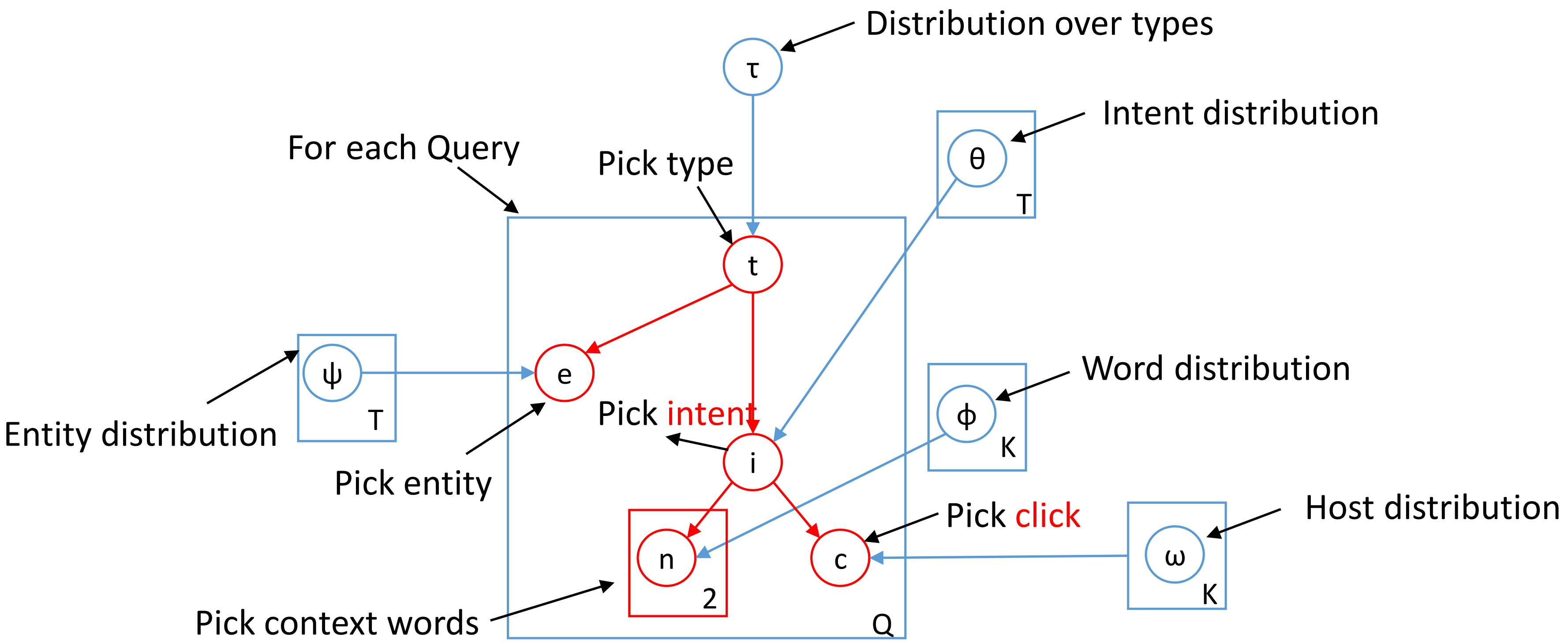
- Motivation

  Predict entity type in Web search



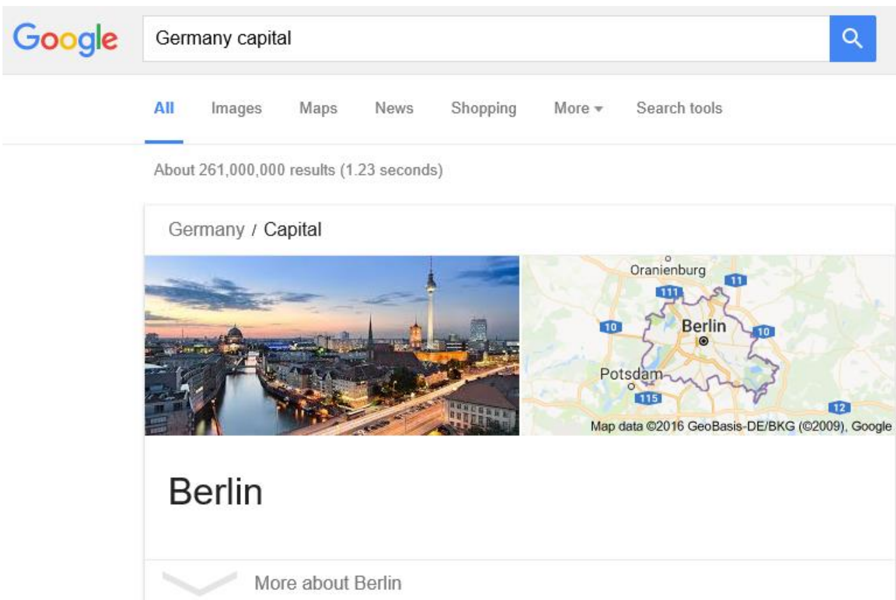
Signal from Click

- Joint Model for Prediction



Telegraphic Query interpretation
[Sawant et al. 2013, Joshi et al. 2014]

- Entity-seeking Telegraphic Queries



Query

Germany capital

Result Entity

Berlin

- Interpretation = Segmentation + Annotation

accuracy

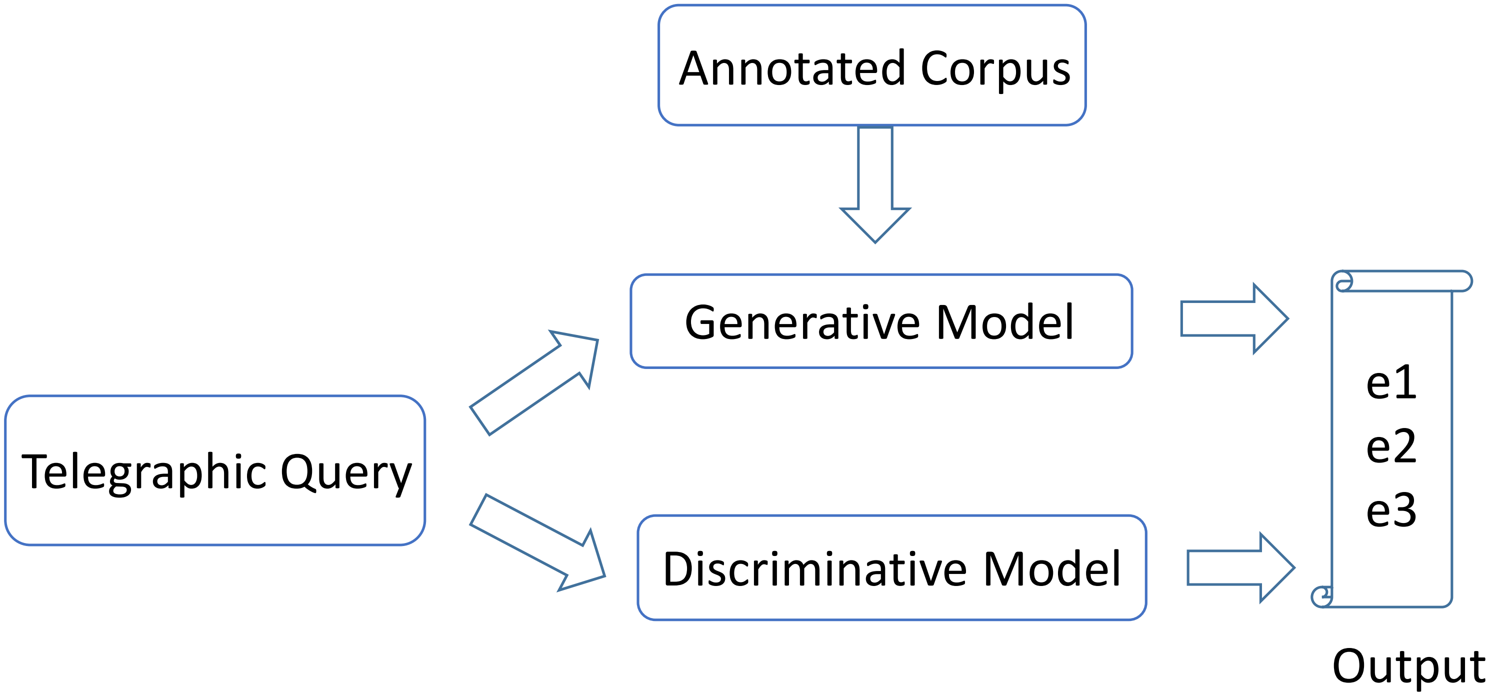
recall

Knowledge base

Large corpus

Joint Interpretation and Ranking
[Sawant et al. 2013, Joshi et al. 2014]

- Overview



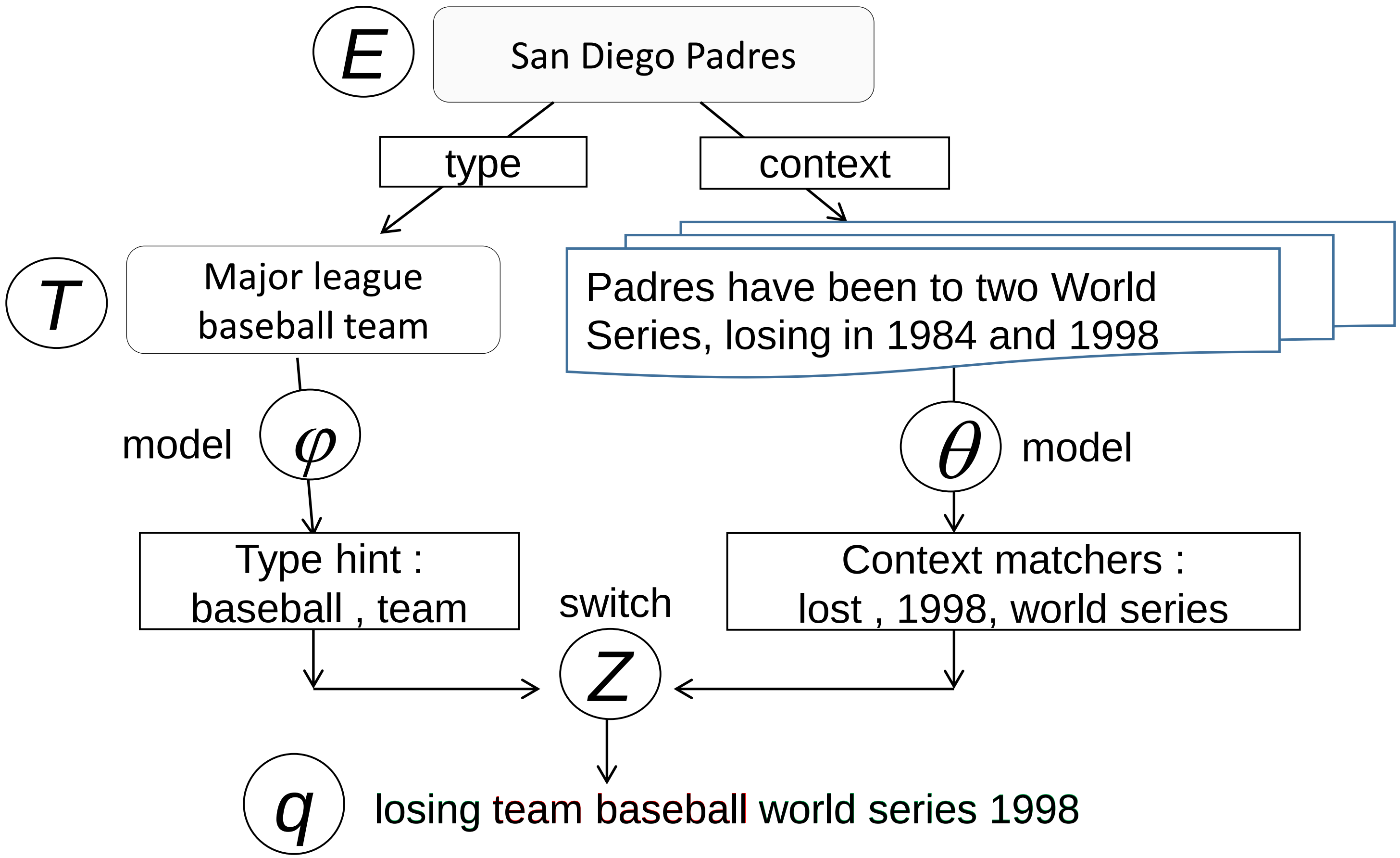
Two Models for Interpretation and Ranking

Joint Interpretation and Ranking

[Sawant et al. 2013]

- Generative Model

Based on Probabilistic Language Models



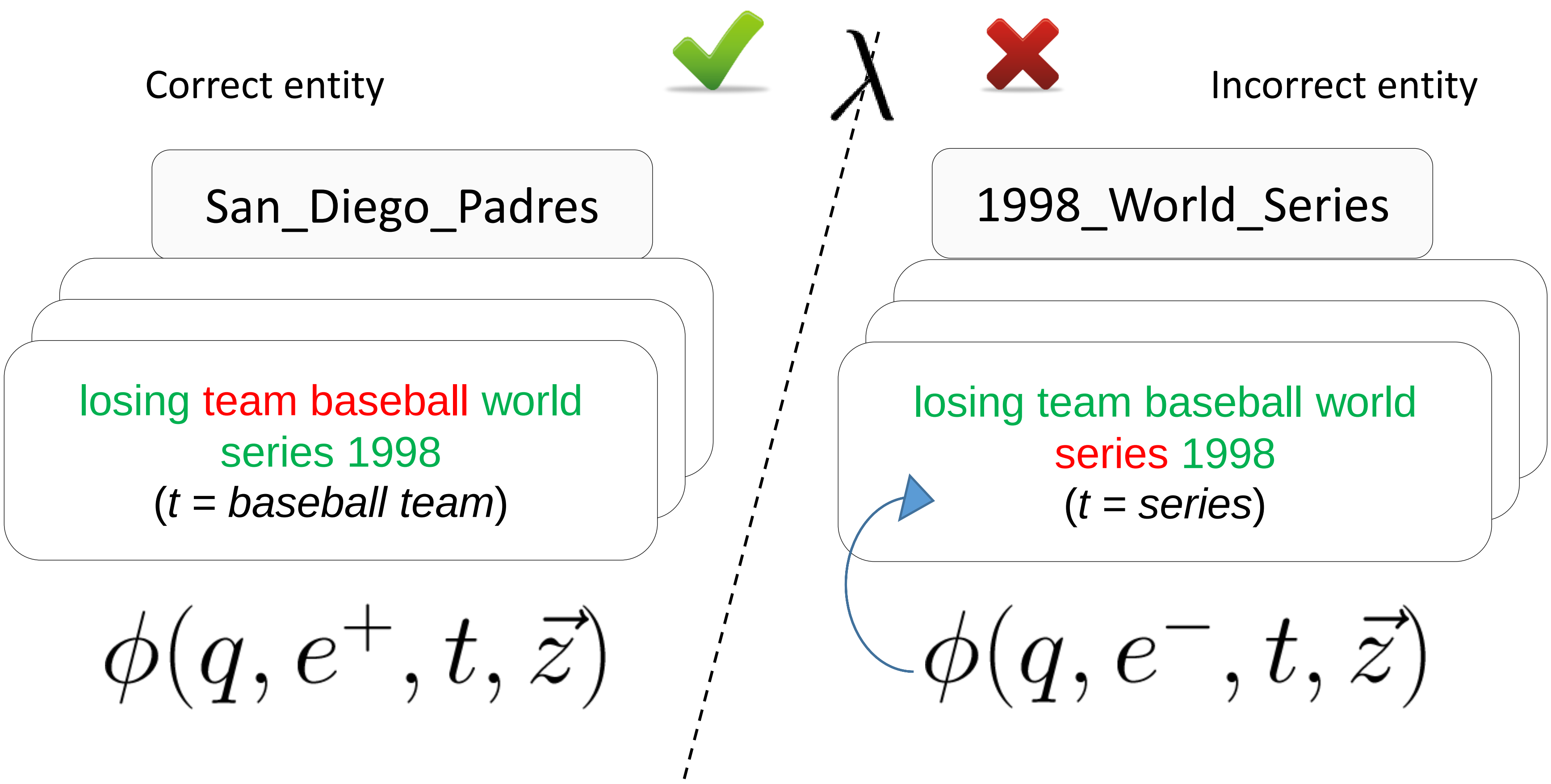
Borrow from U. Sawant (2013)

Joint Interpretation and Ranking

[Sawant et al. 2013]

- Discriminative Model

Based on max-margin discriminative learning



Telegraphic Query Interpretation

[Joshi et al. 2014]

- Queries seek answer entities (e_2)
- Contain (query) *entities* (e_1) , *target types* (t_2), *relations* (r), and *selectors* (s).

query	e_1	r	t_2	s
dave navarro first band	dave navarro	band	band	first
	dave navarro	-	band	first
spider automobile company	spider	automobile company	automobile company	-
	automobile	company	company	spider

Borrow from M. Joshi (2014)

Improved Generative Model

- Generative Model

[Sawant et al. 2013]

$$\mathrm{p}(e|q) \propto \mathrm{p}(e) \sum_{t,\vec{z}} \mathrm{p}(t|e)\mathrm{p}(\vec{z})\mathrm{p}(h(\vec{q},\vec{z})|t)\mathrm{p}(s(\vec{q},\vec{z})|e)$$

[Joshi et al. 2014]

$$\begin{aligned}\mathrm{p}(e|q) \propto\ & \Psi_{T_2}(q, z, t_2)\Psi_R(q, z, \underline{r}) \\ & \Psi_{E_1}(q, z, \underline{e_1})\Psi_S(q, z) \\ & \Psi_{E_2,T_2}(e_2, t_2)\underline{\Psi_{E_1,R,E_2}(e_1, r, e_2)} \\ & \Psi_{E_1,R,E_2,S}(q, z, \underline{e_1, r}, e_2).\end{aligned}$$

Consider e_1 (in q) and r

Improved Discriminative Model

- Discriminative Model

[Sawant et al. 2013]

$$\max_{t,\vec{z}} \lambda \cdot \phi(q, e^+, t, \vec{z}) \geq 1 - \xi_{q,e^+,e^-} + \max_{t',\vec{z}'} \lambda \cdot \phi(q, e^-, t', \vec{z}')$$

Consider e_1 (in q) and r

[Joshi et al. 2014]

$$\max_{q,z,e_1,t_2,r} w \cdot \phi(q, z, \underline{e_1,}\, t_2, \underline{r,}\, e_2^+) + \xi \geq 1 + \max_{q,z,e_1,t_2,r} w \cdot \phi(q, z, \underline{e_1,}\, t_2, \underline{r,}\, e_2^-)$$

Understand Short Texts with A Multi-tiered Model [Hua et al. 2015 (**ICDE Best Paper**)]

- Input: a short text

  wanna watch eagles band

- Output: semantic interpretation

  $\text{watch}_{\text{[verb]}}$ $\text{eagles}_{\text{[entity](band)}}$ $\text{band}_{\text{[concept]}}$

- Three steps in understanding a short text

wanna watch eagles band

Step 1: Text Segmentation – divide into a sequence of terms in vocabulary

watch eagles band

Step 2: Type detection – determine the best type of each term

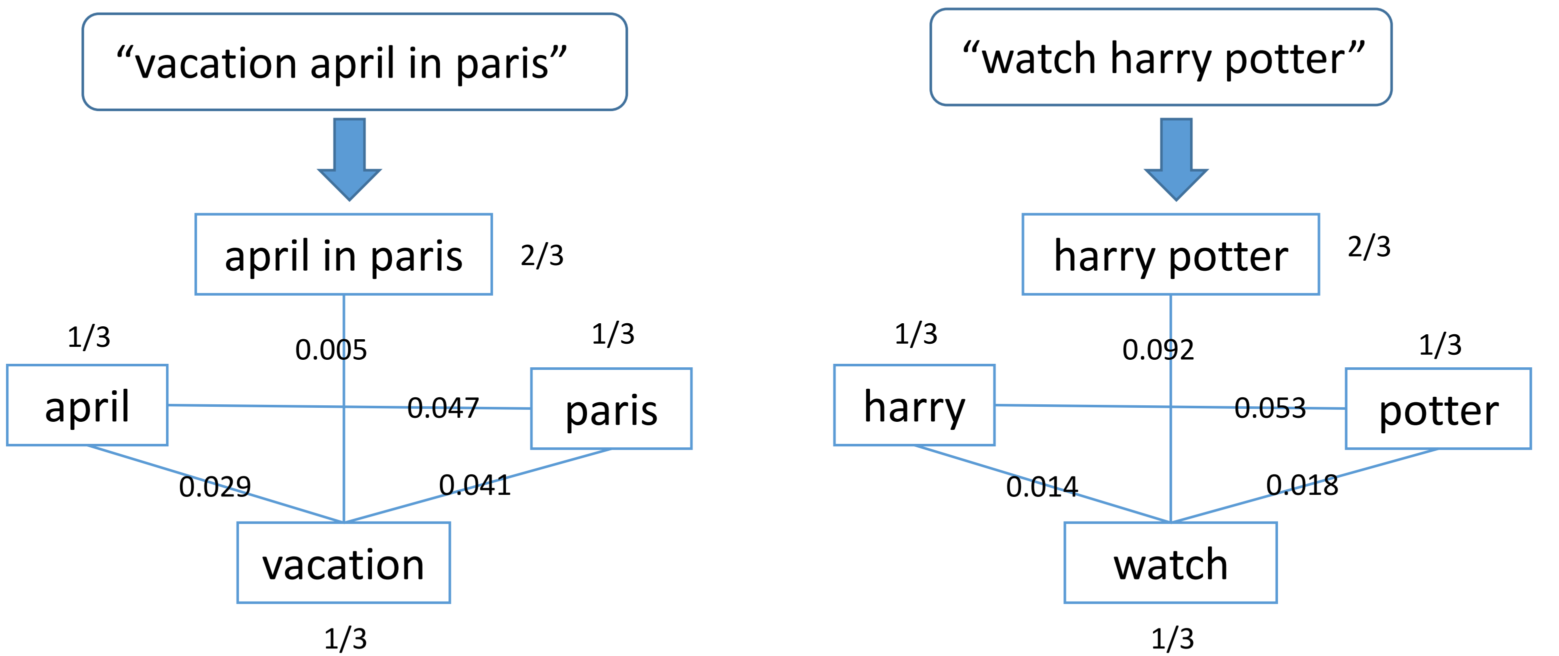
$\text{watch}_{\text{[verb]}}$ $\text{eagles}_{\text{[entity]}}$ $\text{band}_{\text{[concept]}}$

Step 3: Concept Labeling – infer the best concept of each entity within context

$\text{watch}_{\text{[verb]}}$ $\text{eagles}_{\text{[entity](band)}}$ $\text{band}_{\text{[concept]}}$

Text segmentation

- Observations:
  - Mutual Exclusion – terms containing the same word mutually exclude each other
  - Mutual Reinforcement – related terms mutually reinforce each other
- Build a Candidate Term Graph (CTG)

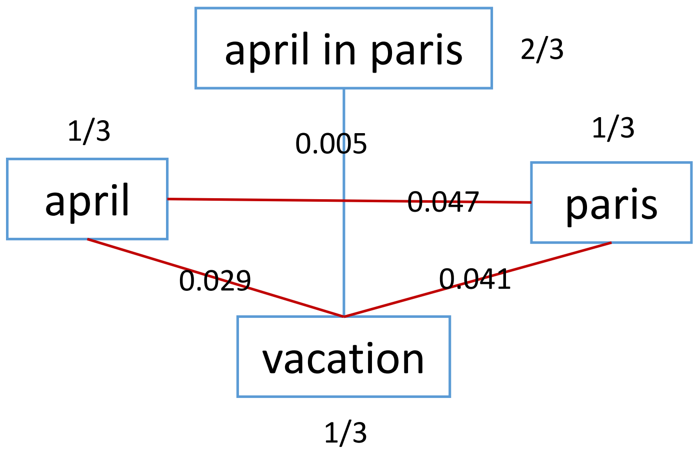


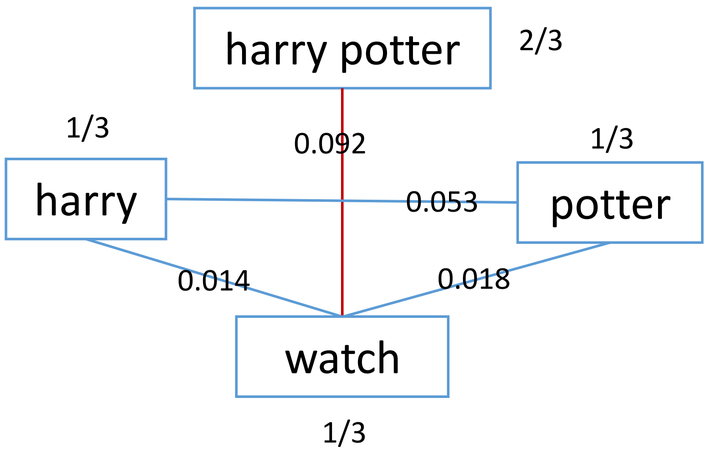
Find best segmentation

- Best segmentation= sub-graph in CTG which
  - Is a complete graph (clique)
    - No mutual exclusion
  - Has 100% word coverage
    - Except for stopwords
  - Has the largest average edge weight

