Text Mining
(Part I: Phrase Mining & Entity Typing)

JIAWEI HAN
COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

MARCH 12, 2017

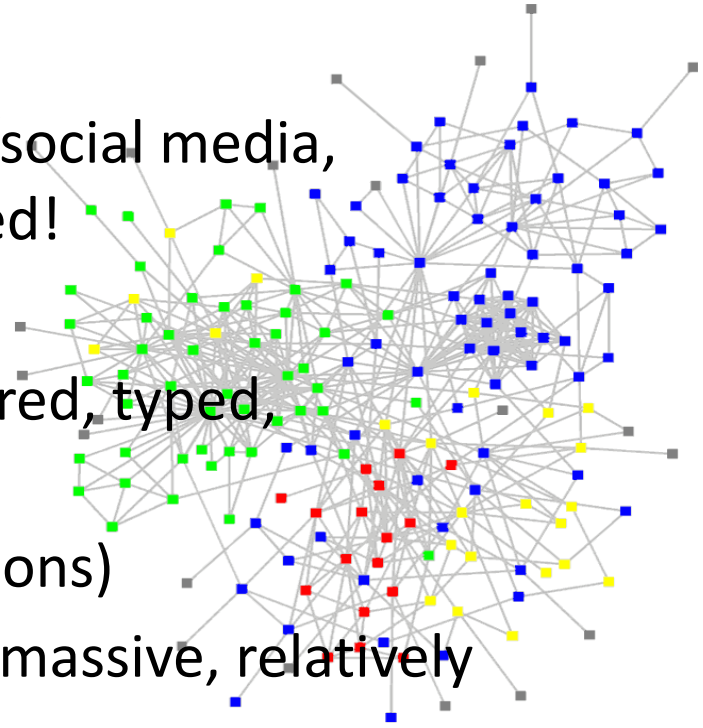
Outline

- Why Text to Networks?
- Phrase Mining and Topic Modeling from Large Text Corpora
  - Why Phrase Mining
  - Previous Approaches
  - TopMine
  - SegPhrase and AutoPhrase
- Entity Recognition and Typing for Massive Text Data
  - ClusType: Entity Extraction and Typing by Relational Graph Construction and Propagation
  - Refined Entity Typing: PLE and AFET
  - CoType: Joint Typing of Entities and Relations

Why Text to Networks?

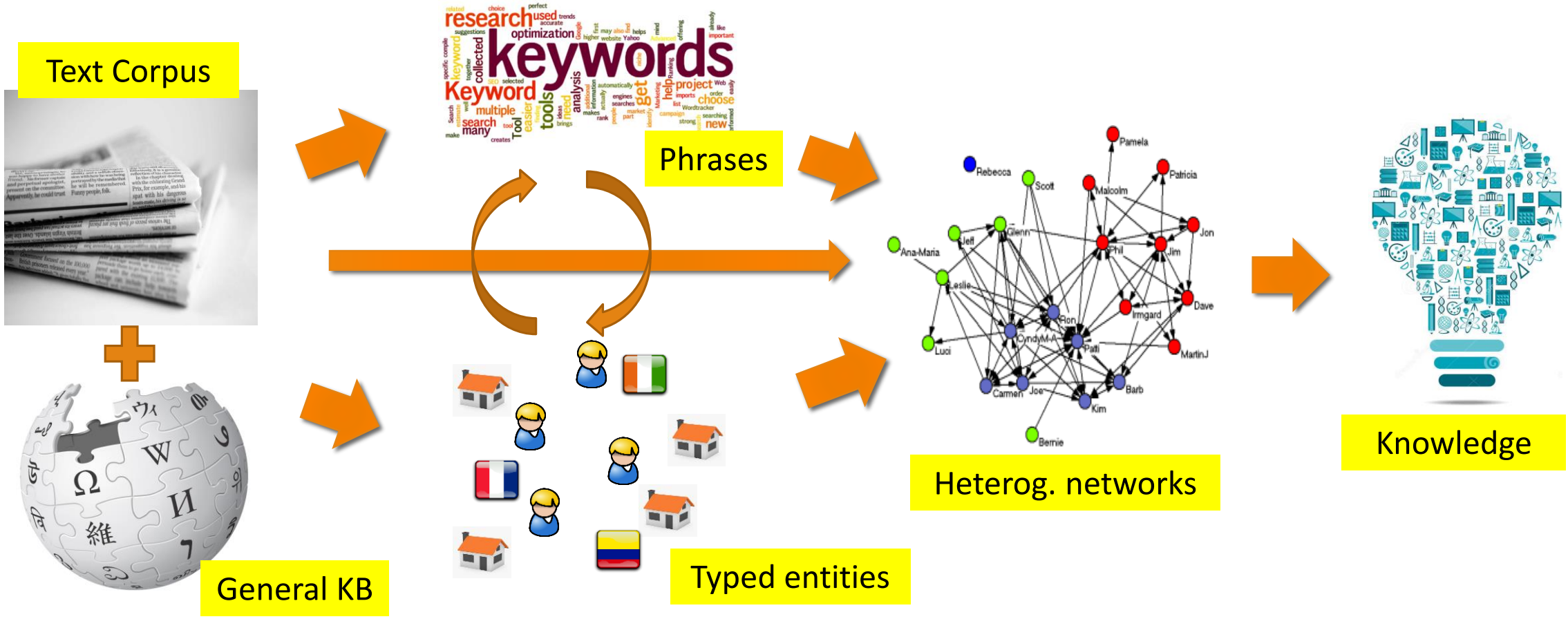
- The major challenge for data scientists of our time
  - The Data-to-Knowledge (D2K) challenge
- Big Data: Over 80% of our data is from text/natural language/social media, unstructured, noisy, dynamic, unreliable, …, but interconnected!
- Keys from big data to big knowledge:
  - Structuring (i.e., transforming unstructured text into structured, typed, interconnected entities/relationships)
  - Networking (take advantage of massive, structured connections)
  - Mining/reasoning (e.g., induction/deduction) effectively on massive, relatively structured, interconnected networks
- D2K game → D2N2K (Data to Network to Knowledge) game