Is a segmentation

Best segmentation



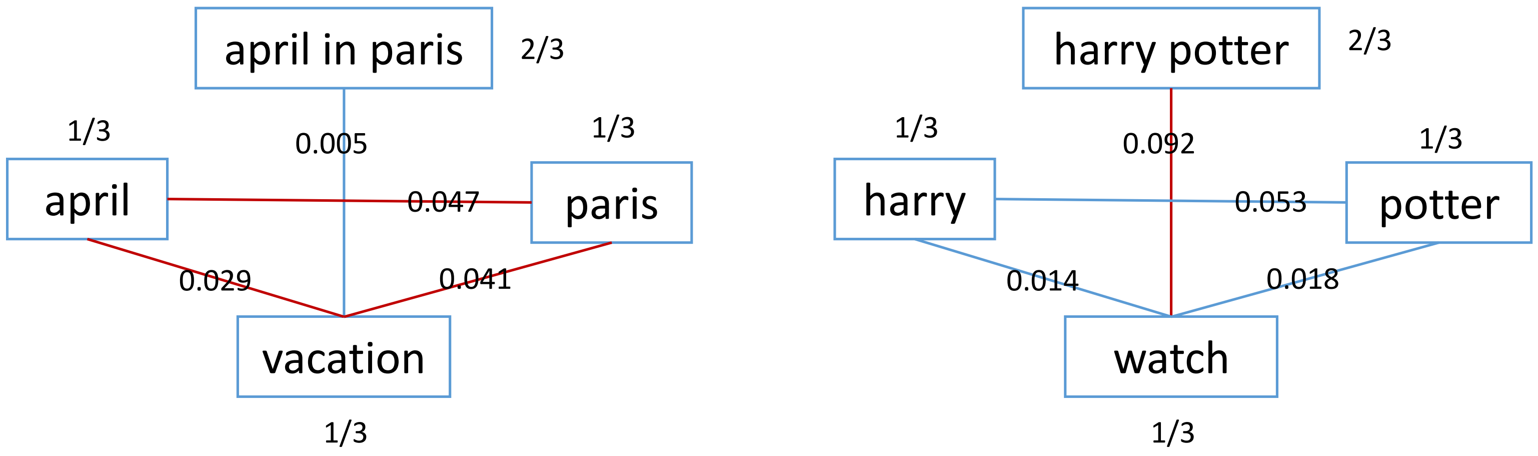


Find best segmentation

- Best segmentation= sub-graph in CTG which
  - Is a complete graph (clique)
    - No mutual exclusion
  - Has 100% word coverage
    - Except for stopwords
  - Has the largest average edge weight

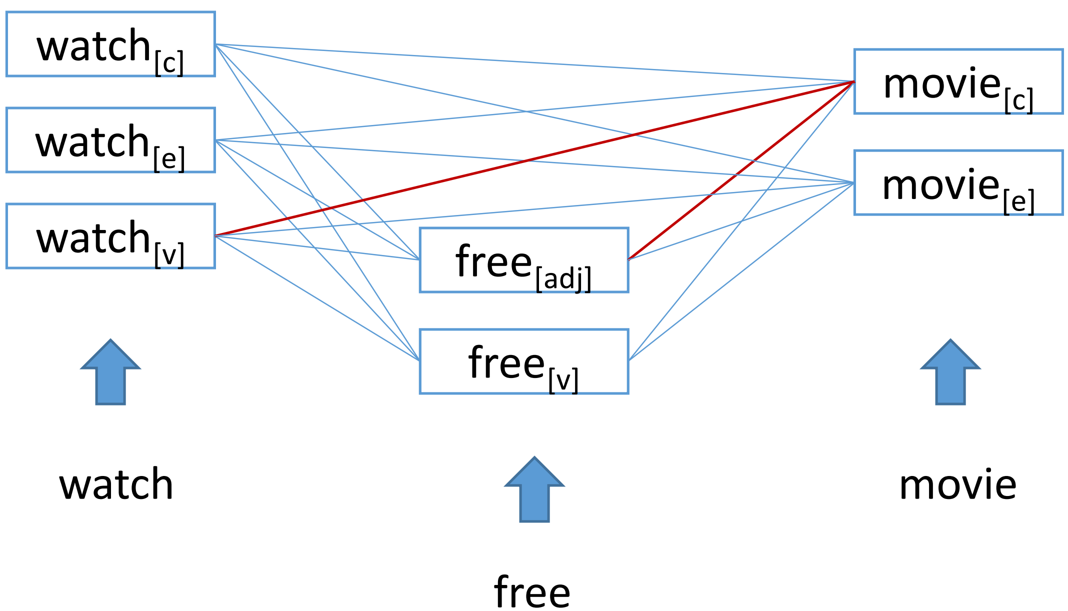
Maximal Clique

Best segmentation



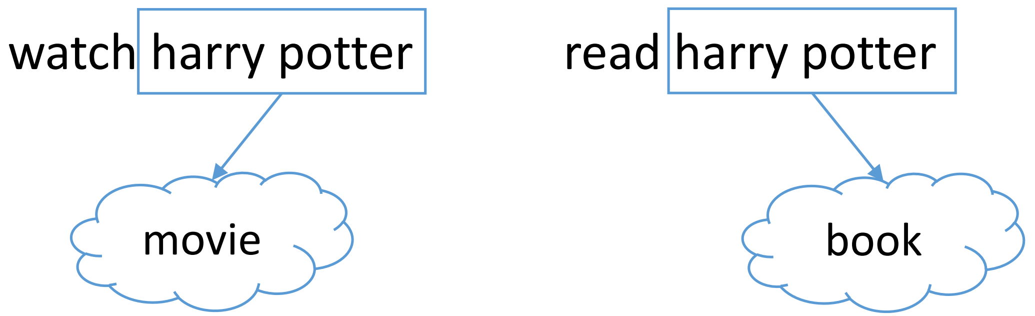
Type Detection

- Pairwise Model
  - Find the best typed-term for each term, so that the Maximum Spanning Tree of the resulting sub-graph between typed-terms has the largest weight



Concept Labeling

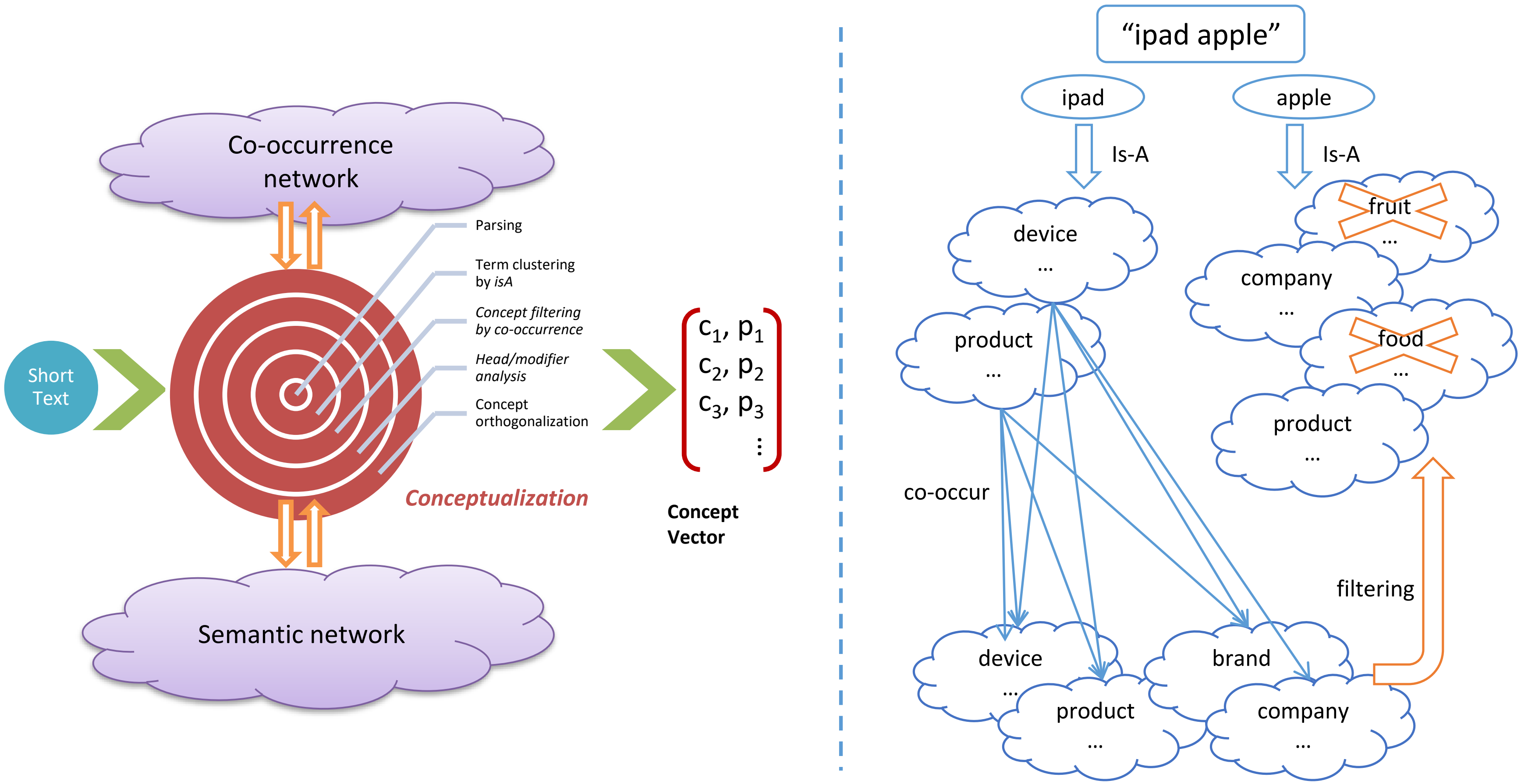
- Entity disambiguation is the most important task of concept labeling
  - Filter/re-rank of the original concept cluster vector



- Weighted-Vote
  - The final score of each concept cluster is a combination of its original score and the support from context using concept co-occurrence

Example of Entity Disambiguation

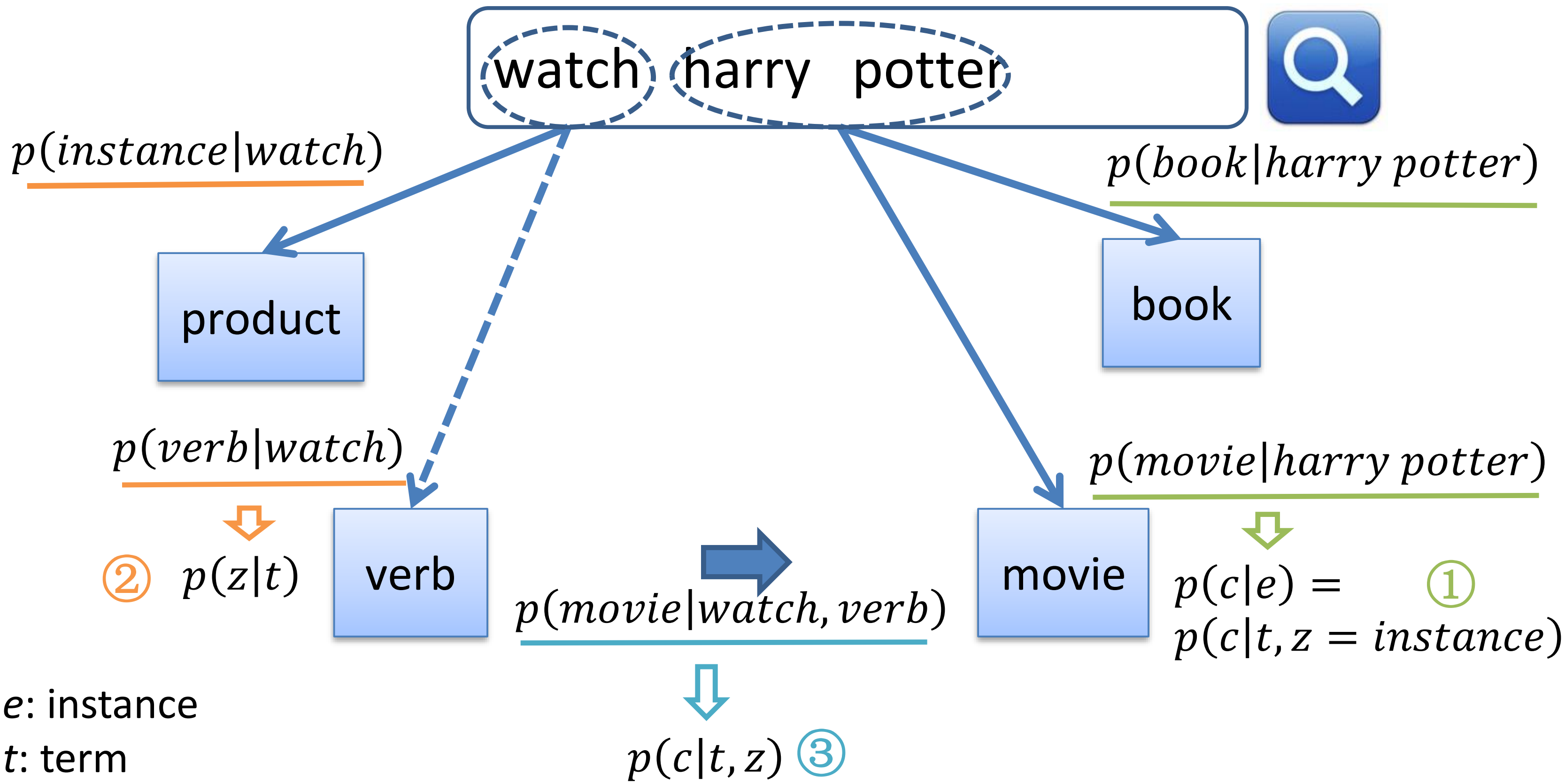
[Hua et al. 2015 (**ICDE Best Paper**), Hua et al. 2016]



Mining Lexical Relationships

[Wang et al. 2015b]

- Lexical knowledge represented by the probabilities



e: instance

t: term

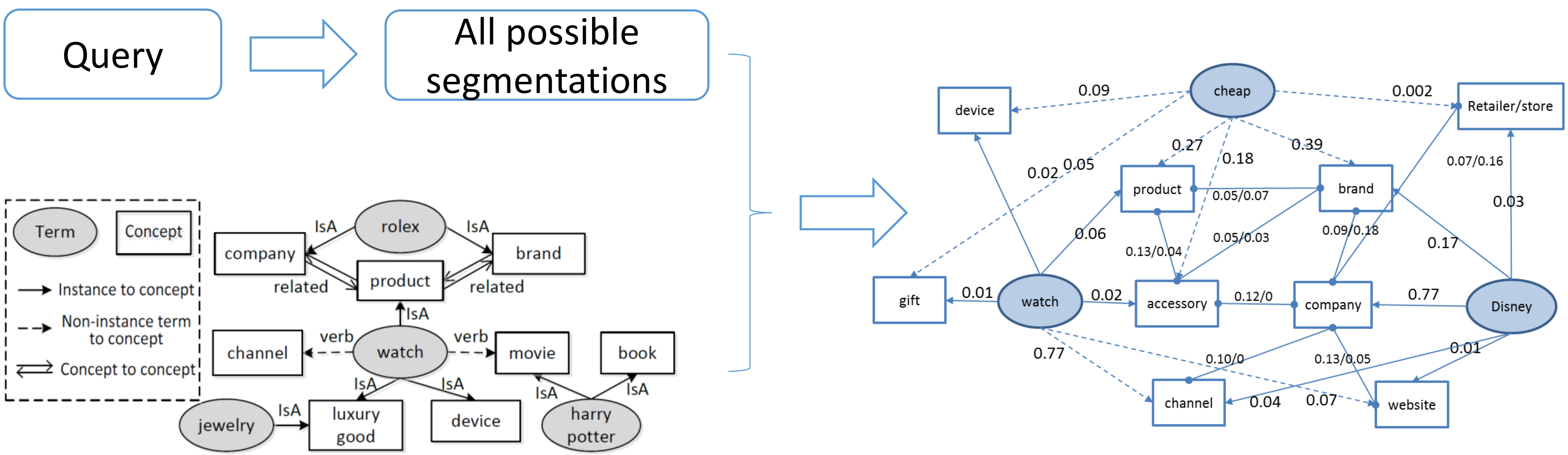
c: concept

z: role

Understanding Queries [Wang et al. 2015b]

- **Goal**: to rank the concepts and find:

$$\arg\max_{c} p(c|t,q)$$



The offline semantic network

Random walk with restart [Sun et al., 2005] on the online subgraph

Short Text Understanding

- How to segment this short text?

- What does this short text mean (its intent, senses, or concepts)?

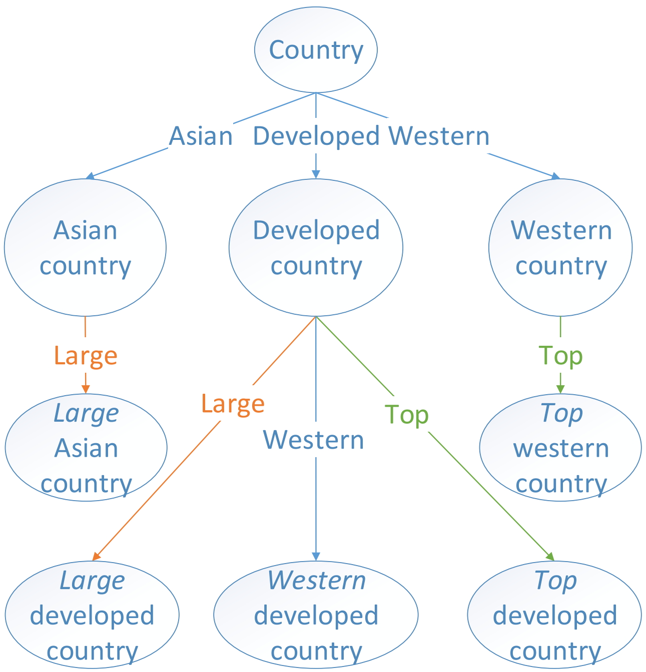
- What are the relations among terms in the short text?

- How to calculate the similarity between short texts?

Head, Modifier, and Constraint Detection in Short Texts [Wang et al. 2014b]

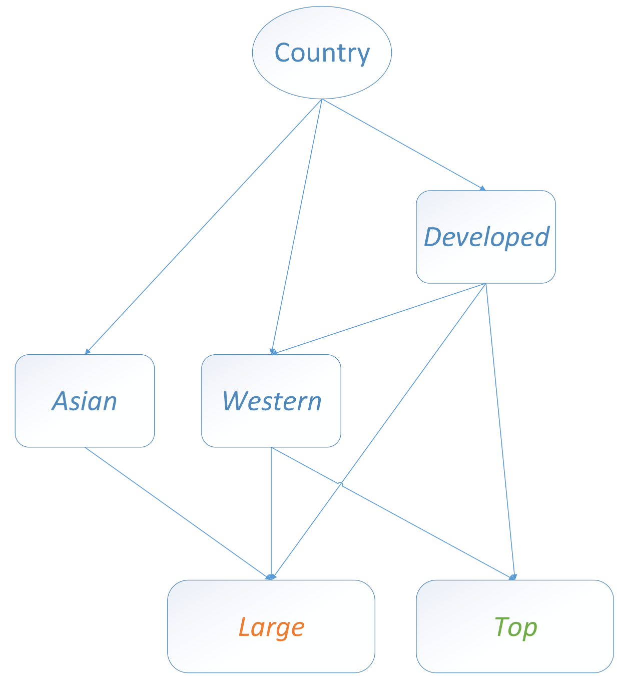
- Example: “popular smart cover iphone 5s”
- **Definition**:
  - ***Head*** acts to name **the general (semantic) category** to which the whole short text belongs. Usually, the *head* is the **intent** of the short text.
    - “smart cover”: intent of the query
  - ***Constraints*** **distinguish** this member from other members of the same category.
    - “iphone 5s”: limit the type of the head
  - ***Non-Constraint Modifiers (a.k.a. Pure Modifiers)*** are **subjective** modifiers which can be dropped without changing intent
    - “popular”: subjective, can be neglected

Non-Constraint Modifiers Mining: Construct Modifier Networks



Concept Hierarchy Tree in "Country" domain

Edges form a Modifier Network



Modifier Network in "Country" domain
In this case, "Large" and "Top" are pure modifiers

Non-Constraint Modifiers Mining: Betweenness centrality

- **Betweenness centrality** is a measure of a node's centrality in a network

- Betweennes of node *v* is defined as:

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

  - where σ_{st} is the total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of those paths that pass through *v*

- Normalization & Aggregation

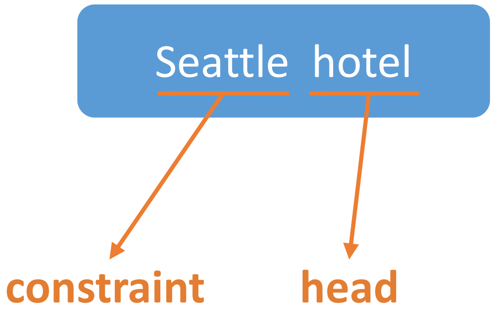
$$NL(g(v)) = \log \frac{g(v) - min(g)}{max(g) - min(g)} \Rightarrow PMS(t) = \sum NL(g(t))$$

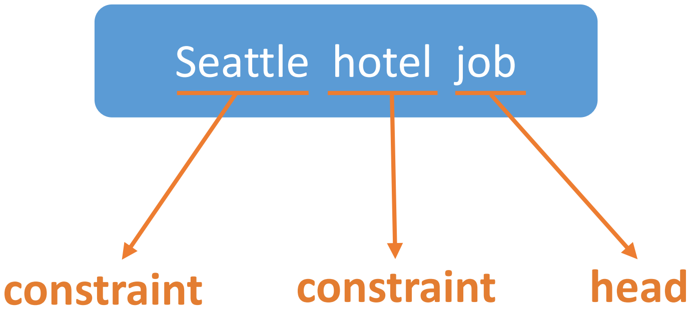
- For a pure modifier, it should have low betweenness centrality aggregation score *PMS(t)*

Head-Constraints Mining [Wang et al. 2014b]

- A term can be a *head sometimes*, and be a *constraint in some other cases*

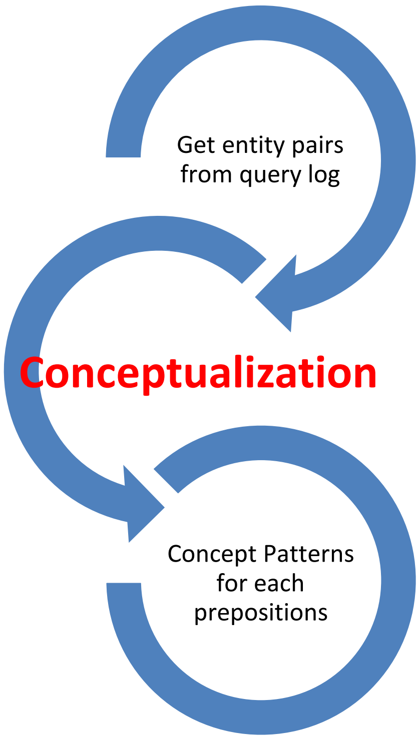
- E.g.

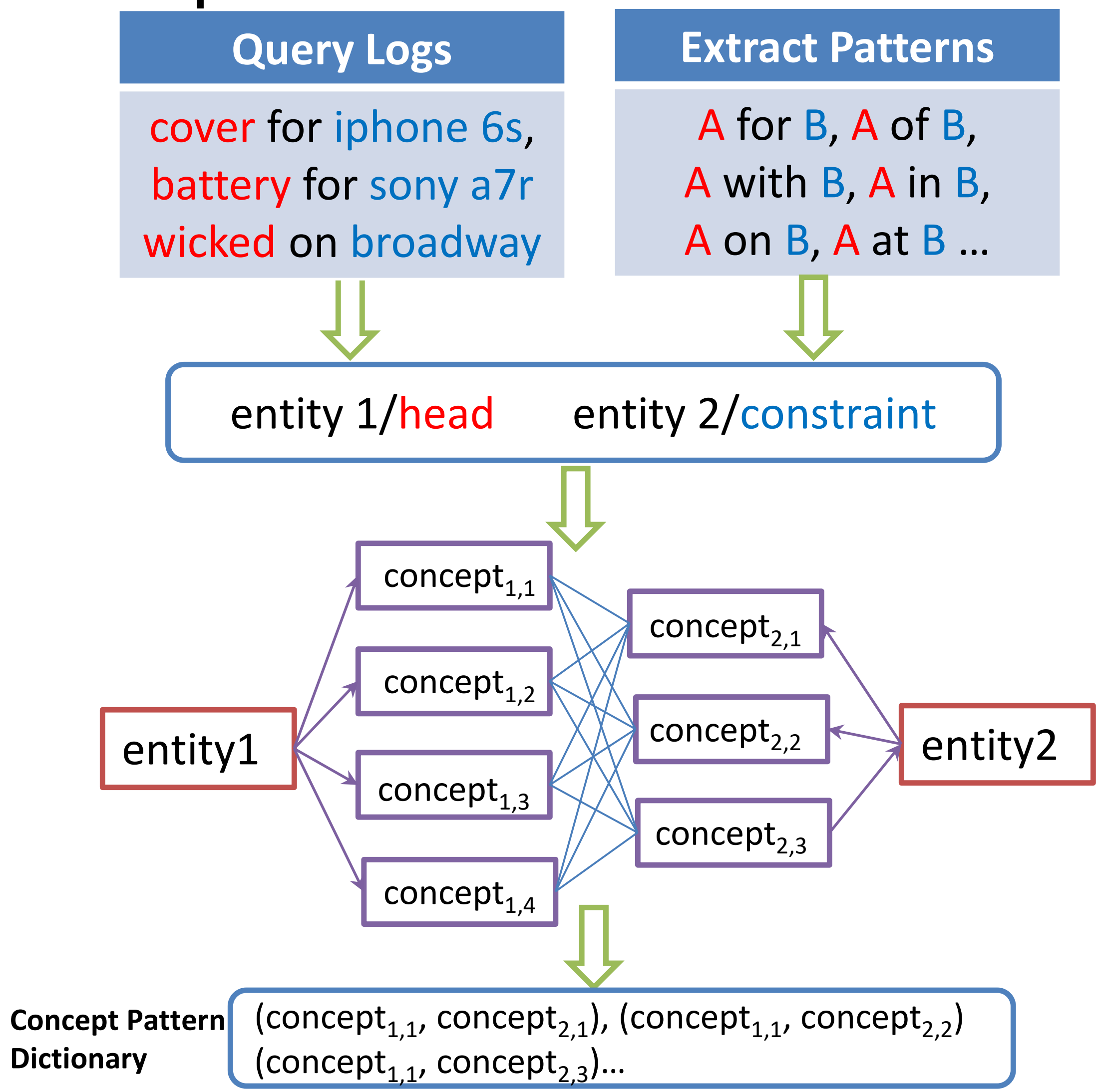




Head-Constraints Mining: Acquiring Concept Patterns

Building concept pattern dictionary:





Why Concepts Can't Be Too General

- It may cause too many concept pattern conflicts: can't distinguish head and modifier for general concept pairs

	Head	Modifier
Derived Concept Pattern	device	company
Supporting Entity Pairs	iphone 4	verizon
	modem	comcast
	wireless router	comcast
	iphone 4	tmobile

Conflict

	Head	Modifier
Derived Concept Pattern	company	device
Supporting Entity Pairs	amazon books	kindle
	netflix	touchpad
	skype	windows phone
	netflix	ps3

Why Concepts Can't Be Too Specific

- It may generate concepts with little coverage

…	…
device	largest desktop OS vendor
device	largest software development company
device	largest global corporation
device	latest windows and office provider
…	…

- Concept regresses to entity
  - Large storage space: up to (million * million) patterns

Basic-level Conceptualization (BLC) is a good choice [Wang et al. 2015b]

Top Concept Patterns

Cluster size	Sum of Cluster Score	head;constraint;score
615	21146.91	breed;state;3572.98460224501
296	7752.357	game;platform;627.403476771856
153	3466.804	accessory;vehicle;533.93705094809
70	1182.59	browser;platform;132.612807637391
22	1010.993	requirement;school;271.407526294823
34	948.9159	drug;disease;154.602405333541
42	899.2995	cosmetic;skin condition;81.4659415003929
16	742.1599	job;city;279.03732555528
32	710.403	accessory;phone;246.513830851194
18	669.2376	software;platform;210.126322725878
20	644.4603	test;disease;239.774028397537
27	599.4205	clothes;breed;98.773996282851
19	591.3545	penalty;crime;200.544192793488
25	584.8804	tax;state;240.081818612579
16	546.5424	sauce;meat;183.592863621553
18	480.9389	credit card;country;142.919087972152
14	473.0792	food;holiday;145.54140330924
11	453.6199	mod;game;257.163856882439
29	435.0954	garment;sport;47.1533326845442
23	399.4886	career information;professional;73.2726483731257
15	386.065	song;instrument;128.189481818135
18	378.213	bait;fish;78.0426514113169
22	372.2948	study guide;book;50.8339765053921
19	340.8953	plugins;browser;55.0326072627126
14	330.5753	recipe;meat;88.2779863422951
18	321.4226	currency;country;110.825444188352
13	318.0272	lens;camera;186.081673263957
9	316.973	decoration;holiday;130.055844126533
16	314.875	food;animal;73.38544366514

game	platform
game	device
video game	platform
game	console game pad
game	gaming platform

Detection

Game (Head)	Platform (Modifier)
angry birds	android
angry birds	ios
angry birds	windows 10
…	…

Head Modifier Relationship

- Train a classifier on

  (head-embedding, modifier-embedding)

- Training data:
  - Positive (head, modifier)
  - Negative (modifier, head)

- Precision >= 0.9, Recall >= 0.9

- Disadvantage: not interpretable

Syntactic Parsing based on HM

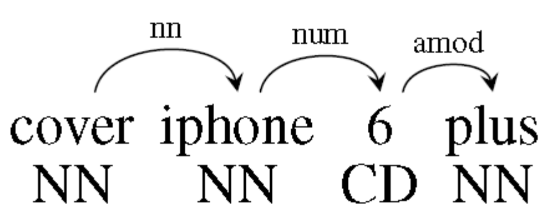
- Information is incomplete:
  - Preposition, and other function words
  - Within a noun compound: el capitan, macbook pro

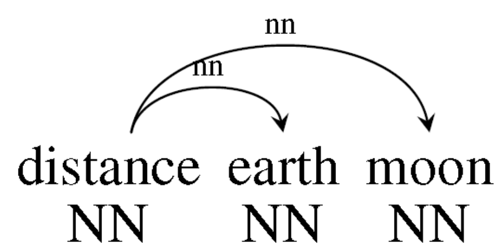
- Why not train a parser for web queries?

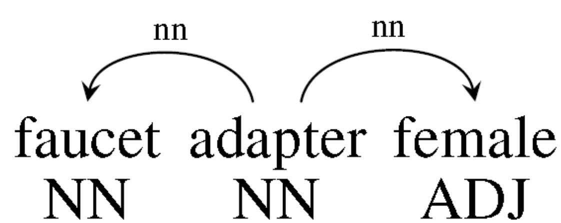
Syntactic Parsing of Short Texts

[Sun et al. EMNLP 2016]

- Syntactic structures are valuable for short text understanding
- Examples:

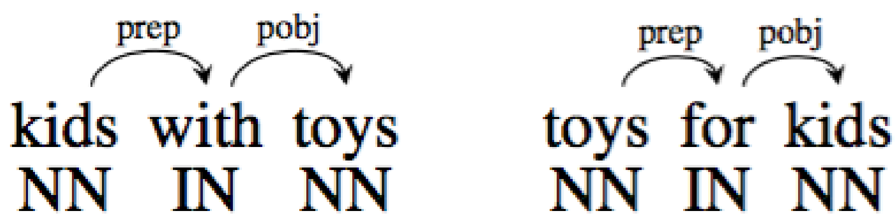






Challenges: Short Texts Lack Grammatical Signals

- Lack function words, word order
- "toys queries" has ambiguous intent



- "distance earth moon" has clear intent
  - many equivalent forms: "earth moon distance", "earth distance moon", …

Challenges: Syntactic Parsing of Queries

- No standard

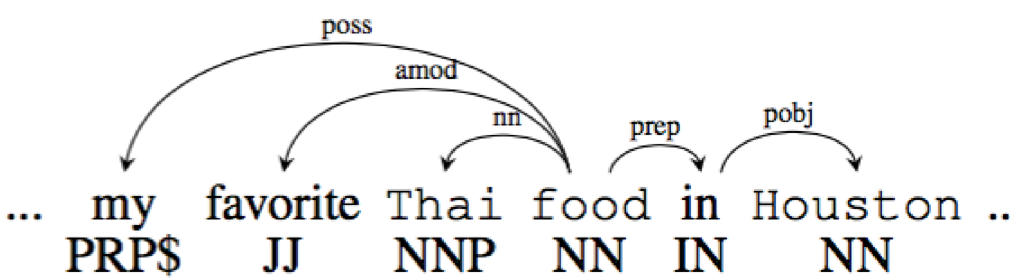
- No ground-truth

Why is syntactic parsing of queries even a legitimate problem?

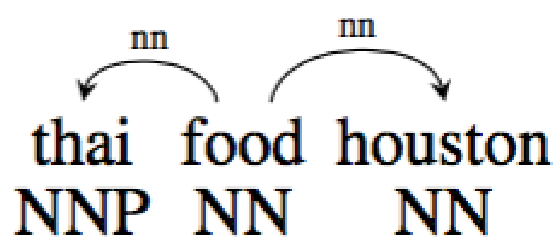
Derive Syntax from Semantics

[Sun et al. 2016]

- Query “thai food houston”
- Clicked sentence:



- Project dependency to the query:



A Treebank for Short Texts

- Given query q
- Given q's clicked sentence $\{s\}$
- Parse each s
- Project dependency from s to q
- Aggregate dependencies

Algorithm of Projection

1. Let T_s denote all the subtrees of the dependency tree of s.

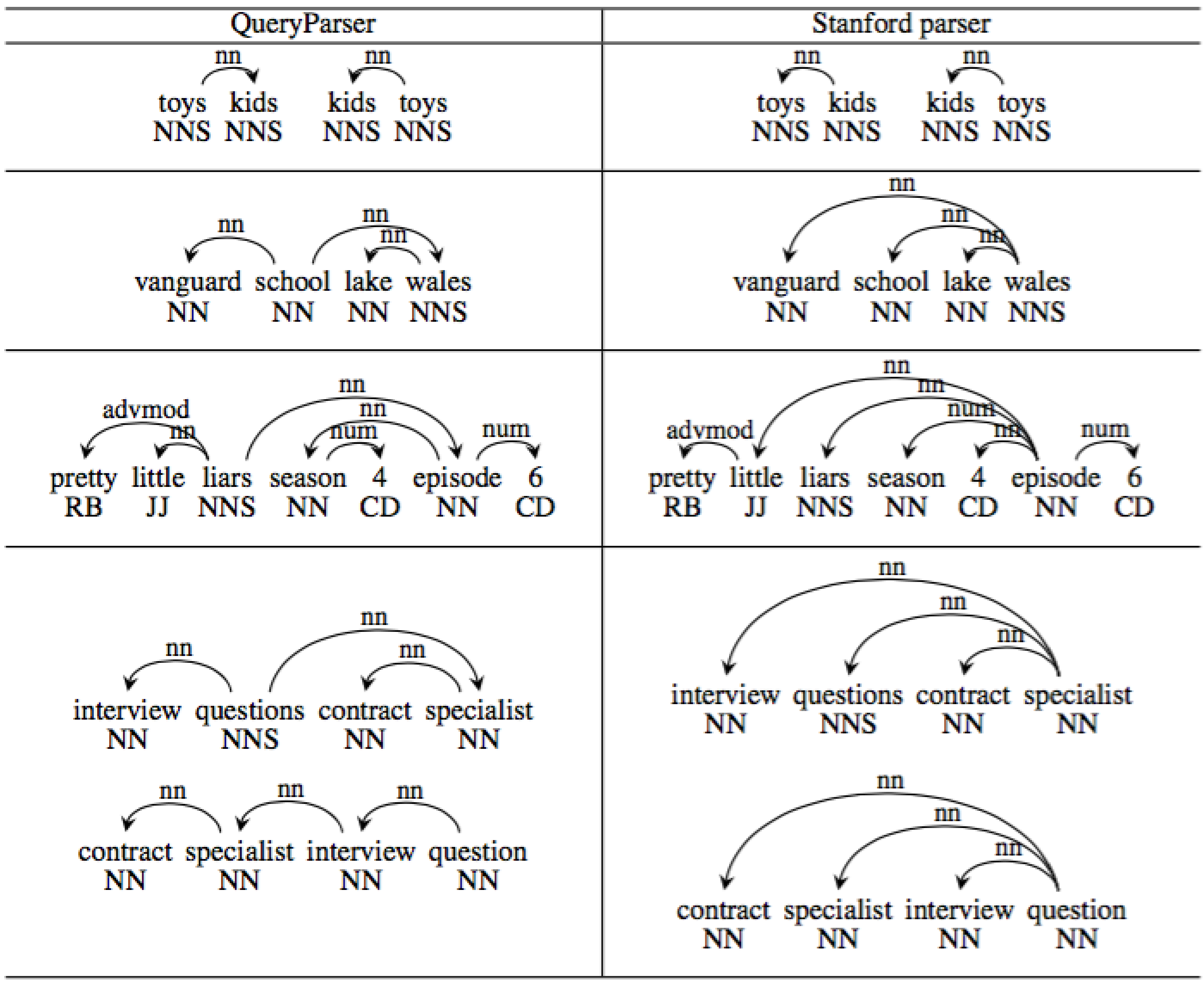
2. Find the minimum subtree $t \in T_s$ such that each $x \in q$ has one and only one match $x' \in t$.

3. Derive dependency tree $t_{q,s}$ for q from t as follows. For any two tokens x and y in q:

   (a) if there is an edge from x' to y' in t, we create a same edge from x to y in $t_{q,s}$.

   (b) if there is a path[1] from x' to y' in t, we create an edge from x to y in $t_{q,s}$, and label it temporarily as *dep*.

Result Examples



Results

- Random queries:

QueryParser	UAS: 0.83, LAS: 0.75
Stanford	UAS: 0.72, LAS: 0.64

- Queries with no function words:

QueryParser	UAS: 0.82, LAS: 0.73
Stanford	UAS: 0.70, LAS: 0.61

- Queries with function words:

QueryParser	UAS: 0.90, LAS: 0.85
Stanford	UAS: 0.86, LAS: 0.80

Short Text Understanding

- How to segment this short text?
- What does this short text mean (its intent, senses, or concepts)?
- What are the relations among terms in the short text?
- How to calculate the similarity between short texts?

Short Text Similarity Using Word Embedding [Kenter and Rijke 2015]

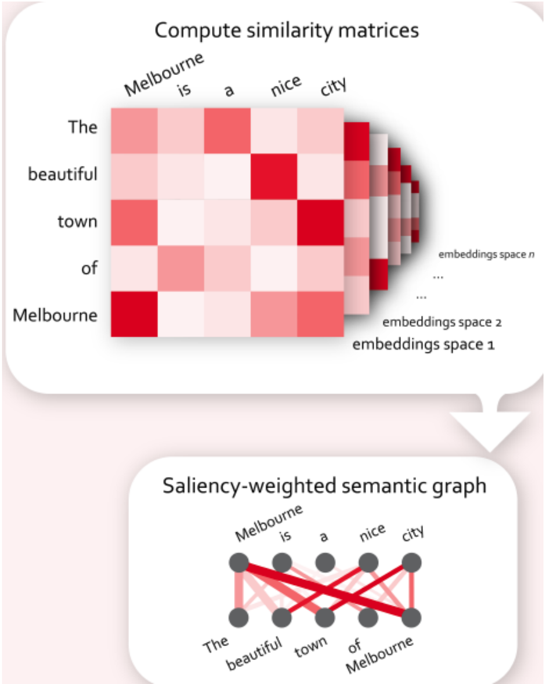
- Measuring **similarity** between two short texts and sentences

Melbourne is a nice city	✓	The beautiful town of Melbourne
Melbourne is a nice city	✗	Sydney is close to Melbourne

- Basic idea: word-by-word comparison using **embedding vector**

Short Text Similarity Using Word Embedding[Kenter and Rijke 2015]

- Use saliency-weighted semantic graph to computer similarity



Features acquired:

✓ Bins of all edges
✓ Bins of max edges

Inspired by BM25

Similarity measurement:

$$f_{sts}(s_l, s_s) = \sum_{w \in s_l} IDF(w) \cdot \frac{sem(w, s_s) \cdot (k_1 + 1)}{sem(w, s_s) + k_1 \cdot \left(1 - b + b \cdot \frac{|s_s|}{avgl}\right)}$$

Short texts

term

Semantic similarity

From the Concept View

Keyword 1: apple ipad
Keyword 2: apple pie
Automatically detect instances: ☑ Popular size: 0
Match

Semantic Similarity Score: 0.00582097483869518

Step1 & 2: Parse & Conceptualization

apple ipad
(Head)

tablet	0.00117904459974019
tablet computer	0.00104731940666755
tablet device	0.000988875154511743
huge device	0.00057990212160024
medium tablet	0.000578601568623813
popular tablet	0.000499126528574994
Tablet PCs	0.000497265042267529
mobile device	0.000496058818023634
large device	0.000494071146245059
popular device	0.000488400488400488

apple pie
(Head)

iconic american dish	0.000973558583505736
dessert recipe	0.000857356039315315
time-honored classic	0.000825939357256928
dessert	0.000764697250267525
characteristic dish	0.000643185494083393
scent	0.000602893890675241
best new game	0.000517363657506097
classic dessert	0.000492610837438424
traditional dessert	0.000487567040468064
pie	0.000481579581025764

Keyword 1: angry birds
Keyword 2: kinectimals
Automatically detect instances: ☑ Popular size: 0
Match

Semantic Similarity Score: 0.857712045881388

Step1 & 2: Parse & Conceptualization

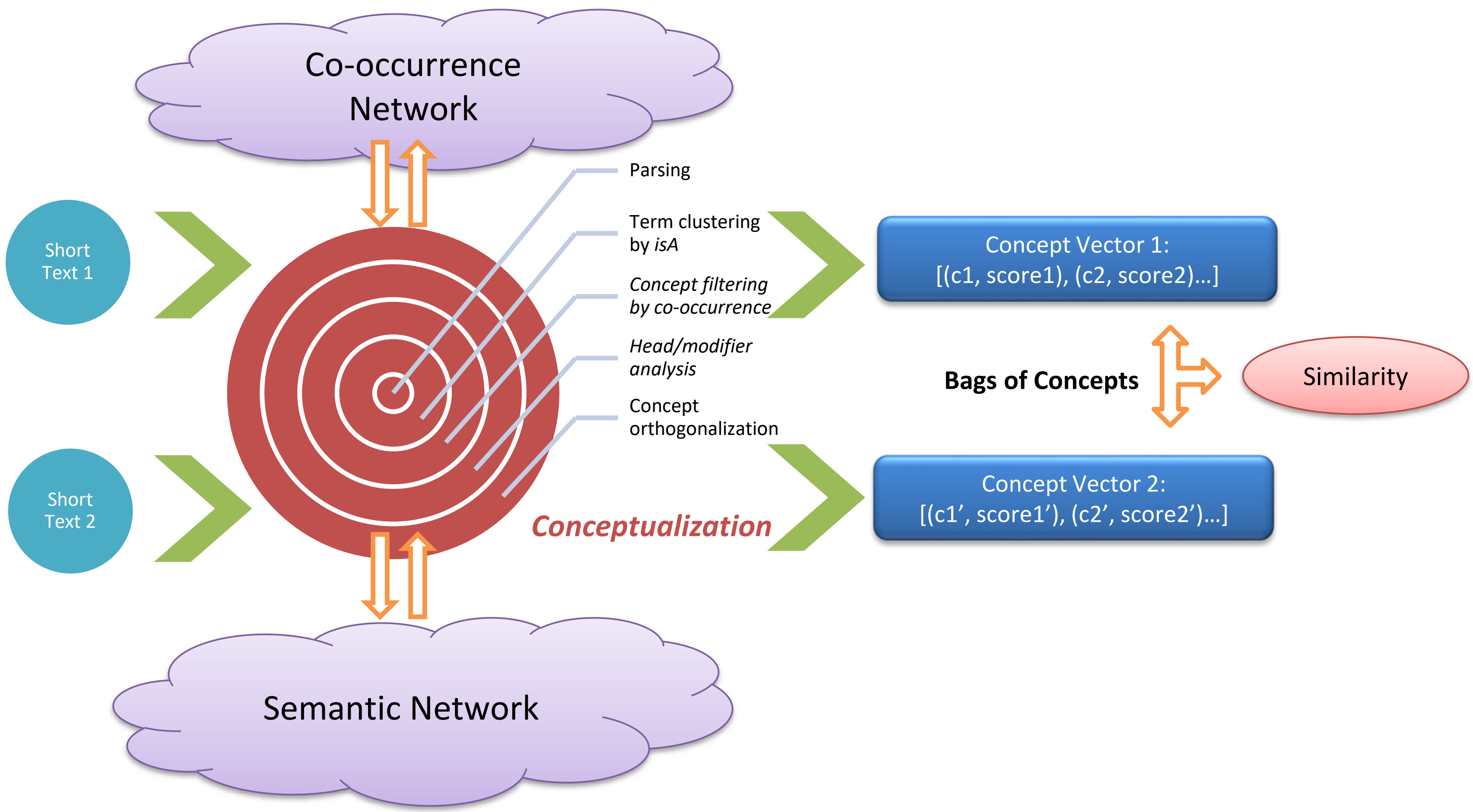
angry birds
(Head)

popular apps	0.000808058403037549
popular game	0.000759157335357753
apps	0.00072838441024954
popular phone game	0.000644789848022239
Gaming apps	0.000579757327063012
simple game	0.000536986145903645
android game	0.00049862877088008
Android apps	0.000495785820525533
casual game	0.000485908649173955
app	0.00045526974732529

kinectimals
(Head)

kinect game	0.000644629052548917
launch title	0.000393975267392781
first-party kinect game	0.000249875062468766
wii-esque game	0.00024981264051961
kinect-specific title	0.00024981264051961
first-party title	0.000249500998003992
basic game	0.00024740227610094
xbox game	0.000247157686604053
title	0.000139659760155662
game	0.000112289231365296

From the Concept View [Wang et al. 2015a]



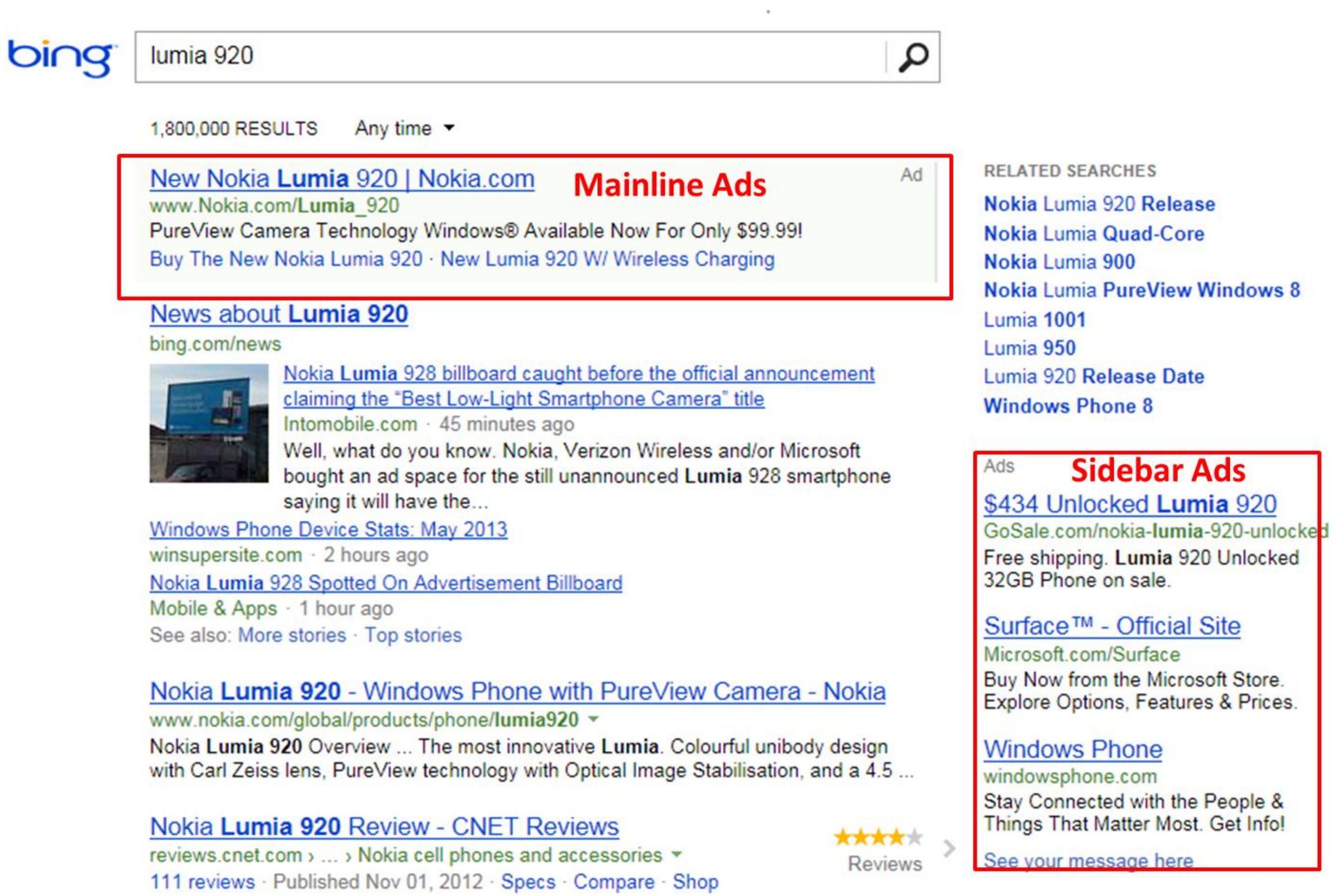
Outline

- Knowledge Bases
- Explicit Representation Models
- Applications

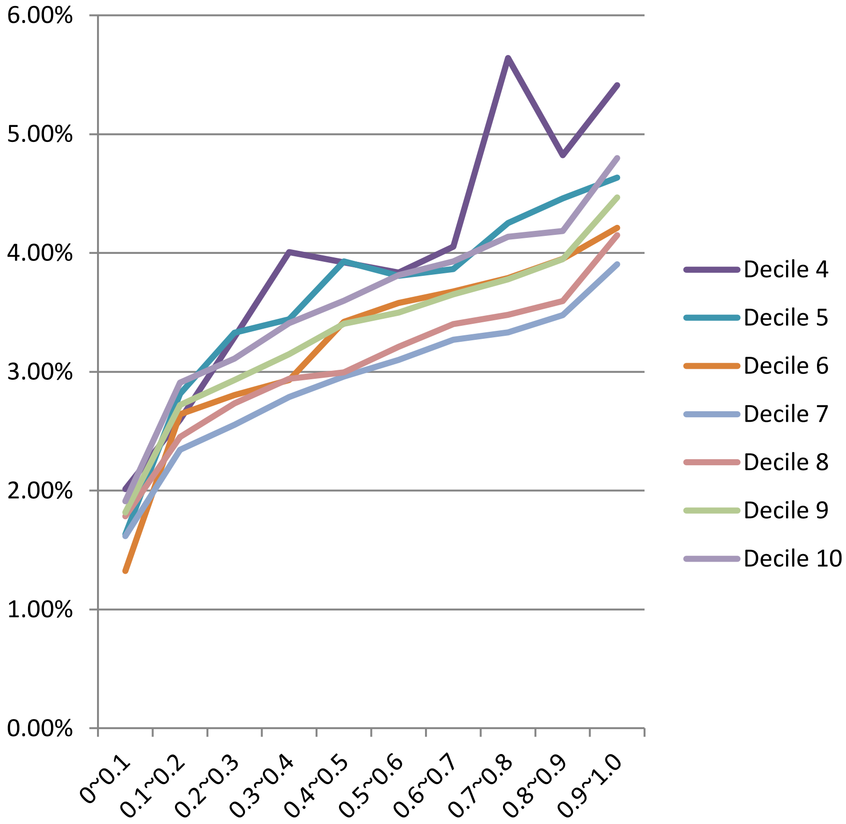
Applications

- Explicit short text understanding benefit lot of application scenarios:
  - Ads/search semantic match
  - Definition mining
  - Query recommendation
  - Web table understanding
  - Semantic search
  - …

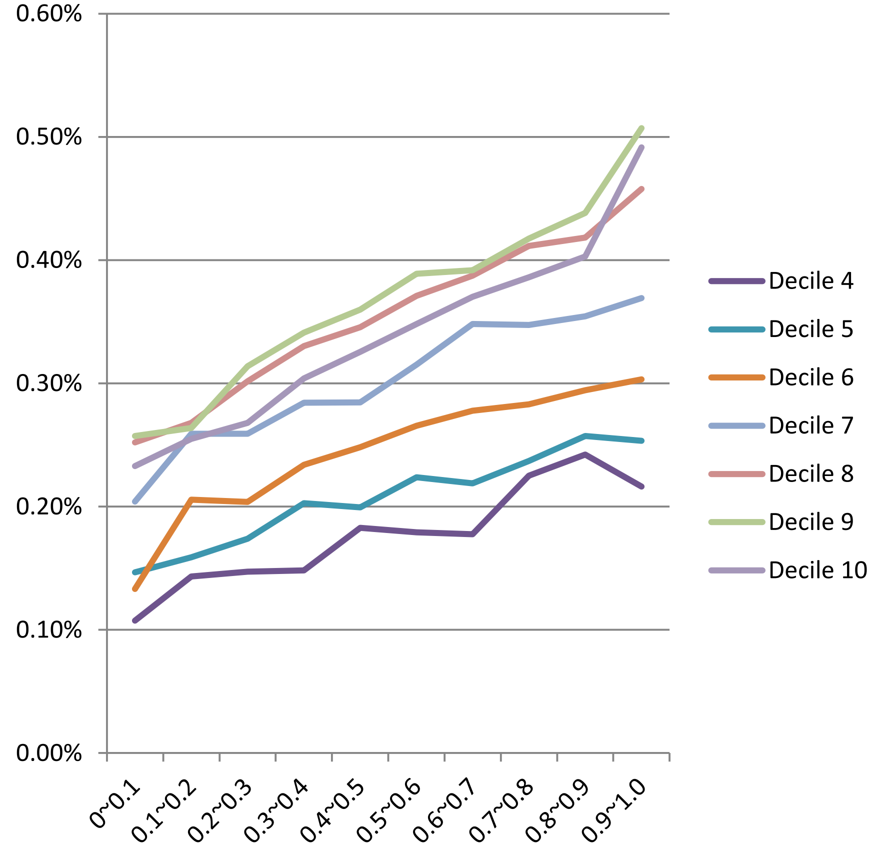
Ads Keyword Selection [Wang et al. 2015a]



Ads Keyword Selection [Wang et al. 2015a]



Mainline Ads



Sidebar Ads

Definition Mining [Hao et al. 2016]

- Definition scenarios: search engines, QnA, etc.
- Why Conceptualization is useful for definition mining?
  - Examples: “What is Emphysema”

Answer 1:

Emphysema is a disease largely associated with smoking and strikes about 2 million Americans each year.

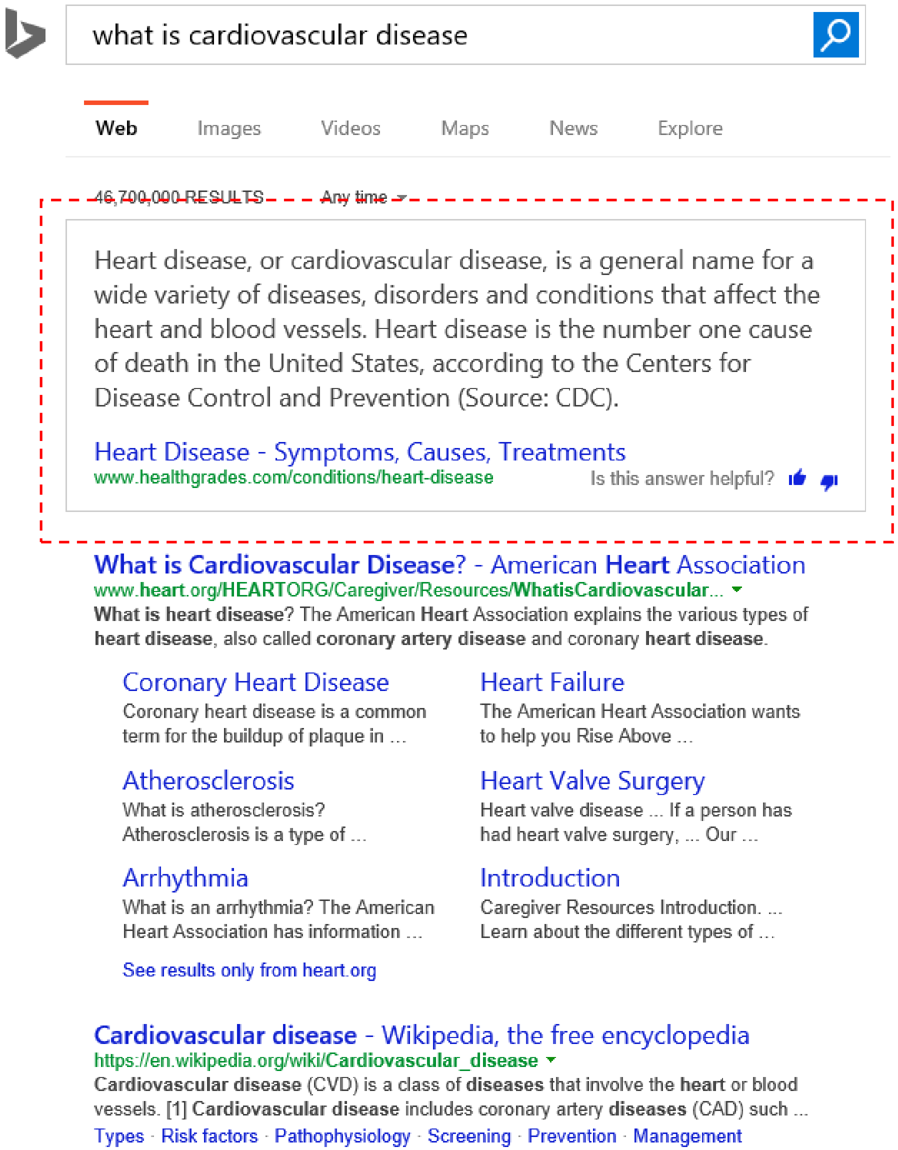
- This sentence has the form of definition
- Embedding is helpful to some extent, but it also return high similarity score for (emphysema, disease) and (emphysema, smoking)

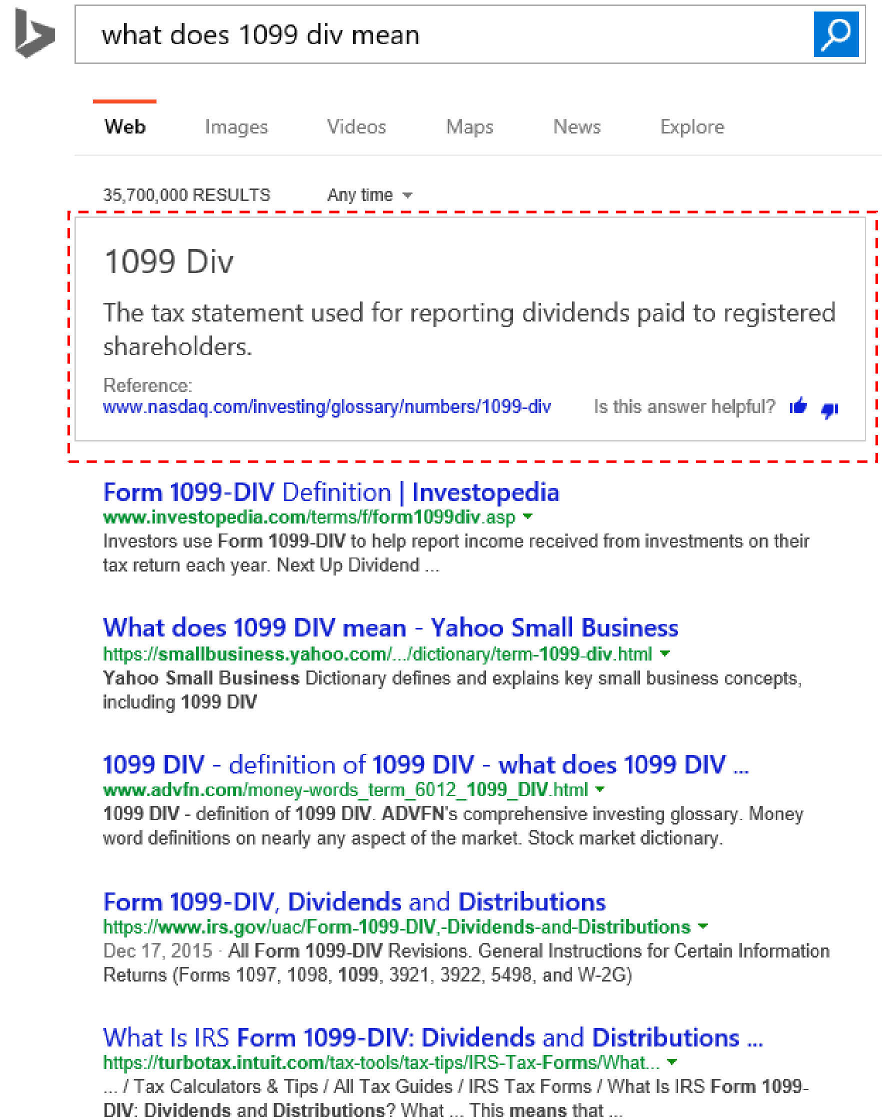
Answer 2:

Emphysema is an incurable, progressive lung disease that primarily affects smokers and causes shortness of breath and difficulty breathing.

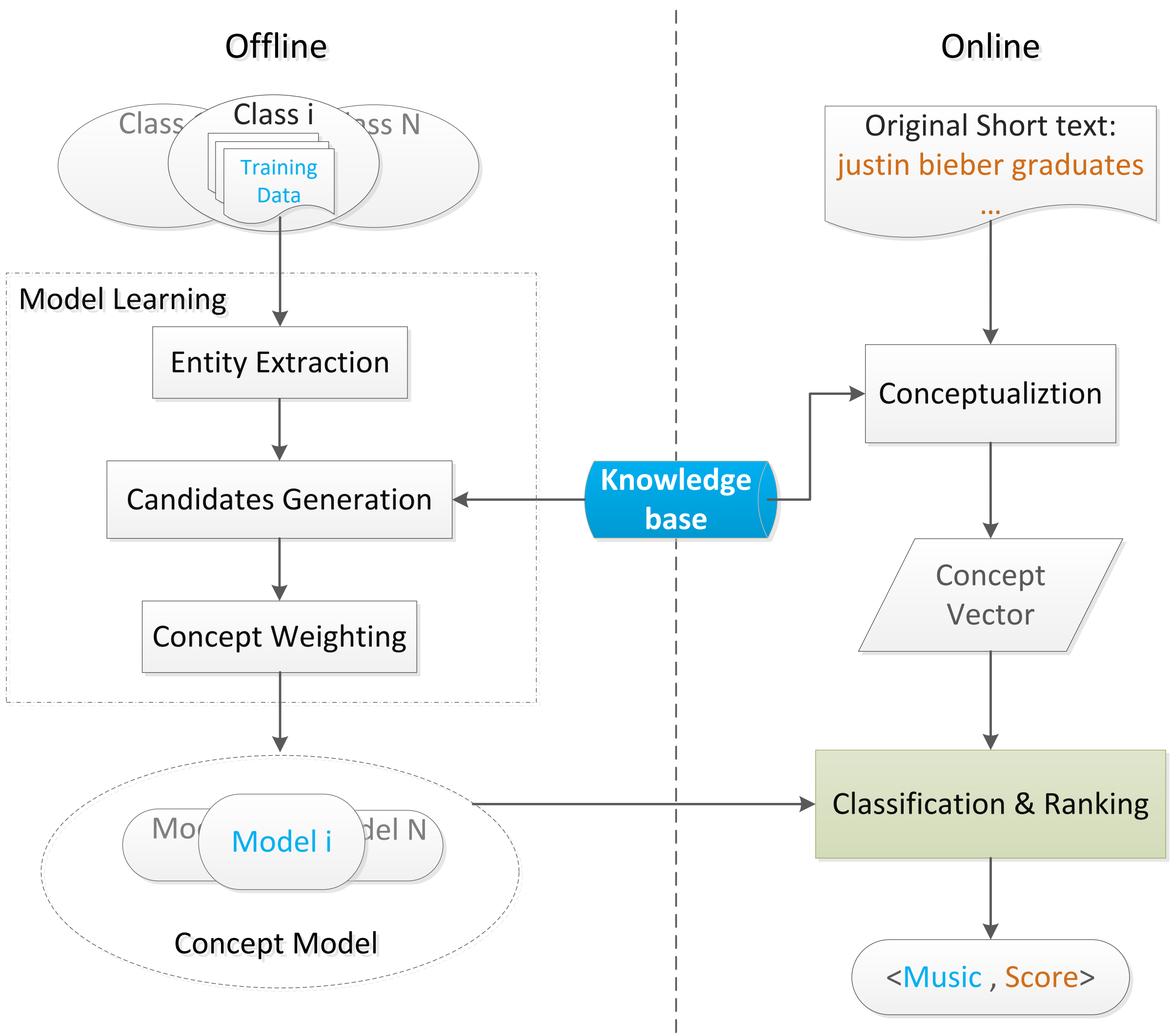
- Conceptualization can provide strong semantics
- Contextual embedding can also provide semantic similarity beyond Is-A

Definition Mining [Hao et al. 2016]

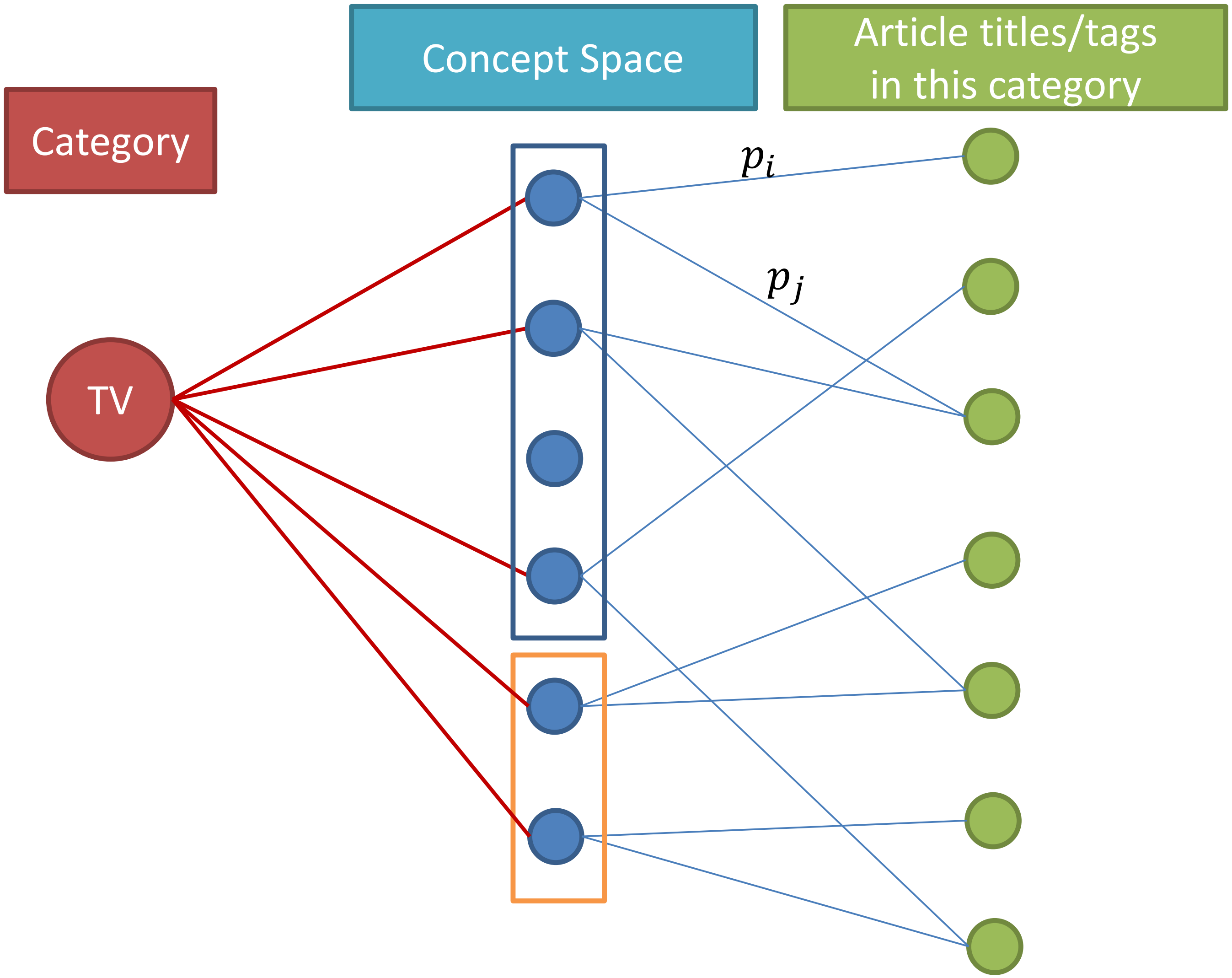




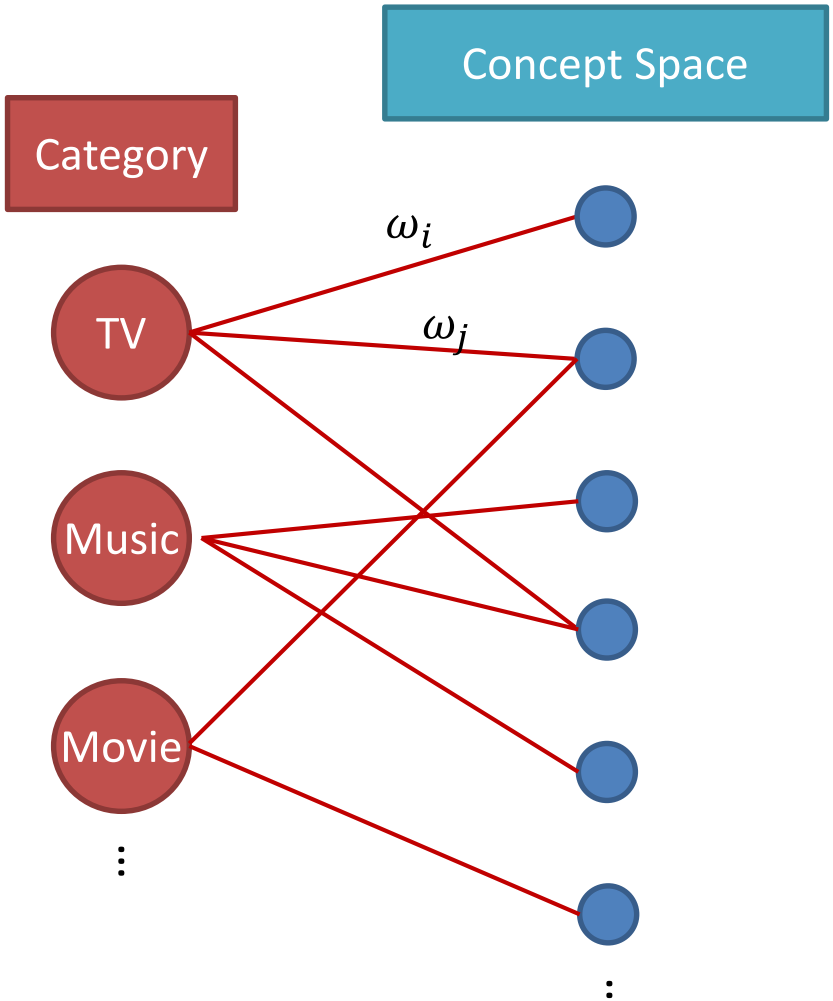
Concept based Short Text Classification and Ranking [Wang et al. 2014a]



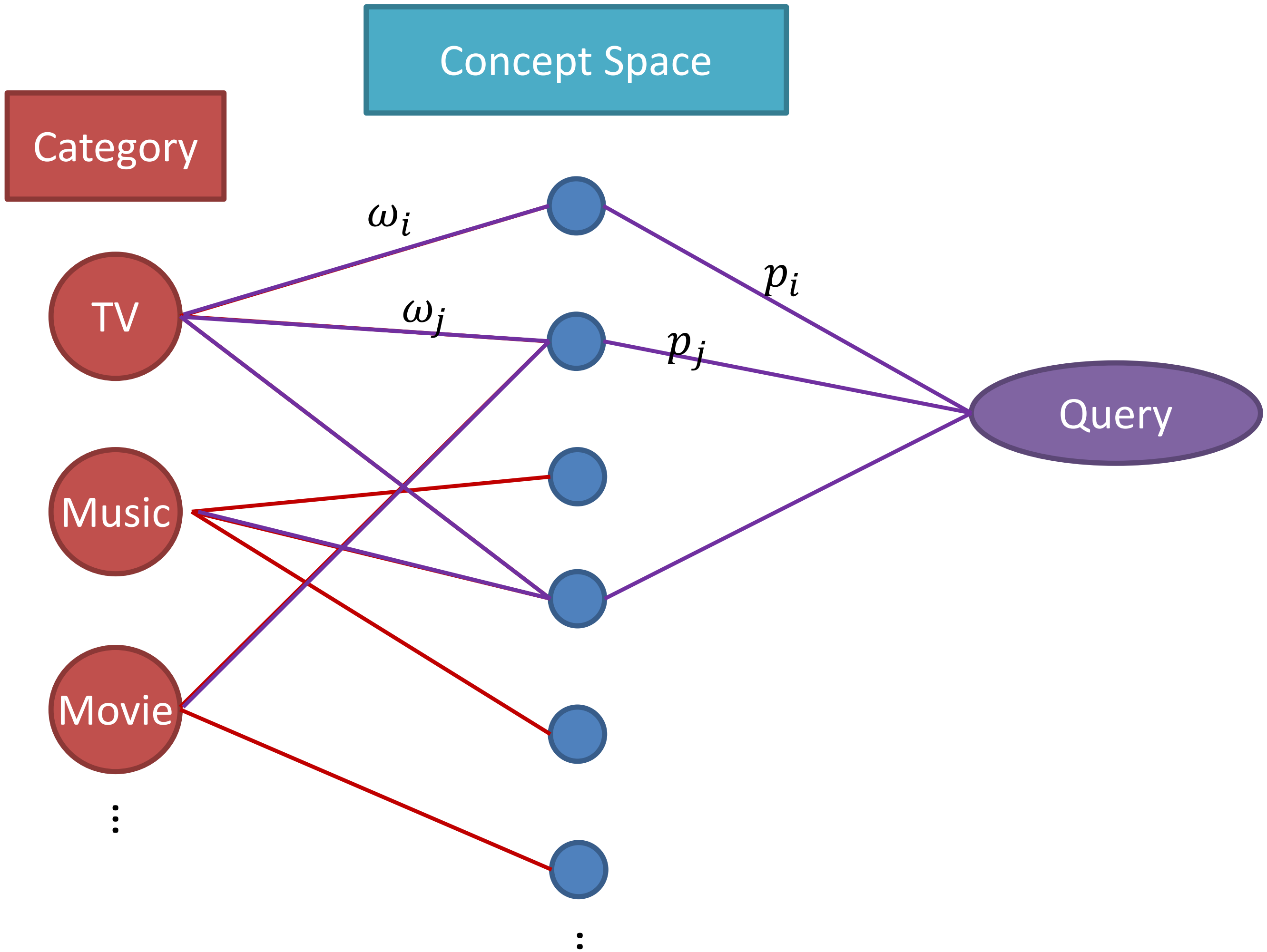
Concept based Short Text Classification and Ranking [Wang et al. 2014a]



Concept based Short Text Classification and Ranking [Wang et al. 2014a]

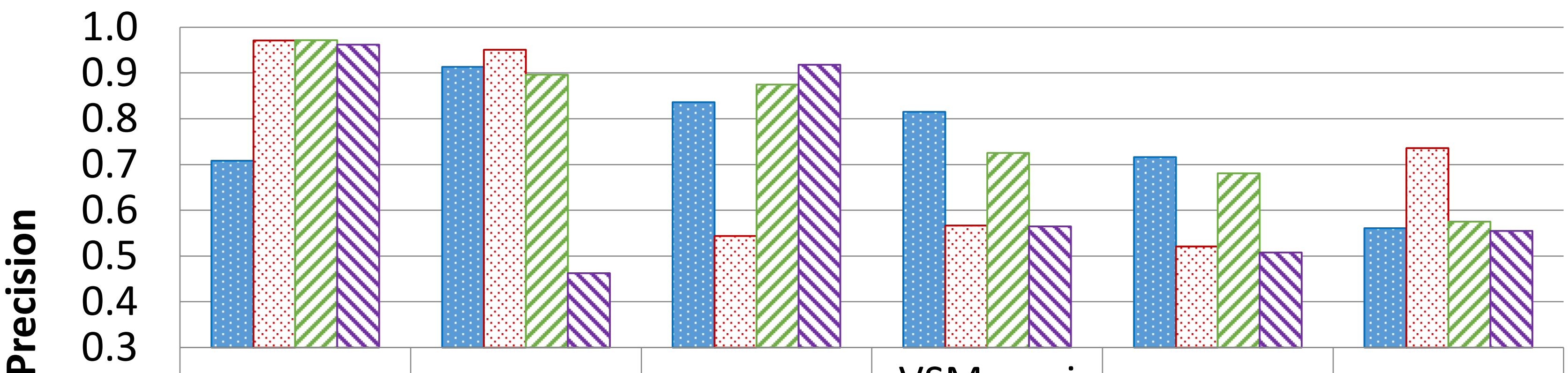


Concept based Short Text Classification and Ranking [Wang et al. 2014a]



Precision performance on each category

[Wang et al. 2014a]



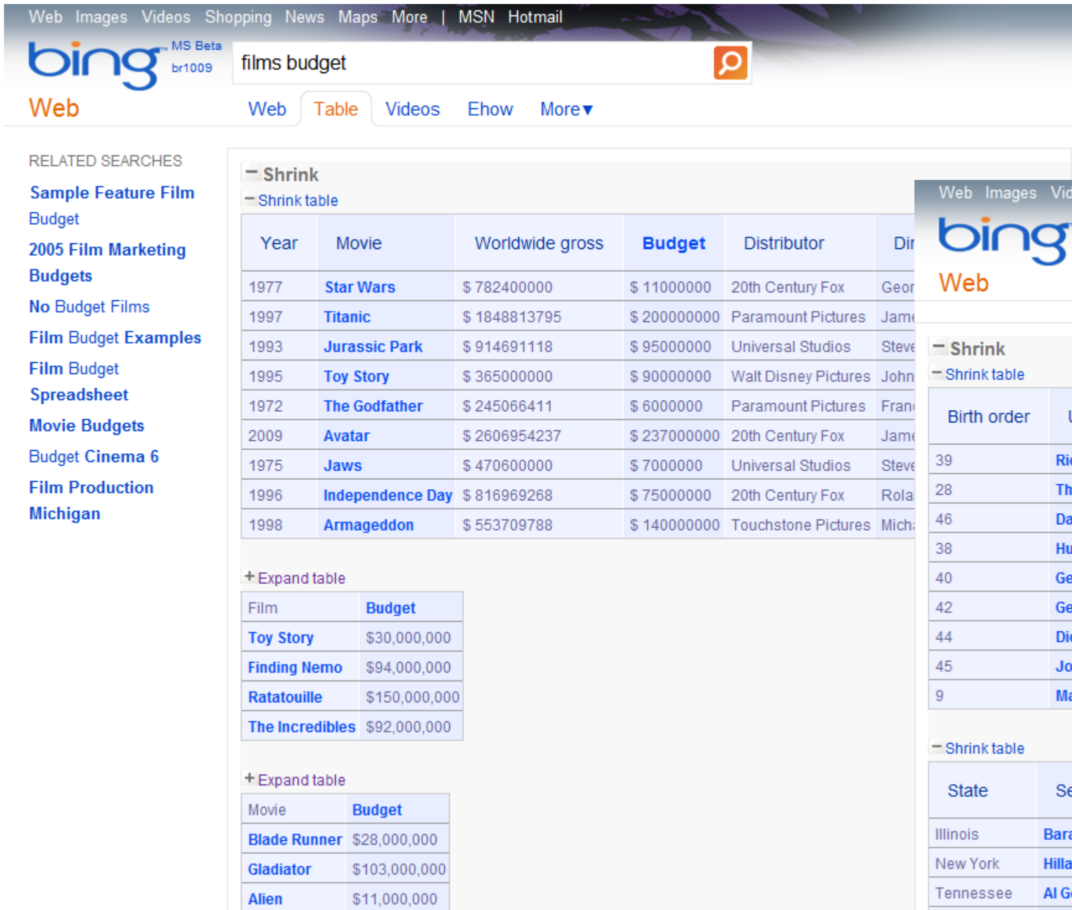
	BocSTC	LM_ch	SVM	VSM_cosine	LM_d	Entity_ESA
Movie	0.71	0.91	0.84	0.81	0.72	0.56
Money	0.97	0.95	0.54	0.57	0.52	0.74
Music	0.97	0.90	0.88	0.73	0.68	0.58
TV	0.96	0.46	0.92	0.56	0.51	0.55

Examples [Wang et al. 2014a]

Table 7: Top 5 recommendations for each channel

Channel	BocSTC	LM_{ch}	SVM
Movie	tom hanks rita wilson bosom buddies lindsay lohan as elizabeth taylor christian bale gets choked up emma watson 2012 george clooney dated lucy liu	2012 mtv movie awards red carpet depp movie awards film composer morricone prometheus review watch men in black 3	dark night rises jonny depp prometheus review titanic movie avengers trailer
Music	elton john worries lady gaga the temptations madonna nazi image sara lownds dylan and bob dylan faith hill tim mcgraw married for 15 years	donna summer funeral music youtube chris brown music mtv music videos beyonce music	carly simon donna summer funeral elton john worries lady gaga paul simon elton john lady gaga health
TV	today show recipes teen mom sentenced news channel 8 mtv music videos master chef 2012	yolanda adams morning show yard crashers world trade center workaholics season 3 words with friends cheats	secret life of the american teenager american horror story season 2 canceled shows 2012 make it or break it master chef 2012
Money	comercia web banking neighbors credit union tn unemployment morgan stanley clientserv chase credit cards	national grid online bill pay stock market plunge union first market bank small business marketing bad company	ally financial gmac astoria federal savings central bank chargeback it commerce bank online

Table Understanding [Wang et al. 2012a]



Year	Movie	Worldwide gross	Budget	Distributor	Dir
1977	Star Wars	$ 782400000	$ 11000000	20th Century Fox	Geor
1997	Titanic	$ 1848813795	$ 200000000	Paramount Pictures	Jame
1993	Jurassic Park	$ 914691118	$ 95000000	Universal Studios	Steve
1995	Toy Story	$ 365000000	$ 90000000	Walt Disney Pictures	John
1972	The Godfather	$ 245066411	$ 6000000	Paramount Pictures	Fran
2009	Avatar	$ 2606954237	$ 237000000	20th Century Fox	Jame
1975	Jaws	$ 470600000	$ 7000000	Universal Studios	Steve
1996	Independence Day	$ 816969268	$ 75000000	20th Century Fox	Rola
1998	Armageddon	$ 553709788	$ 140000000	Touchstone Pictures	Mich

+Expand table

Film	Budget
Toy Story	$30,000,000
Finding Nemo	$94,000,000
Ratatouille	$150,000,000
The Incredibles	$92,000,000

+Expand table

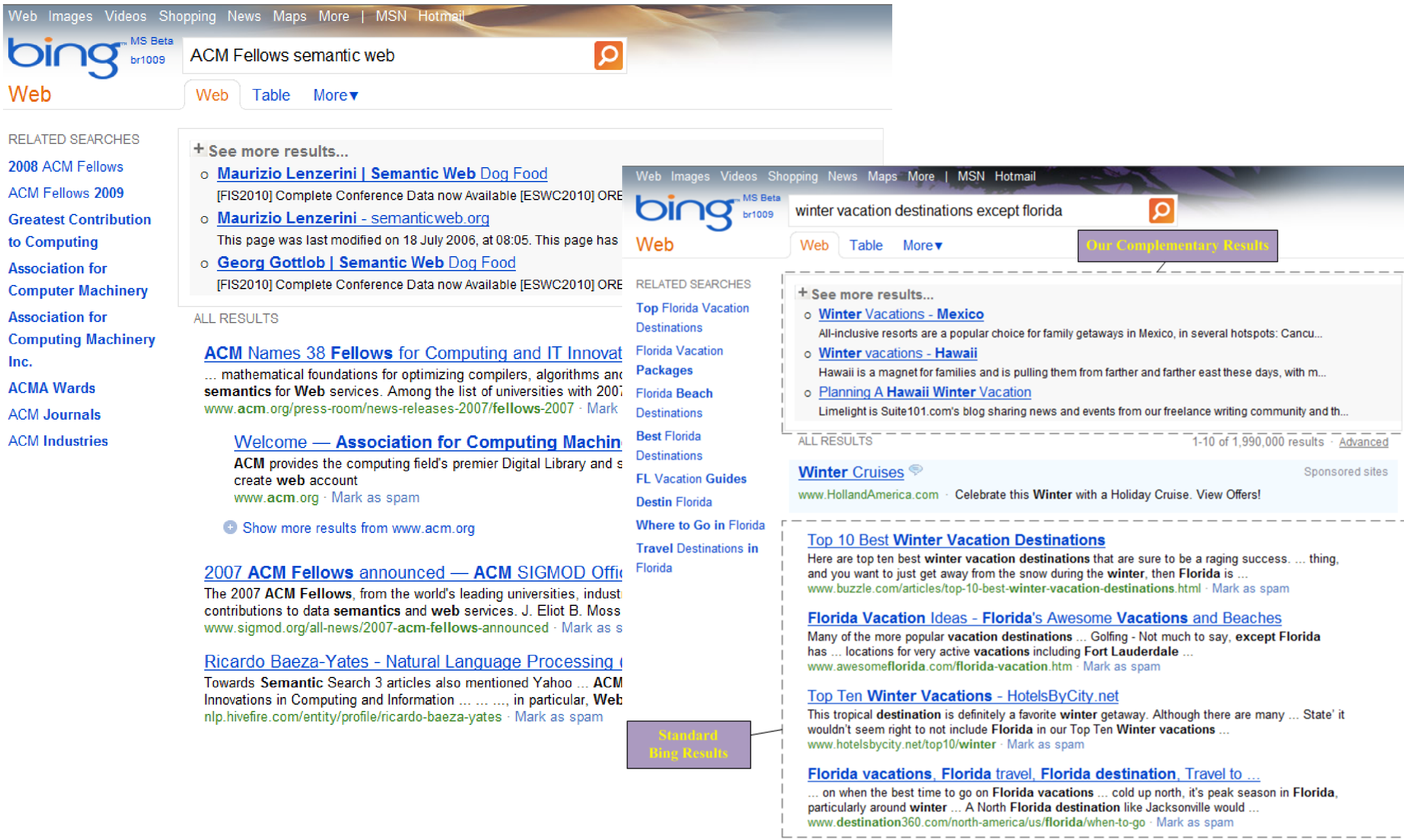
Movie	Budget
Blade Runner	$28,000,000
Gladiator	$103,000,000
Alien	$11,000,000

Birth order	U.S. Vice President	Birthdate	Century	Order of office	Birthplace
39	Richard Nixon	January 9, 1913	20th	36	Yorba Linda , California
28	Theodore Roosevelt	October 27, 1858	19th	25	New York City , New York
46	Dan Quayle	February 4, 1947	20th	44	Indianapolis , Indiana
38	Hubert Humphrey	May 27, 1911	20th	38	Wallace , South Dakota
40	Gerald Ford	July 14, 1913	20th	40	Omaha , Nebraska
42	George H. W. Bush	June 12, 1924	20th	43	Milton , Massachusetts
44	Dick Cheney	January 30, 1941	20th	46	Lincoln , Nebraska
45	Joseph Biden	November 20, 1942	20th	47	Scranton , Pennsylvania
9	Martin Van Buren	December 5, 1782	18th	8	Kinderhook , New York

−Shrink table

State	Senator	Party	Date of birth	Term	Age (Years/Days)
Illinois	Barack Obama	Democratic	August 4, 1961	2005 - 2008	1961 8 4
New York	Hillary Clinton	Democratic	October 26, 1947	2001 - 2009	1947 10 26
Tennessee	Al Gore	Democratic	March 31, 1948	1985 - 1993	1948 3 31
North Carolina	John Edwards	Democratic	June 10, 1953	1999 - 2005	1953 6 10
Kansas	Bob Dole	Republican	July 22, 1923	1969 - 1996	1923 7 22
Indiana	Dan Quayle	Republican	February 4, 1947	1981 - 1989	1947 2 4
Delaware	Joe Biden	Democratic	November 20, 1942	1973 - 2009	1942 11 20
Colorado	Gary Hart	Democratic	November 28, 1936	1975 - 1987	1936 11 28
Nebraska	Chuck Hagel	Republican	October 4, 1946	1997 - 2009	1946 10 6

Semantic Search [Wang et al. 2012b]



References

- [Stark et al. 1998] Michael M. Stark, and Richard F. Riesenfeld, WordNet: An Electronic Lexical Database. Proceedings of 11th Eurographics Workshop on Rendering. 1998.
- [Banko et al. 2007] Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead and Oren Etzioni, Open Information Extraction from the Web in IJCAI 2007.
- [Etzioni et al. 2011] Etzioni, Oren, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. "Open Information Extraction: The Second Generation." In IJCAI, vol. 11, pp. 3-10. 2011.
- [Carlson et al. 2010] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka Jr. and T.M. Mitchell, Toward an Architecture for Never-Ending Language Learning. In Proceedings of the Conference on Artificial Intelligence (AAAI), 2010.
- [Wu et al. 2012]   Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Zhu, Probase: A Probabilistic Taxonomy for Text Understanding. in ACM International Conference on Management of Data (SIGMOD), May 2012.
- [Bollacker et al. 2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, Jamine Taylor, Freebase: a collaboratively created graph database for structuring human knowledge</i> in SIGMOD 2008.
- [Auer et al. 2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, Zachary G. Ives, DBpedia: A Nucleus for a Web of Open Data. In ISWC/ASWC 2007.

References

- [Suchanek et al. 2007] Fabian M. Suchanek, Gjergji Kasneci, Gerhard Weikum, Yago: a core of semantic knowledge in WWW 2007.
- [Wu et al. 2015] Sen Wu, Ce Zhang, Christopher De Sa, Jaeho Shin, Feiran Wang, and C. Ré, Incremental Knowledge Base Construction Using DeepDive, in VLDB 2015
- [Navigli et al. 2012] R. Navigli and S. Ponzetto. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. in Artificial Intelligence, 2012.
- [Nastase et al. 2010] Vivi Nastase, Michael Strube, Benjamin Börschinger, Cäcilia Zirn, and Anas Elghafari, WikiNet: A very large scale multi-lingual concept network in LREC, 2010.
- [Speer et al. 2013] Robert Speer, and Havasi Catherine, ConceptNet 5: A large semantic network for relational knowledge, The People's Web Meets NLP. Springer Berlin Heidelberg, 2013.
- [Hua et al. 2016] Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng and Xiaofang Zhou. "Understand Short Texts by Harvesting and Analyzing Semantic Knowledge". IEEE Transactions on Knowledge and Data Engineering (TKDE), 2016.
- [Hua et al. 2015] Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng, and Xiaofang Zhou, Short Text Understanding Through Lexical-Semantic Analysis. in International Conference on Data Engineering (ICDE), April 2015.

References

- [Li et al. 2013] Peipei Li, Haixun Wang, Kenny Q. Zhu, Zhongyuan Wang, and Xindong Wu. Computing term similarity by large probabilistic isa knowledge. In ACM International Conference on Information and Knowledge Management (CIKM), 2013.
- [Li et al. 2015] Peipei Li, Haixun Wang, Kenny Q. Zhu, Zhongyuan Wang, Xue-Gang Hu, and Xindong Wu: A Large Probabilistic Semantic Network based Approach to Compute Term Similarity. In IEEE Transactions on Knowledge and Data Engineering (TKDE), 27(10): 2604-2617, 2015.
- [Rosch et al. 1976] Eleanor Rosch, Carolyn B Mervis, Wayne D Gray, David M Johnson, and Penny BoyesBraem. Basic objects in natural categories. Cognitive psychology, 8(3):382–439, 1976.
- [Manning and Schutze 1999] Christopher D Manning and Hinrich Schutze. Foundations of statistical natural language processing. In volume 999. MIT Press, 1999.
- [Wang et al. 2015b] Zhongyuan Wang, Kejun Zhao, Haixun Wang, Xiaofeng Meng, and Ji-Rong Wen, Query Understanding through Knowledge-Based Conceptualization In IJCAI, July 2015.
- [Bergsma et al. 2007]Shane Bergsma, Qin Iris Wang. Learning Noun Phrase Query Segmentation. In EMNLP-CoNLL 2007: 819-826.
- [Tan et al. 2008] Bin Tan, Fuchun Peng. Unsupervised query segmentation using generative language models and wikipedia. In WWW 2008: 347-356.

References

- [Li et al. 2011] Yanen Li, Bo-June Paul Hsu, ChengXiang Zhai, Kuansan Wang. Unsupervised query segmentation using clickthrough for information retrieval. In SIGIR 2011: 285-294.
- [Guo et al. 2009] Jiafeng Guo, Gu Xu, Xueqi Cheng, Hang Li. Named entity recognition in query. In SIGIR 2009: 267-274.
- [Pantel et al. 2012] Patrick Pantel, Thomas Lin, Michael Gamon. Mining Entity Types from Query Logs via User Intent Modeling. In ACL 2012: 563-571.
- [Joshi et al. 2014] Mandar Joshi, Uma Sawant, Soumen Chakrabarti. Knowledge Graph and Corpus Driven Segmentation and Answer Inference for Telegraphic Entity-seeking Queries. In EMNLP 2014: 1104-1114.
- [Sawant et al. 2013] Uma Sawant, Soumen Chakrabarti. Learning joint query interpretation and response ranking. In WWW 2013: 1099-1110.
- [Wang et al. 2014b] Zhongyuan Wang, Haixun Wang, and Zhirui Hu, Head, Modifier, and Constraint Detection in Short Texts in International Conference on Data Engineering (ICDE), 2014.
- [Sun et al. 2016] Xiangyan Sun, Haixun Wang, Yanghua Xiao, Zhongyuan Wang, Syntactic Parsing of Web Queries In EMNLP 2016.

References

- [Kenter and Rijke 2015] Tom Kenter and Maarten de Rijke. Short text similarity with word embeddings In CIKM, 2015.
- [Wang et al. 2015a] Zhongyuan Wang, Haixun Wang, Ji-Rong Wen, and Yanghua Xiao, An Inference Approach to Basic Level of Categorization In CIKM, October 2015.
- [Hao et al. 2016] Zehui Hao, Zhongyuan Wang, Xiaofeng Meng and Jun Yan, Combining Language Model with Conceptualization for Definition Ranking. MSR-Technical Report, 2016.
- [Wang et al. 2014a] Fang Wang, Zhongyuan Wang, Zhoujun Li, and Ji-Rong Wen, Concept-based Short Text Classification and Ranking In CIKM, 2014.
- [Wang et al. 2012a] Jingjing Wang, Haixun Wang, Zhongyuan Wang, and Kenny Zhu, Understanding Tables on the Web. In International Conference on Conceptual Modeling, October 2012.
- [Wang et al. 2012b] Yue Wang, Hongsong Li, Haixun Wang, and Kenny Zhu,Toward Topic Search on the Web. In International Conference on Conceptual Modeling, October 2012.

Part III. Implicit Representation for Short Text Understanding

Zhongyuan Wang (Microsoft Research)

Haixun Wang (Facebook Inc.)

Tutorial Website:
http://www.wangzhongyuan.com/tutorial/ACL2016/Understanding-Short-Texts/

“Implicit” model

- Goal:
  - A distributed representation of a short text that captures its semantics.
- Why?
  - To solve the sparsity problem
  - Representation readily used as features in downstream models

Short Text vs. **Phrase** Embedding

- There's a lot of work on embedding phrases.
- A short text (e.g., a web query) is often not well formed
  - e.g., no word order, no functional words
- A short text (e.g., a web query) is often more expressive
  - e.g., "distance earth moon"

Applications

GOOGLE TECH

Google is using an AI called 'RankBrain' to answer ambiguous questions

By Colin Lecher on October 26, 2015 09:54 am @colinlecher

http://www.theverge.com/2015/10/26/9614836/google-search-ai-rankbrain

THIRD MOST IMPORTANT SEARCH SIGNAL

RankBrain

- A huge vocabulary
  - Contains every possible token
- Query, doc title, doc URL representation
  - Average word embedding
- Architecture:
  - 3 – 4 hidden layers
- Data
  - Months of search log data

The Core Problem (for the rest of us)

- What is the objective function used in training the representation?
- Does the optimal solution force the representation to capture the full semantics?

Traditional Representation of Text

- Bag-of-Words (BOW) model: Text (such as a sentence or a document) is represented as a bag (multiset) of words, disregarding grammar and word order but keeping multiplicity.

  1. John likes to watch movie, Mary likes movie too.
  2. John also likes to watch football games.

  The sentences are represented by two 10-entry vectors;

  (1) [1,2,1,1,2,0,0,0,1,1]

  (2) [1,1,1,1,0,1,1,1,0,0]

- Disadvantages: No word order. Matrix is sparse.

Assumption: Distributional Hypothesis

- **Distributional Hypothesis**: Words that are used and occur in the same contexts tend to purport similar meaning (Wikipedia).

- E.g. *Paris is the capital of France*.

- In this assumption, “Paris” will be close in semantic space with “London” , which would also be surrounded by “capital of” and country’s name.

- Based on this assumption, researchers proposed many models to learn the text representations from corpus.

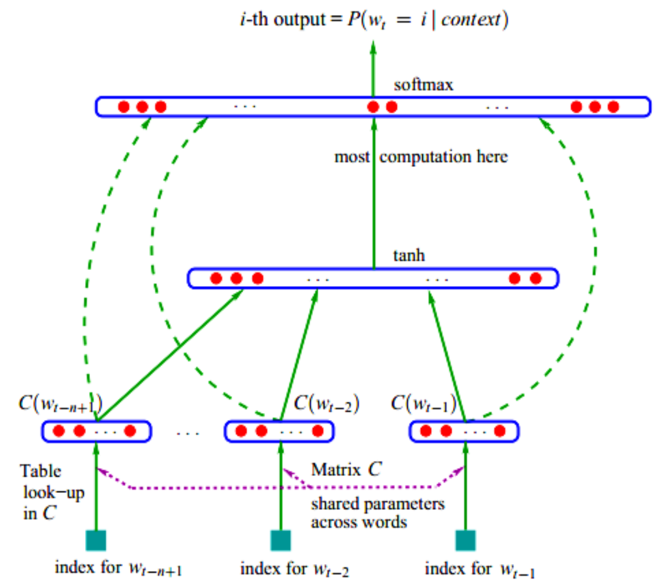
Neural Network Language Model (Bengio et al. 2003)

Statistical model

$$\hat{P}(w_1^T) = \prod_{t=1}^{T} \hat{P}(w_t | w_1^{t-1})$$

Assuming a word is determined by its **previous words**.

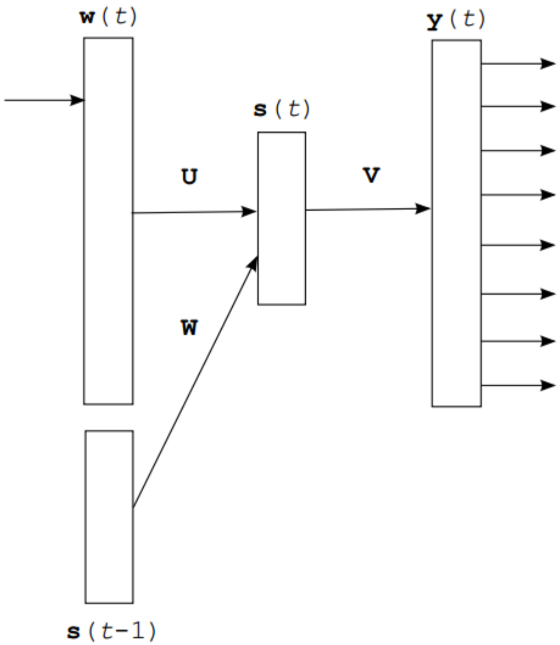
Two words with same previous words will share similar semantics.



Neural architecture: $f(i, w_{t-1}, \cdots, w_{t-n+1}) = g(i, C(w_{t-1}), \cdots, C(w_{t-n+1}))$ where g is the neural network and $C(i)$ is the i-th word feature vector.

Yoshua Bengio, Réjean Ducharme ,Pascal Vincent, Christian Jauvin "A Neural Probabilistic Language Model" Journal of Machine Learning Research 3 (2003) 1137–1155

Recurrent Neural Net Language Model (Mikolov, 2012)



Output Values:

$$s(t) = f(Uw(t) + Ws(t-1))$$

$$y(t) = g(Vs(t))$$

$w(t)$: *input word at time t*

y(t): *output probability distribution over words*

s(t): *hidden layer*

U,V,W: *transformation matrix*

- Generate much more meaningful text than n-gram models

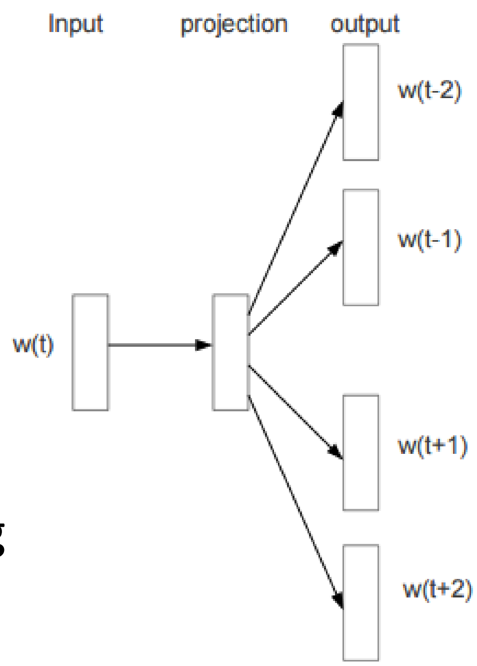
- The sparse history h is projected into some continuous low-dimensional space, where similar histories get clustered

Word2Vector Model (Mikolov et al. 2013)

- The word2vec projects words in a **shallow** layer structure.

maximize $$\sum_{(w,c)\in D} \sum_{w_j \in c} \log P(w|w_j)$$

- **Directly** learn the representation of words using context words
- Optimizing the objective function in whole corpus.

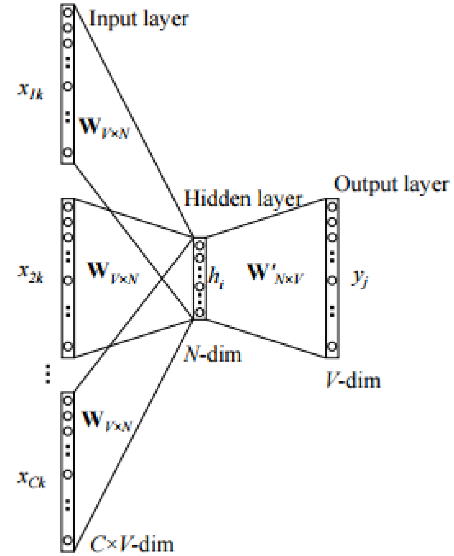


Efficient estimation of word representations in vector space. Mikolov et al 2013

Word2Vector Model (Mikolov et al. 2013)

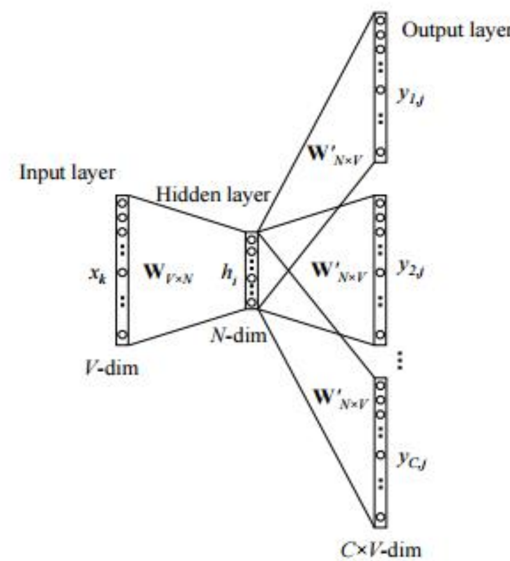
CBOW

- Given the **word**, predicting the **context**
- Faster to train than the skip-gram, better accuracy for the **frequent words**



Skip-gram

- Given the **context**, predicting the **word**
- Works well with small training data, represents well even **rare words** or **phrases**



GloVe: Global Vectors for Word Representation (Pennington et al. 2014)

- Constructing the word-word co-occurrence matrix of whole corpus.
- Inspired by LSA, using matrix factorization to produce word representation.

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

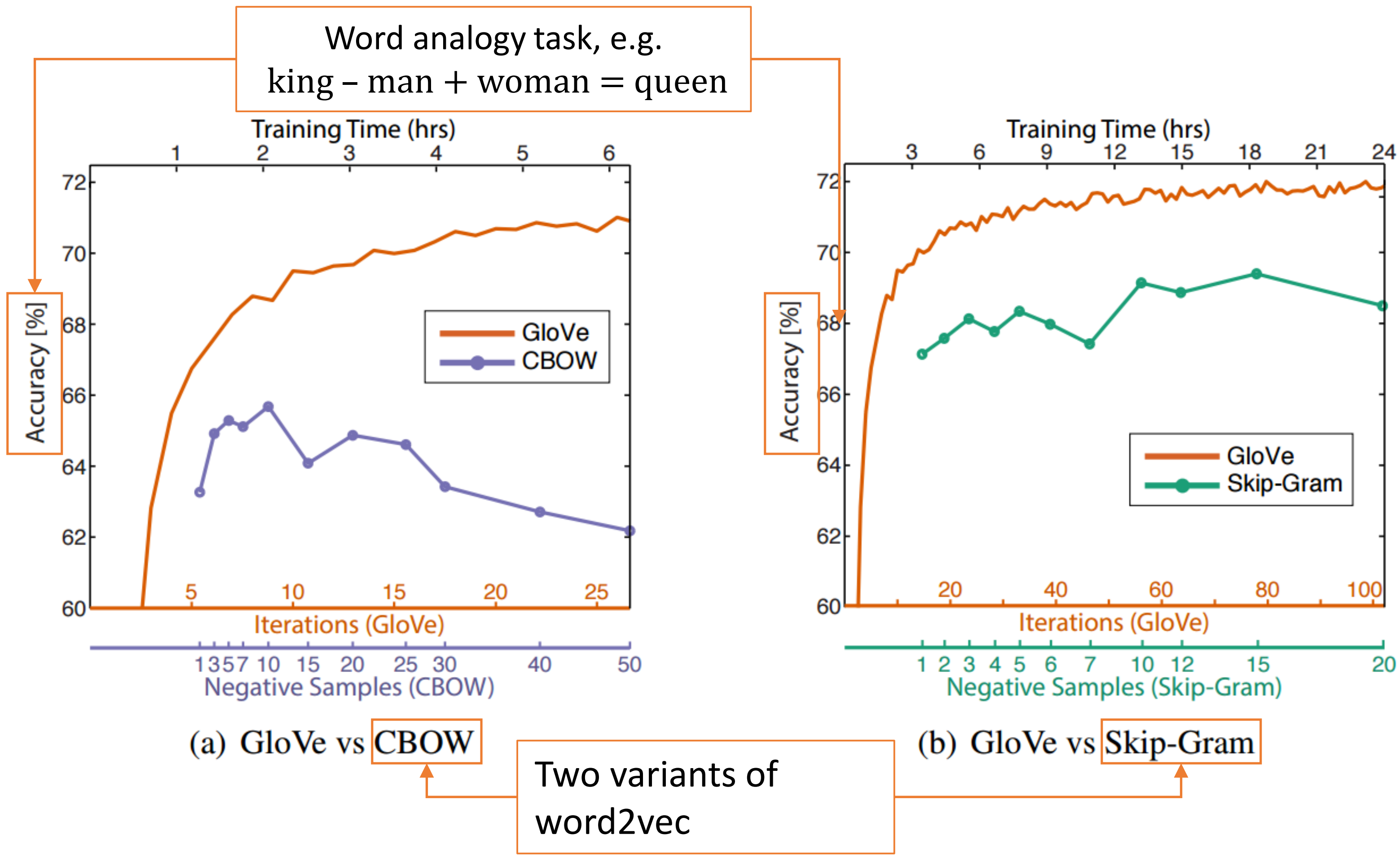
Loss function: $\hat{J} = \sum_{i,j} f(X_{ij}) \left(w_i^T \widetilde{w}_j - \log X_{ij}\right)^2$

X-ij is the count of if j-th word occurs, the occurrence of i-th word. ***w*** are word vectors. Minimize loss function.

Global Vectors for Word Representation, Pennington et al, 2014

GloVe: Global Vectors for Word Representation (Pennington et al. 2014)

- GloVe vs Word2Vec



Beyond words

Word embedding is a great success.

Phrase and sentence embedding is much harder:

- Sparsity: from atomic symbols to compositional structures
- Ground truth: from syntactic context to semantic similarity

Composition methods

- Algebraic composition

- Composition tied with syntax (dependency tree of phrase / sentences)

Averaging

- Expand vocabulary to include ngrams

- Otherwise go with bag of unigrams.

"A cat is being chased by a dog in yard"

$$\frac{v_1 + v_2 + \cdots + v_n}{n} = v_{sentence} \, ?$$

- But a "jade elephant" is not an "elephant"

Linear transformation

- $p = f(u, v)$, where

  u, v are embedding of uni-grams u, v

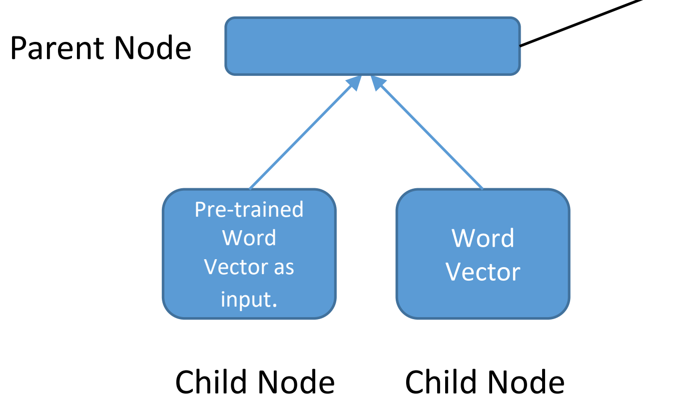
  f is a composition function

- Common composition model: **linear transformation**

- training data: unigram and bigram embeddings

Recursive Auto-encoder with Dynamic Pooling

- **Recursive Auto-encoder**
- **From bottom to top**, leaves to root.
- After parsing, **important** components in sentence will trend to get on **higher** level.



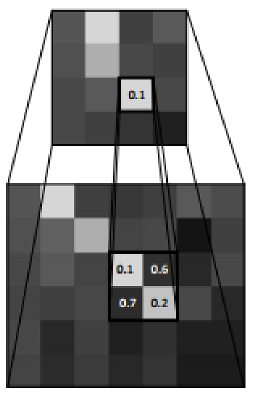
Non-linear activation function

$$p = f(W_e[c_1; c_2] + b)$$

$[c_1 : c_2]$ is the concatenation of two word vectors

Recursive Auto-encoder with Dynamic Pooling

- **Dynamic Pooling**



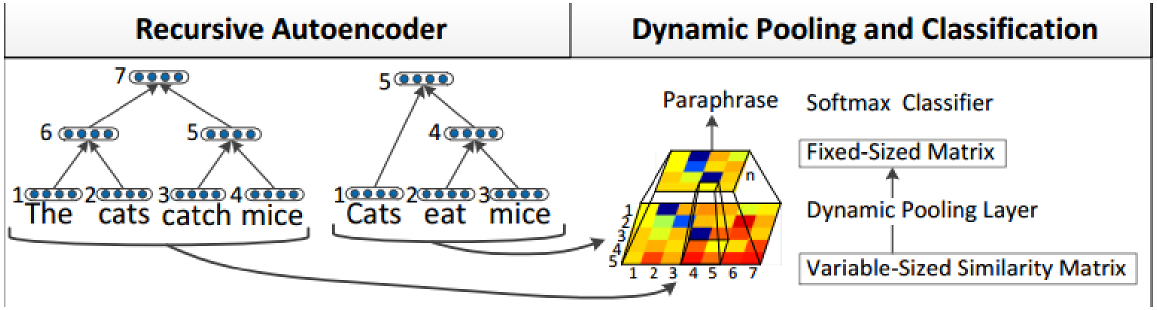
Example of the dynamic min-pooling layer finding the **smallest** number in a pooling window region of the original similarity matrix S.

- The sentences are not fixed-size. Using pooling to map them into fix-sized vector.

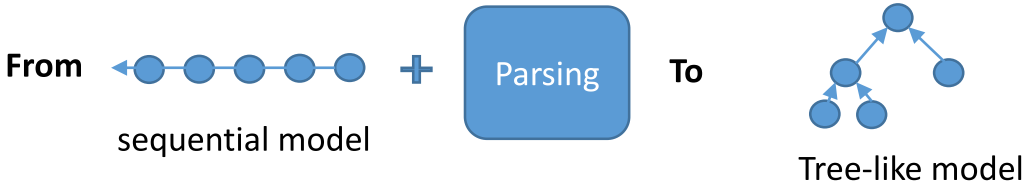
- Using fixed-size matrix as input of neural network or other classifiers.

Recursive Auto-encoder with Dynamic Pooling [Socher et al. 2011]

- Using **dependency parser** to transform sequence to tree structure, which retains **syntactical info**
- Using **dynamic pooling** to map varied-size sentence to a **fixed-size form**



Most time, the para2vec model or traditional RNN/LSTM doesn't consider the syntactical information of sentences.

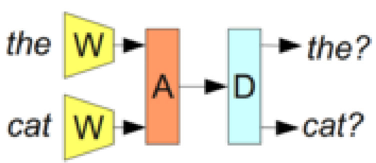


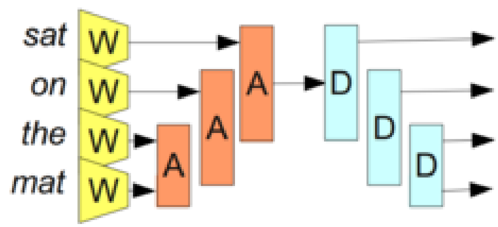
Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, Christopher D. Manning: "Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection." NIPS 2011: 801-809

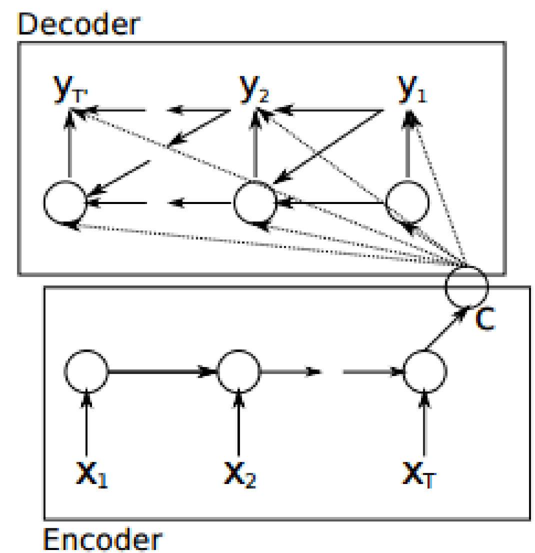
RNN encoder-decoder

(Cho et al. 2014)

- Create a **reversible** sentence representation.
- The representation can be reconstructed to an actual sentence form which is reasonable and novel.



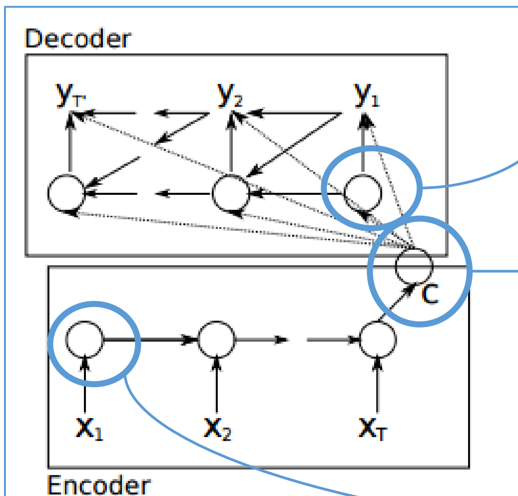




Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio: "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation." EMNLP 2014: 1724-1734

RNN encoder-decoder

(Cho et al. 2014)



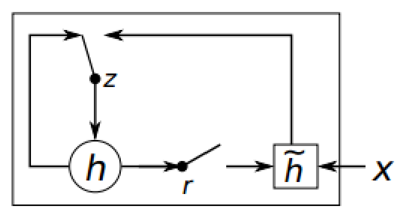
- The conditional distribution of next symbol.

$$P(y_t|y_{t-1}, y_{t-2}, \dots, y_1, c) = g(h_{<t>}, y_{t-1}, c)$$

- Add a summary(constant) symbol, it will hold the semantics of sentence.

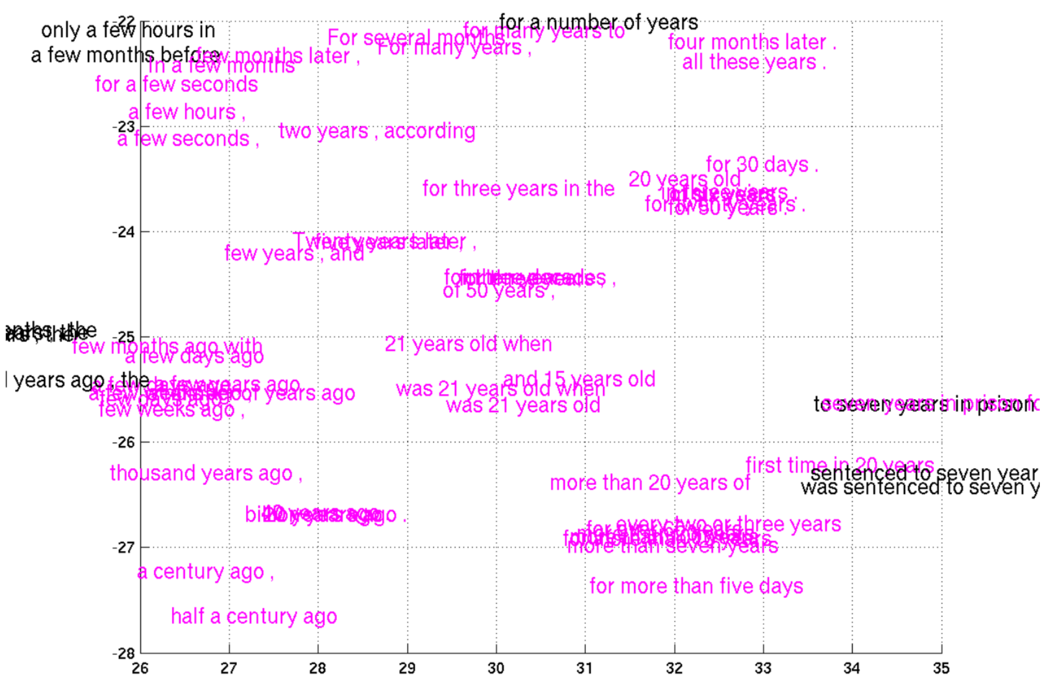
$$h_{<t>} = f(h_{<t-1>}, y_{t-1}, c)$$

- For long sentences, adding hidden unit to remember/forget memory.



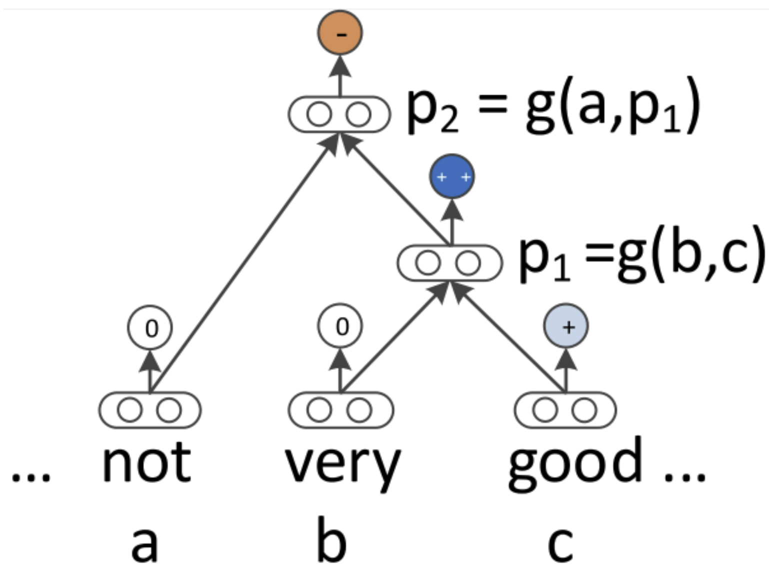
RNN encoder-decoder

(Cho et al. 2014)



Small section of the t-SNE of the phrase representation

RNN for composition [Socher et al 2011]



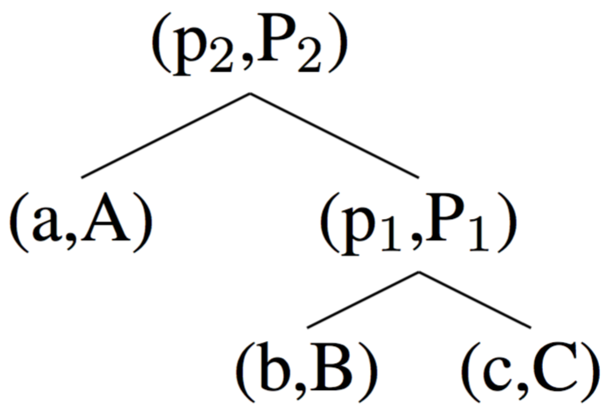
$$p_1 = f\left(W\begin{bmatrix} b \\ c \end{bmatrix}\right), p_2 = f\left(W\begin{bmatrix} a \\ p_1 \end{bmatrix}\right)$$

f = tanh is a standard element-wise nonlinearity

W is shared

MV-RNN [Socher et al. 2012]

- Each composition function depends on the actual words being combined.

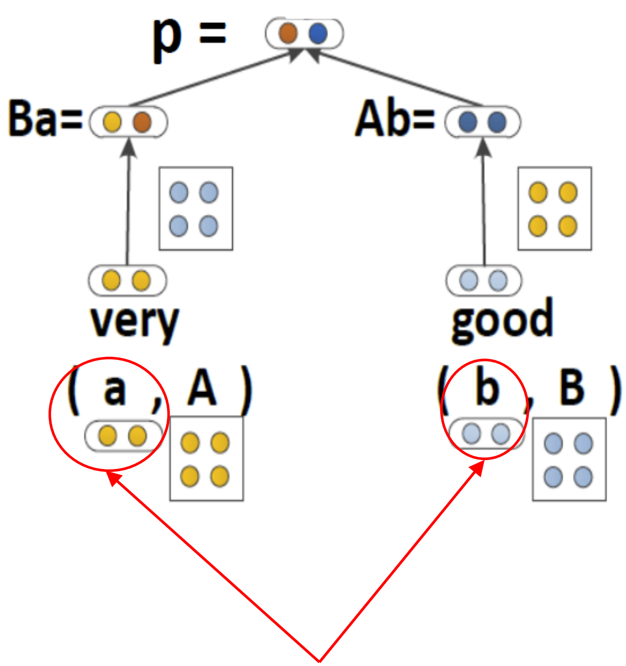


$$p_1 = f\left(W \begin{bmatrix} Cb \\ Bc \end{bmatrix}\right), P_1 = f\left(W_M \begin{bmatrix} B \\ C \end{bmatrix}\right)$$

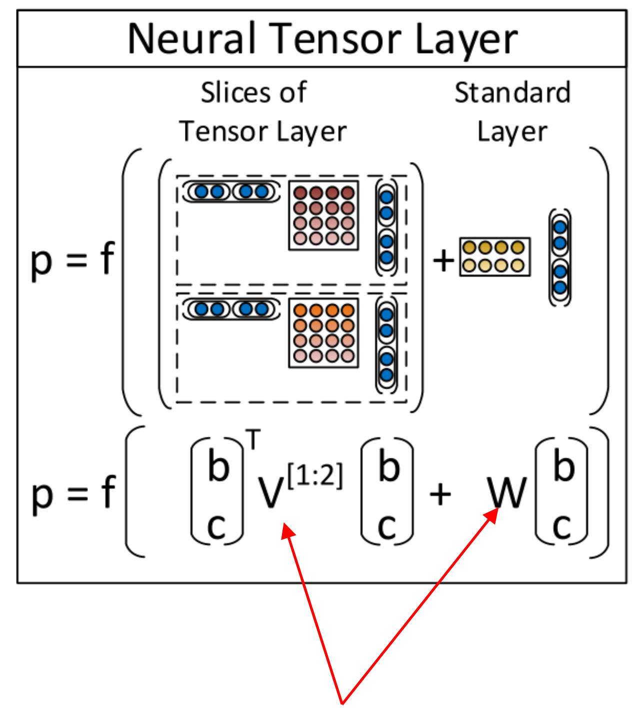
- Represent every word and phrase as both a vector and a matrix.

Recursive Neural Tensor Network [Socher et al. 2013]

- Number of parameters is very large for MV-RNN



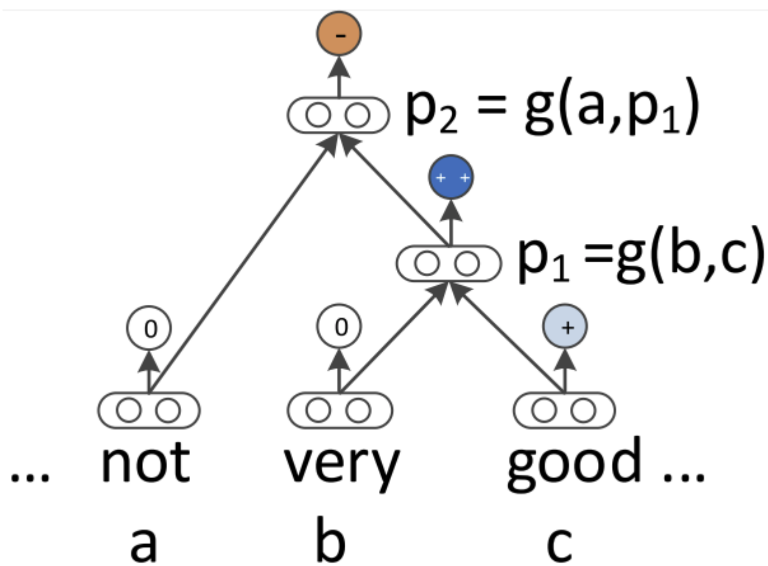
MV-RNN: need to train a new parameter for each leaf node



Use tensor: unified parameter for all nodes

Recursive Neural Tensor Network [Socher et al. 2013]

- Interpret each slice of the tensor as capturing a specific type of composition



$$p_1 = f\left(\begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} b \\ c \end{bmatrix} + W \begin{bmatrix} b \\ c \end{bmatrix}\right)$$

$$p_2 = f\left(\begin{bmatrix} a \\ p_1 \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} a \\ p_1 \end{bmatrix} + W \begin{bmatrix} a \\ p_1 \end{bmatrix}\right)$$

Assign label to each node via:

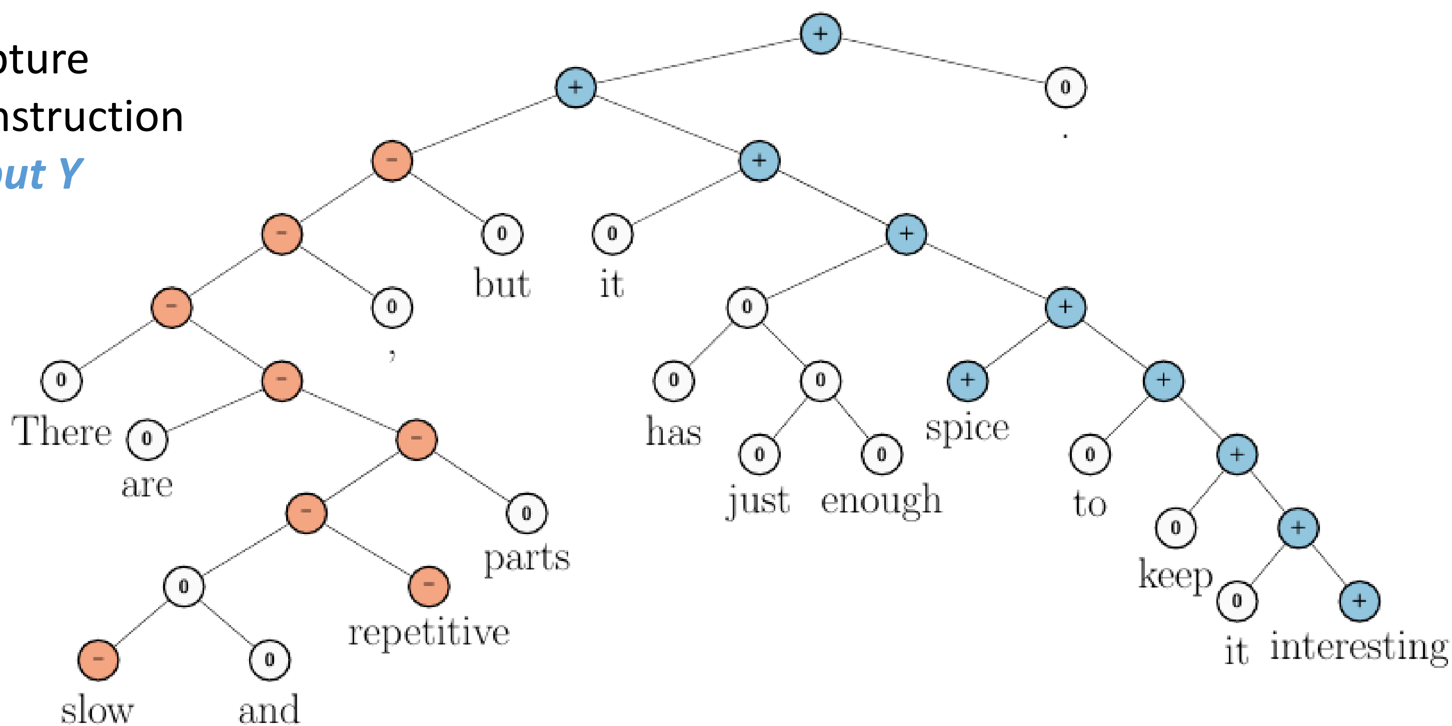
$$y^a = \mathrm{softmax}(W_s a)$$

Recursive Neural Tensor Network

- Target : sentiment analysis

Sentence: There are slow and repetitive parts,
but it has just enough spice to keep it interesting

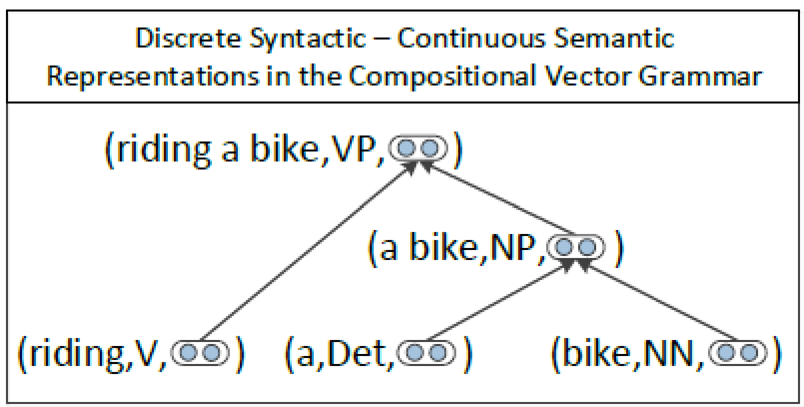
capture
construction
X but Y



Demo: http://nlp.stanford.edu:8080/sentiment/rntnDemo.html

CVG (Compositional Vector Grammars)
[Socher et al. 2013]

- Task: Represent phrase and categories
  - PCFG: capture discrete categorization of phrases
  - RNN: capture fine-grained syntactic and compositional-semantic information
- Parse and represent phrases as vector



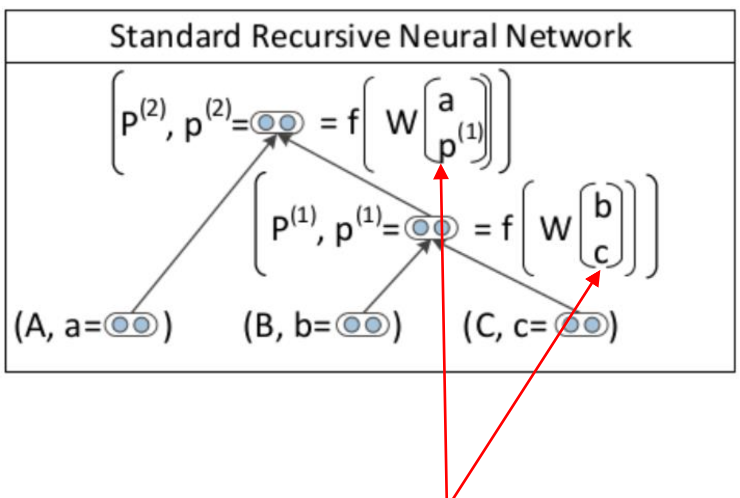
An example of CVG Tree

Parsing with Compositional Vector Grammars, Socher et al 2013

CVG

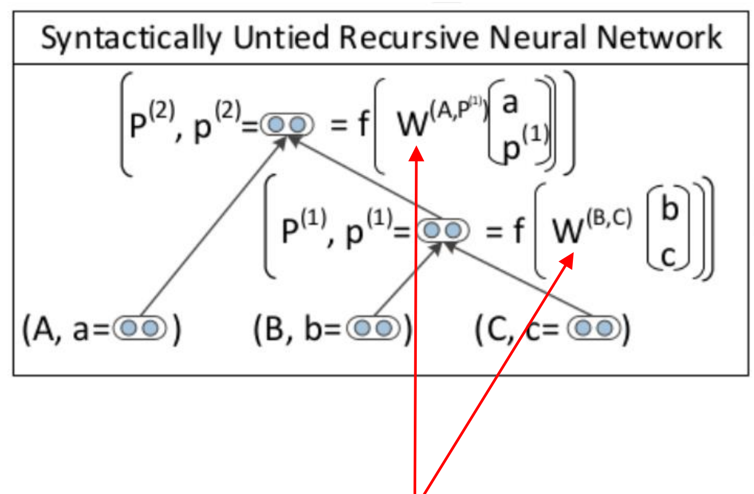
- Weights at each node are conditionally dependent on categories of the child constituents
- Combined with Syntactically Untied RNN

Normal RNN



Replicated weight matrix

SU-RNN

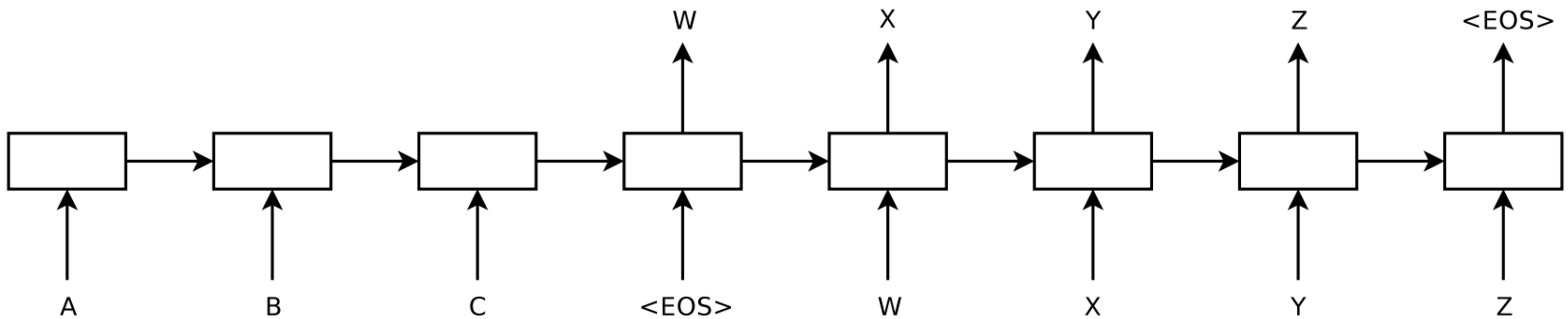


depends on syntactic categories of its children

Phrases & Sentences

- Composition based approaches
  - Algebraic composition not powerful enough
  - Syntactic composition requires parsing
- Non-composition based approaches
  - translation based approaches
  - extend word2vec to sentences, phrases
  - ground truth: search log, dictionary, image

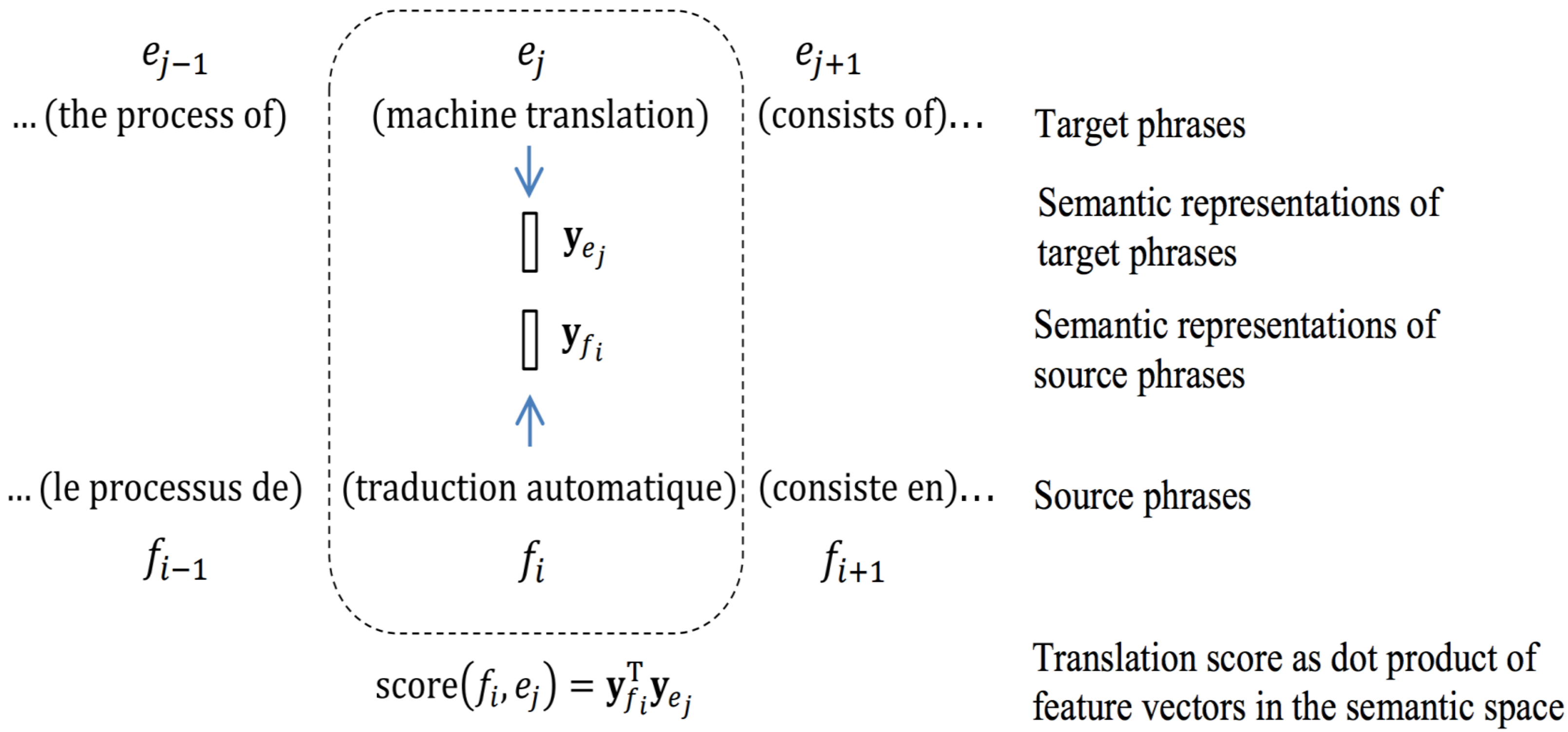
Sequence to sequence translation



- Last node "remembers" the semantics of the input sentence

- Not feasible for embedding web queries

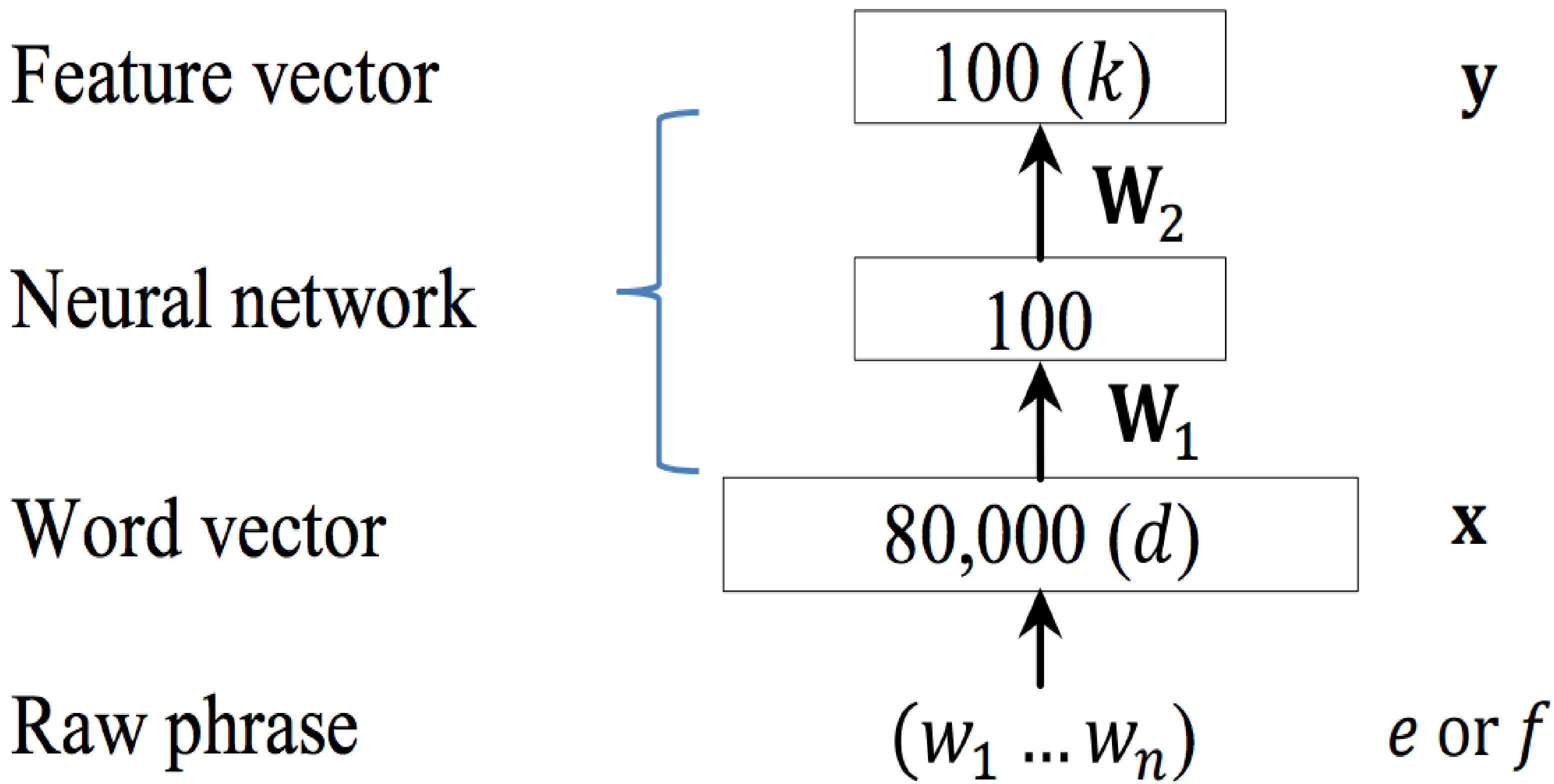
Phrase Translation Model [Gao et al 2013]



The quality of a phrase translation is judged implicitly through the translation quality (BLEU) of the sentences that contain the phrase pair.

Learning Semantic Representations for the Phrase Translation Model, Gao et al 2013

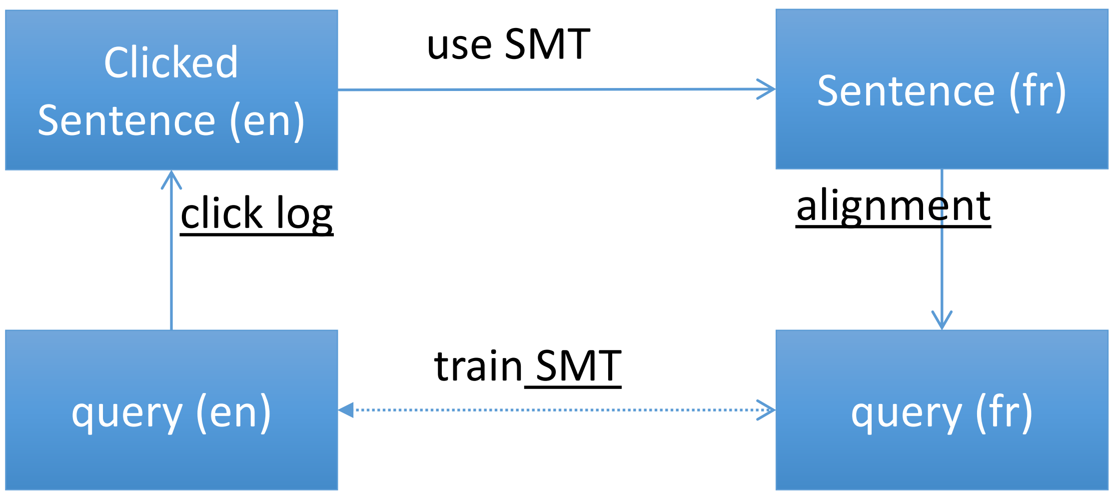
Phrase Translation Model



The core is the bag-of-words approach

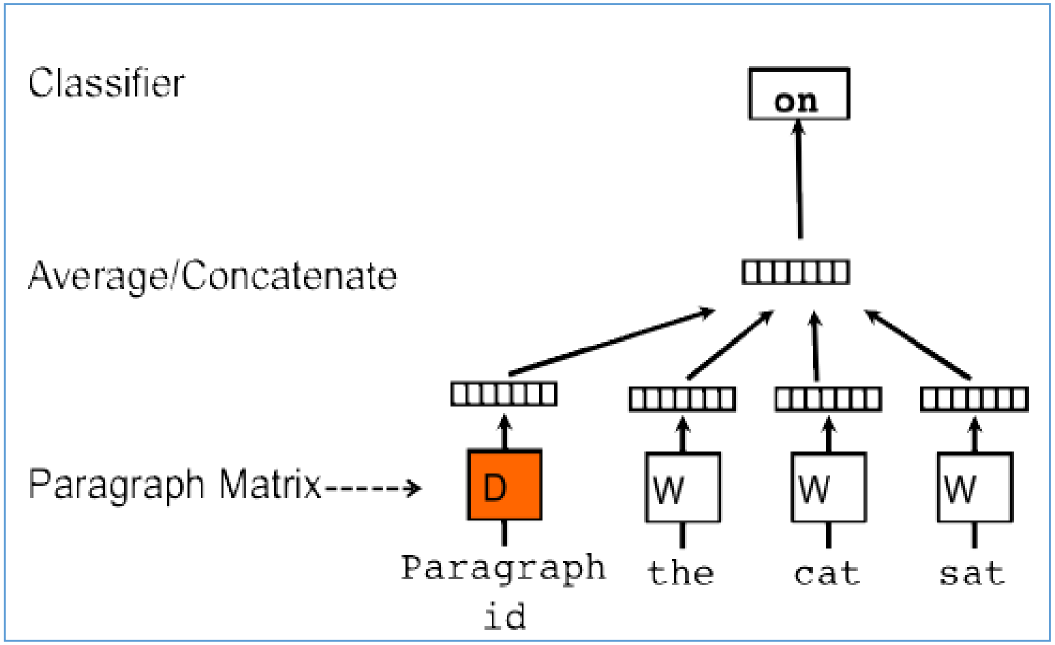
Web query translation model

- Training data



- Train an NN translation model on query(en) and query(fr) pair

Doc2Vec (Quoc Le et al 2014)



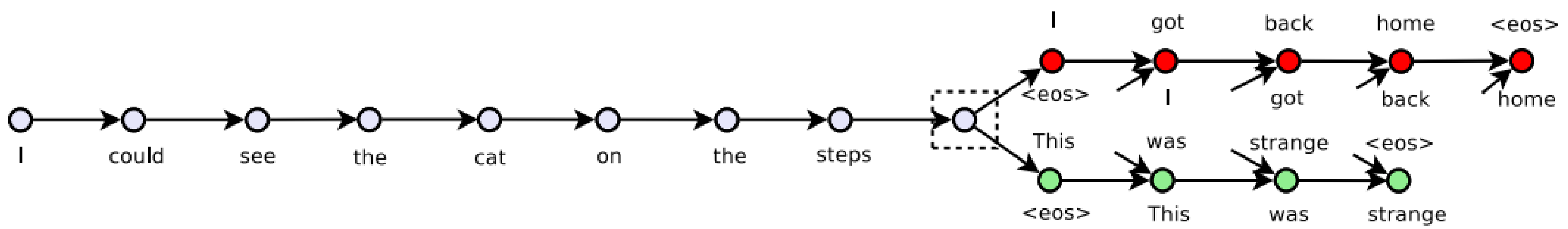
Distributed Representations of Sentences and Documents, Quoc Le et al. 2014

LDA vs. Doc2Vec

LDA	**Paragraph Vectors**
Artificial neural network	Artificial neural network
Predictive analytics	Types of artificial neural networks
Structured prediction	Unsupervised learning
Mathematical geophysics	Feature learning
Supervised learning	Predictive analytics
Constrained conditional model	Pattern recognition
Sensitivity analysis	Statistical classification
SXML	Structured prediction
Feature scaling	Training set
Boosting (machine learning)	Meta learning (computer science)
Prior probability	Kernel method
Curse of dimensionality	Supervised learning
Scientific evidence	Generalization error
Online machine learning	Overfitting
N-gram	Multi-task learning
Cluster analysis	Generative model
Dimensionality reduction	Computational learning theory
Functional decomposition	Inductive bias
Bayesian network	Semi-supervised learning

Similar topics to “Machine Learning” returned by LDA and Doc2Vec

Skip Thought Vectors (Kiros et al 2015)



Given a tuple (s_{i-1}, s_i, s_{i+1}) of contiguous sentences, with s_i the i-th sentence of a book, the sentence s_i is encoded and tries to reconstruct the previous sentence $s_{i-1}$and next sentence s_{i+1}.

In this example, the input is the sentence triplet *I got back home*. *I could see the cat on the steps. This was strange.* Unattached arrows are connected to the encoder output. Colors indicate which components share parameters. <eos> is the end of sentence token.

Skip Thought Vector

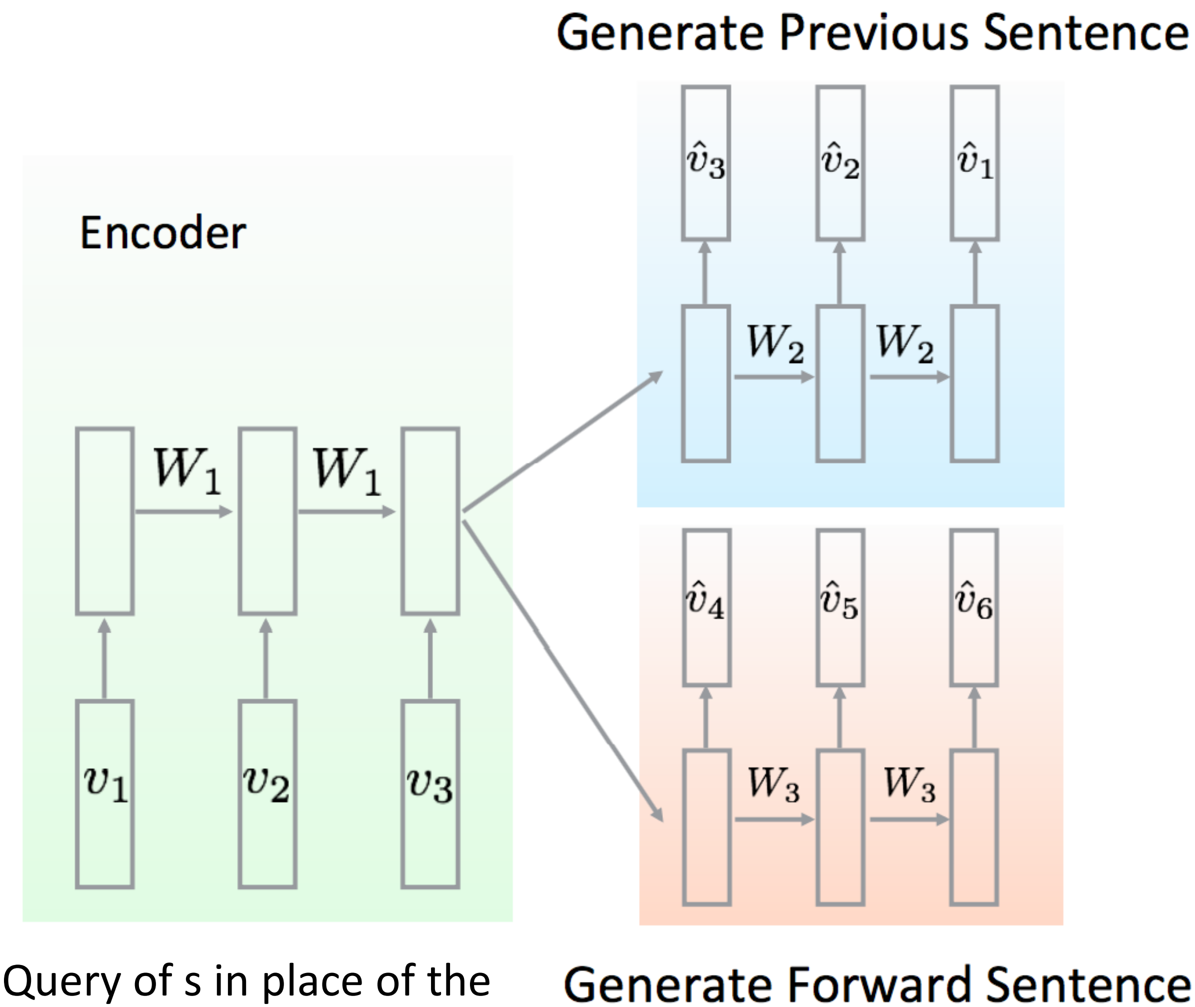
(Ryan Kiros et al. 2015)

- Semantic relatedness:

Sentence 1	**Sentence 2**	**GT**	**pred**
A little girl is looking at a woman in costume	A young girl is looking at a woman in costume	4.7	4.5
A little girl is looking at a woman in costume	The little girl is looking at a man in costume	3.8	4.0
A little girl is looking at a woman in costume	A little girl in costume looks like a woman	2.9	3.5
A sea turtle is hunting for fish	A sea turtle is hunting for food	4.5	4.5
A sea turtle is not hunting for fish	A sea turtle is hunting for fish	3.4	3.8
A man is driving a car	The car is being driven by a man	5	4.9
There is no man driving the car	A man is driving a car	3.6	3.5

GT is ground truth relatedness, Pred is prediction by trained model.

Skip thought vector for phrases

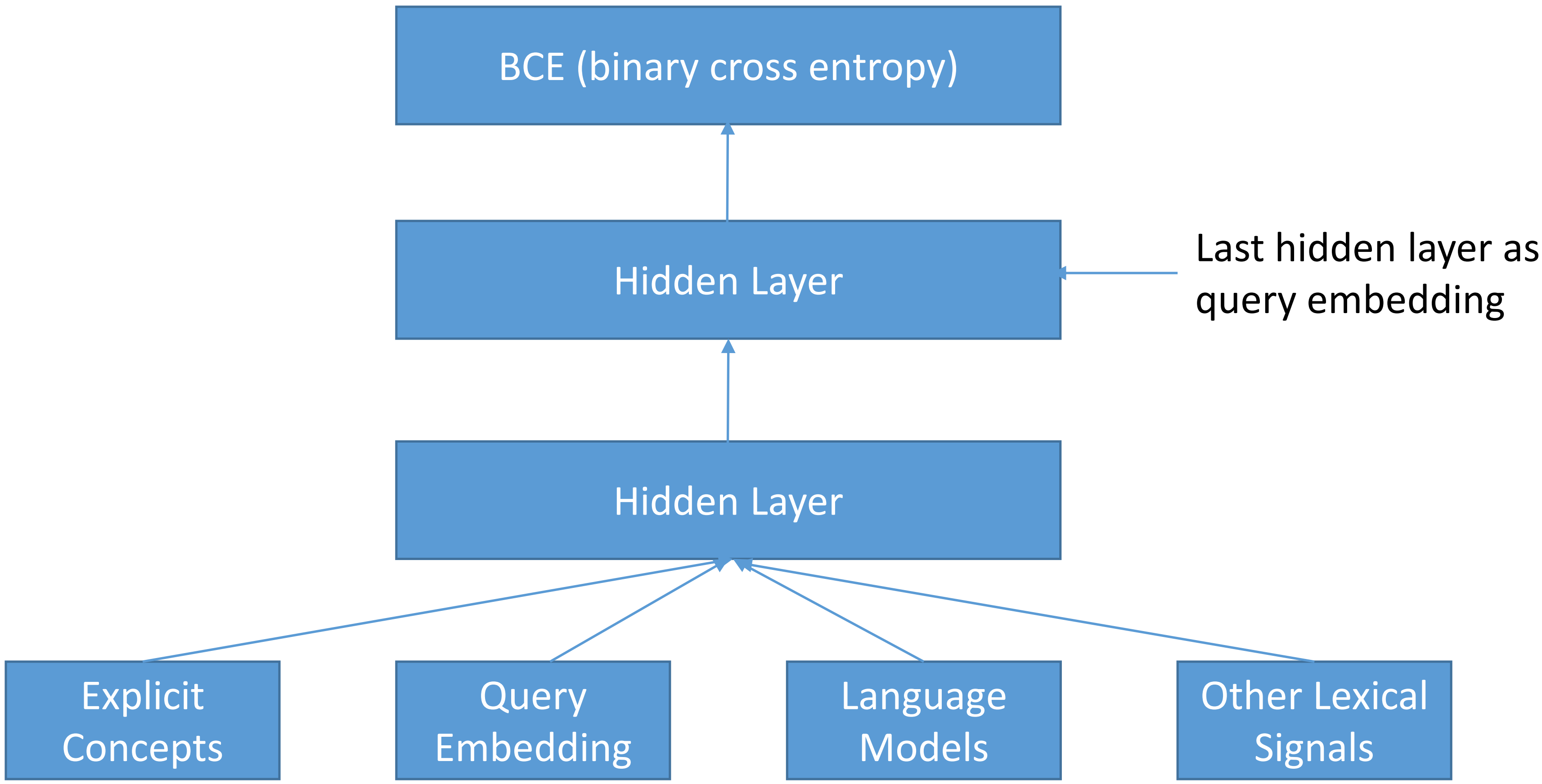


Query of s in place of the original, clicked sentence s

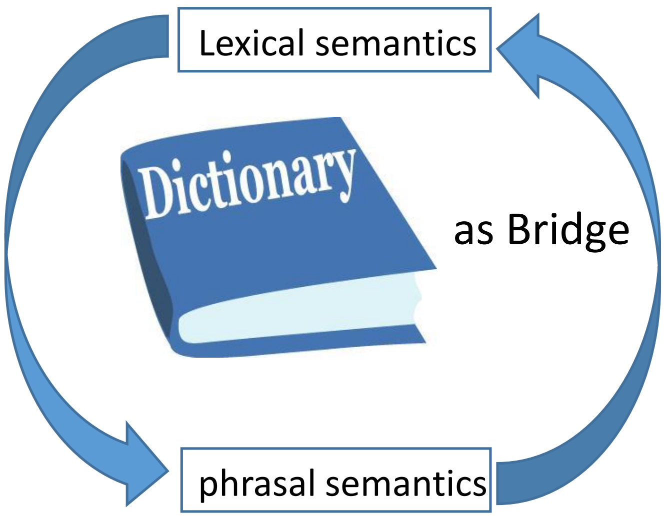
Translation vs. Syntactic context

- Different property of representation
- Different perplexity
- Different applications

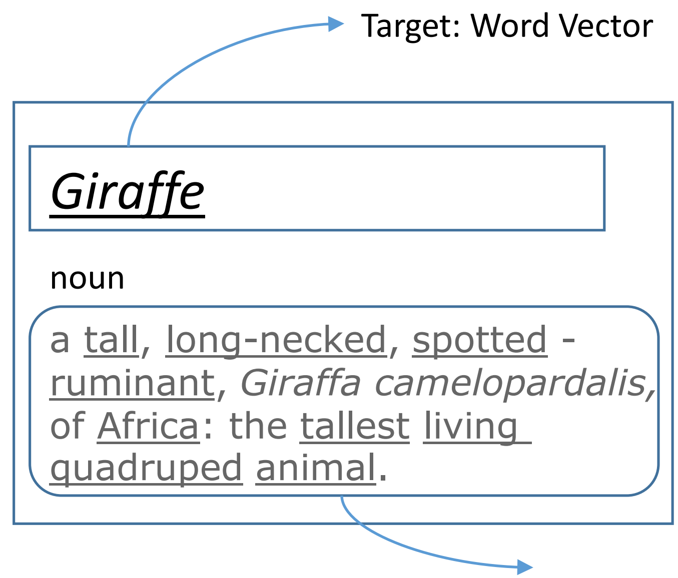
Phrase Embedding: Using a Multi-label Classifier



Phrase Embedding: Using a Dictionary



Goal: From word representation to phrase and sentence representation



The representation of definition should be closed with defining word vector.

Felix Hill, Kyunghyun Cho, Anna Korhonen, Yoshua Bengio:
Learning to Understand Phrases by Embedding the Dictionary. TACL 4: 17-30 (2016)

Phrase Embedding: Using a Dictionary

Model:

Pre-trained Input Representation $\xrightarrow{\text{input}}$ Neural Language Model $\xrightarrow{\text{optimize}}$ Objective Function

Pre-trained Input Representation

Definitions

"control consisting of a mechanical device for controlling fluid flow"

"when you like one thing more than another thing"

.
.
.

Word2Vec as each word's representation

Neural Language Model

$$A_t = \phi(Uv_{t-1} + Wv_t + b)$$

Recurrent Neural Networks

or

Bag-of-Words

$$A_t = A_{t-1} + Wv_t$$

Objective Function

Words

"Valve"

"Prefer"

$$\max(0, m - \cos(M(s_c), v_c) - \cos(M(s_c), v_r))$$

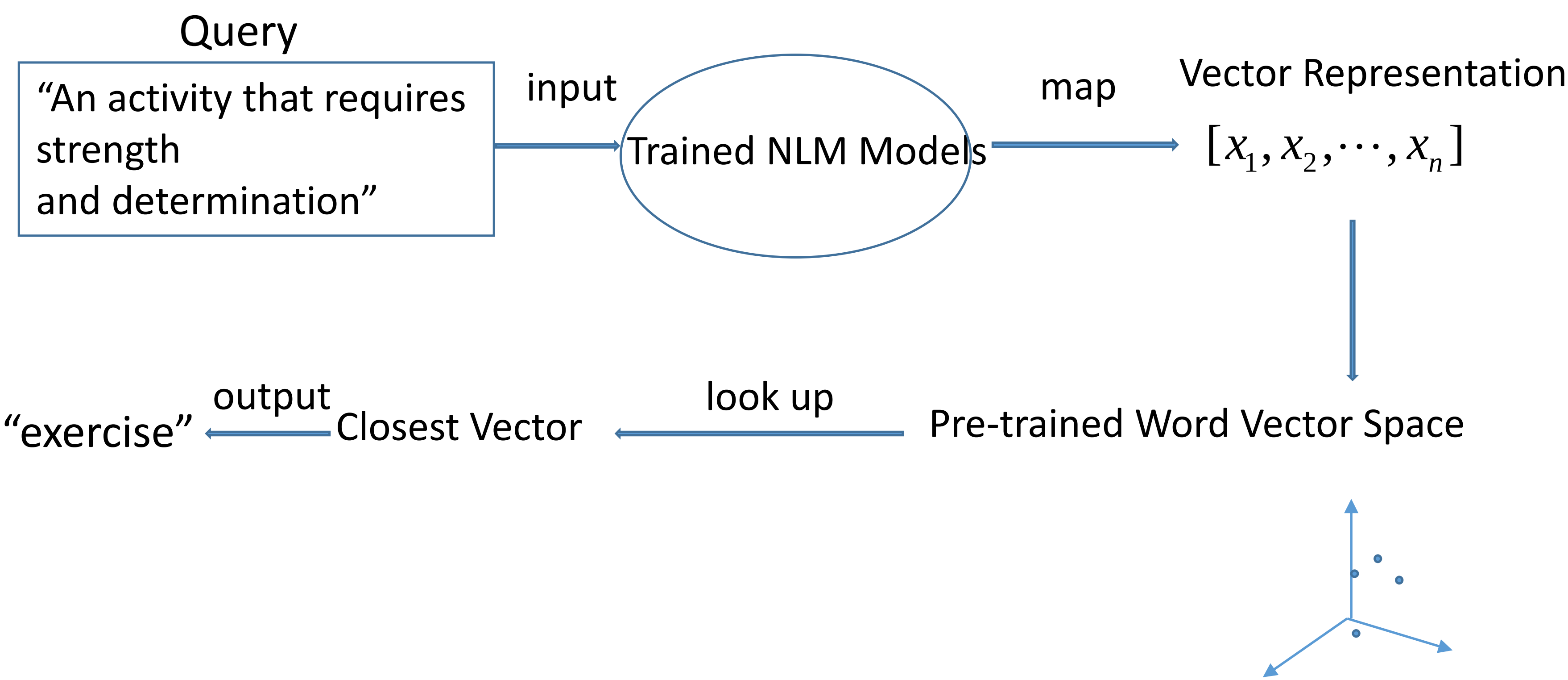
s_c : input phrase embedding

v_c : pre-trained embedding of defining word

v_r : randomly selected word from vocabulary

Phrase Embedding: Using a Dictionary

- Application: Reverse Dictionaries
  - Given a test description, definition, or question, all models produce a ranking of possible word answers based on the proximity of their representations of the input phrase and all possible output words.

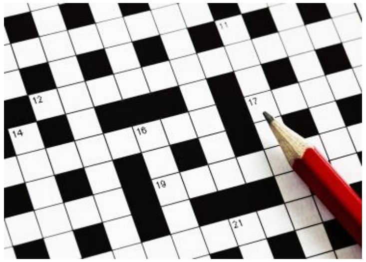


Phrase Embedding: Using a Dictionary

- Application: Crossword Question Answering
  - Given the absence of a knowledge base or web-scale information in our architecture, they narrow the scope of the task by focusing on general knowledge crossword questions

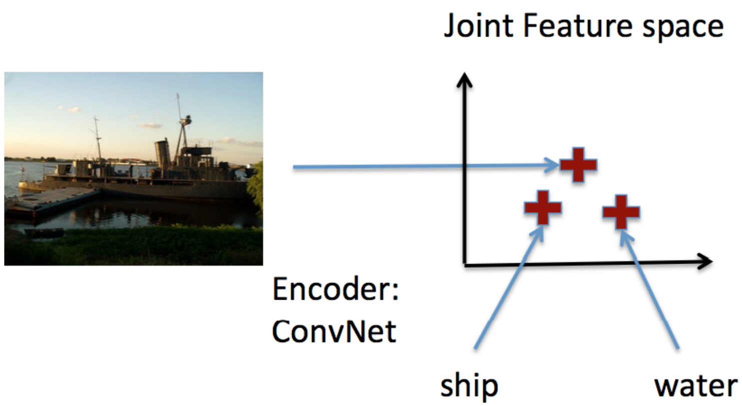
Test set	Description	Word
Long (150 Char)	“French poet and key figure in the development of Symbolism”	Baudelaire
Short (120 Char)	”devil devotee”	satanist
Single-Word (30 Char)	”culpability”	guilt

+ several constrains to reduce the target space



Learning to Understand Phrases by Embedding the Dictionary (Felix Hill et al. 2016)

Phrase Embedding: Using Images



A Deep Visual-Semantic Embedding Model, NIPS 2013

Zero-Shot Learning Through Cross-Modal Transfer, NIPS 2013

Caption: a girl in a blue shirt is on a swing

Keywords: girl, blue shirt, swing

Phrase Embedding: Using Images

- (image, query)

- But image maps to multiple queries

- (*image*, girl)
- (*image*, blue shirt)
- (*image*, swing)

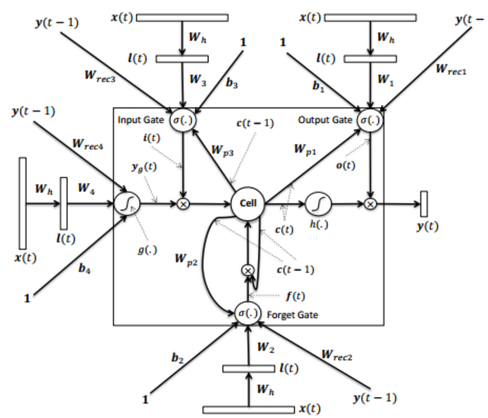
- The image places unnecessary constraint on the 3 queries.

Query Embedding: Using clicked data

Query Side: Shanghai Hotel

(CTR data indicates the semantic relation between Query Side and Document Side)

Document Side: “shanghai hotels accommodation hotel in shanghai discount and reservation”



Basic LSTM architecture for sentence embedding

Deep Sentence Embedding Using Long Short-Term Memory Networks, Palangi et al 2016

Latent Semantic Model with Convolutional-Pooling Structure
(Yelong Shen, et al. 2014)

Search Engine	Microsoft *office* excel
Search Engine	welcome to the apartment *office*

Query examples on internet

what's the meaning of *office* ?

Traditional method: Bag-of-Words   Contextual Information   *office 1 = office 2*

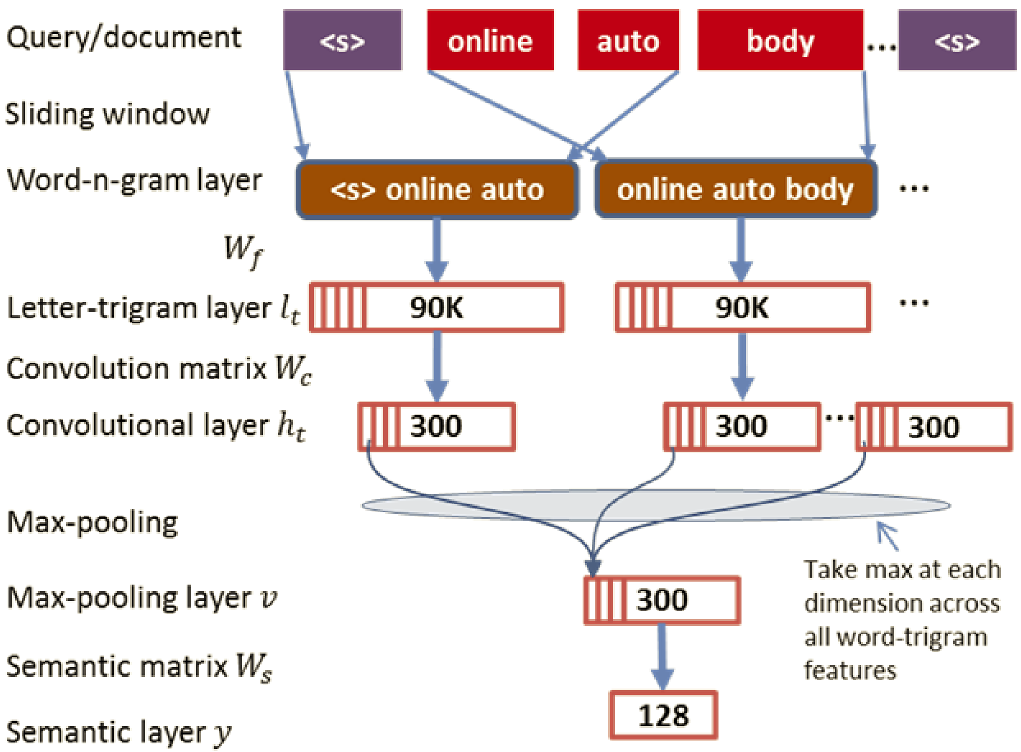
Word Sequence + convolutional-pooling structure   office 1 $\neq$ office 2

low-dimentional, semantic vector representations
for search queries and web document

Shen, Yelong, et al. "A latent semantic model with convolutional-pooling structure for information retrieval." Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. ACM, 2014.

Latent Semantic Model with Convolutional-Pooling Structure
(Yelong Shen, et al. 2014)

- Models:



The CLSM maps a **variable-length** word sequence to **a low-dimensional vector** in a latent semantic space.

Latent Semantic Model with Convolutional-Pooling Structure

(Yelong Shen, et al. 2014)

- Models:

$$f(boy) = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

Indices of *#-b-o*, *b-o-y*, *o-y-#* in the letter-tri-gram list, respectively.

is word boundary symbol

Letter-trigram based Word-n-gram Representation

Concatenating

$$l_t = [f_{t-d}^T, \ldots, f_t^T, \ldots, f_{t+d}^T]^T, \qquad t = 1, \ldots, T$$

Word trigram vector

Convolution operation

$$h_t = tanh(W_c \cdot l_t), \qquad t = 1, \ldots, T$$

Variable length sequence of feature vectors

max pooling

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

microsoft **office excel** could allow remote **code execution**
welcome to the **apartment office**
online **body fat** percentage **calculator**
online **auto body** repair **estimates**
vitamin a the **health** benefits given by **carrots**

Bold words win max operation

Latent Semantic Model with Convolutional-Pooling Structure

(Yelong Shen, et al. 2014)

- Models:

    - Latent Semantic Vector Representations

$$y = \tanh(W_s \cdot v)$$

v is the global feature vector after max pooling, W_s is the semantic projection matrix, and *y* is the vector representation of the input query.

Using cosine similarity to measure relatedness between queries and documents

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

Summary

- Bag of words not powerful enough unless we have huge amount of high quality pairs.

- Web queries are not phrases. Simple composition or phrase translation does not work for web queries.

- Sentiment, classification as targets are not powerful enough to capture full semantics.

- Translation is a better target, as it forces the representation to contain the full semantics.

Conclusion

For short text understanding:

- Short text’s understanding is still hard because the complexity meaning of combining the word in short text, the absence of certain context and syntactical structure.

- There are not very suitable embedding approaches. But just like Hamid’s work, we can incorporate some external data to help do the similarity measurement.

- Word Embedding can be a good feature but not the only feature, we can utilize more NLP tools such as POS or Entity Recognition to do disambiguation.

Reference

[Bengio et al. 2003] Yoshua Bengio, Réjean Ducharme ,Pascal Vincent, Christian Jauvin. A Neural Probabilistic Language Model. In Journal of Machine Learning Research 3 (2003) 1137–1155.

[Mikolov et al. 2013a] Mikolov, Tomas, Chen, Kai, Corrado, Greg, and Dean, Jeffrey. Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013)

[Mikolov et al. 2013b] Mikolov, Tomas, et al. Distributed representations of words and phrases and their compositionality.In NIPS, 2013.

[Pennington et al. 2014] J Pennington, R Socher, CD Manning. Glove: Global Vectors for Word Representation.In EMNLP 2014, 1532-1543.

[Socher et al. 2011] Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, Christopher D. Manning. Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. In NIPS 2011: 801-809

[Cho et al. 2014] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.In EMNLP 2014: 1724-1734

[Gao et al. 2013] Jianfeng Gao, Xiaodong He, Wen-tau Yih, Li Deng:Learning Semantic Representations for the Phrase Translation Model. CoRR abs/1312.0482 (2013)

Reference

[Quoc et al. 2014] Quoc V. Le, Tomas Mikolov: Distributed Representations of Sentences and Documents.In ICML 2014: 1188-1196.

[Ryan Kiros et al. 2015] Kiros, Ryan, et al. Skip-thought vectors. In NIPS 2015.

[Felix Hill et al. 2016] Felix Hill, Kyunghyun Cho, Anna Korhonen, Yoshua Bengio. Learning to Understand Phrases by Embedding the Dictionary.In TACL 4: 17-30 (2016)

[Hamid et al. 2016] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, Rabab K. Ward. Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval. IEEE/ACM Trans. Audio, Speech & Language Processing 24(4): 694-707 (2016)

[Shen, et al. 2014]   Shen et al. A latent semantic model with convolutional-pooling structure for information retrieval. In CIKM 2014.

[Socher et al. 2012] R. Socher, B. Huval, C. Manning and A. Ng. Semantic Compositionality through Recursive Matrix-Vector Spaces. In EMNLP 2012.

[Socher et al. 2013a] R. Socher, J. Bauer, C. Manning and A. Ng. Parsing with Compositional Vector Grammars. In ACL 2013.

[Socher et al. 2013b] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng and C. Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In EMNLP 2013.

Part IV: Conclusion

Zhongyuan Wang (Microsoft Research)

Haixun Wang (Facebook Inc.)

Tutorial Website:
http://www.wangzhongyuan.com/tutorial/ACL2016/Understanding-Short-Texts/

Takeaways

- **Short text understanding** is challenging because of it is *sparse*, *noisy*, and *ambiguous*
- **Short text understanding** can benefit lots of applications such as *search engines*, *automatic question answering*, *online advertising*, *recommendation systems*, and *conversational bot*.

Takeaways

- **Explicit short text understanding**: using *semantic networks* or *knowledge bases* for short text *segmentation*, *sense disambiguation*, *syntax structure*, and *semantic similarity*
  - Single instance
    - Is this instance ambiguous?
    - What are its basic-level concepts?
    - What are its similar instances?
  - Short text
    - How to segment this short text?
    - What does this short text mean (its intent, senses, or concepts)?
    - What are the relations among terms in the short text?
    - How to calculate the similarity between short texts?

Takeaways

- **Implicit short text understanding**: leverage *big data* and *learning based techniques* (especially deep learning) to model *latent semantic representation* for short texts
  - Word level
    - With Distributional Hypothesis
    - Without Distributional Hypothesis
  - Phrase/ Short Text Level
    - Using contextual knowledge
    - Using External Knowledge
  - Short Text/Sentence Level
    - Composition Models
    - Non-Composition Models

Contacts

- Zhongyuan Wang, Microsoft Research (wzhy@outlook.com; zhy.wang@microsoft.com)
- Haixun Wang, Facebook Inc. (haixun@fb.com)

- News:
  - Microsoft Research just release their data and model for short text understanding:
    - ***https://concept.research.microsoft.com/***