→ Construction and mining of typed, heterogeneous information networks



Construction: Structures Facilitate Information Mining

- Network construction: Generate structured networks from unstructured text data
  - Automated mining of phrases, topics, entities, links and types from large corpora
  - Constructing types, heterogeneous networks from mined data



Outline

- Why Text to Networks?
- Phrase Mining and Topic Modeling from Large Text Corpora
  - Why Phrase Mining
  - Previous Approaches
  - TopMine
  - SegPhrase and AutoPhrase
- Entity Recognition and Typing for Massive Text Data
  - ClusType: Entity Extraction and Typing by Relational Graph Construction and Propagation
  - Refined Entity Typing: PLE and AFET
  - CoType: Joint Typing of Entities and Relations

Why Phrase Mining?

- Unigrams vs. phrases
  - **Unigrams** (single words) are *ambiguous*
    - Example: “United”: United States? United Airline? United Parcel Service?
  - **Phrase**: A natural, meaningful, *unambiguous* semantic unit
    - Example: “United States” vs. “United Airline”
- Mining semantically meaningful phrases
  - Transform text data from *word granularity* to *phrase granularity*
  - Enhance the power and efficiency at manipulating unstructured data using database technology

Mining Phrases: Why Not Use NLP Methods?

- Phrase mining: Originated from the NLP community—"Chunking"
  - Model it as a sequence labeling problem (B-NP, I-NP, O, …)
- Need annotation and training
  - Annotate hundreds of documents as training data
  - Train a supervised model based on part-of-speech features
- Recent trend:
  - Use distributional features based on web n-grams (Bergsma et al., 2010)
  - State-of-the-art performance: ~95% accuracy, ~88% phrase-level F-score
- Limitations
  - High annotation cost, not scalable to a new language, a new domain or genre
  - May not fit domain-specific, dynamic, emerging applications
    - Scientific domains, query logs, or social media, e.g., Yelp, Twitter

Data Mining Approaches

- General principle: Fully exploit information redundancy and data-driven criteria to determine phrase boundaries and salience
- Phrase Mining and Topic Modeling from Large Corpora
  - Strategy 1: Simultaneously Inferring Phrases and Topics
  - Strategy 2: Post Topic Modeling Phrase Construction
  - Strategy 3: First Phrase Mining then Topic Modeling (ToPMine)
- Integration of Phrase Mining with Document Segmentation

Frequent Pattern Mining for Text Data: Phrase Mining and Topic Modeling

- Motivation: Unigrams (single words) can be difficult to interpret
- Ex.: The topic that represents the area of Machine Learning

learning
reinforcement
support
machine
vector
selection
feature
random
:

versus

learning
support vector machines
reinforcement learning
feature selection
conditional random fields
classification
decision trees
:

Outline

- Why Text to Networks?
- Phrase Mining and Topic Modeling from Large Text Corpora
  - Why Phrase Mining
  - Previous Approaches
  - TopMine
  - SegPhrase and AutoPhrase
- Entity Recognition and Typing for Massive Text Data
  - ClusType: Entity Extraction and Typing by Relational Graph Construction and Propagation
  - Refined Entity Typing: PLE and AFET
  - CoType: Joint Typing of Entities and Relations

Various Strategies: Phrase-Based Topic Modeling

- Strategy 1: Generate bag-of-words → generate sequence of tokens
  - Bigram topical model [Wallach'06], topical n-gram model [Wang, et al.'07], phrase discovering topic model [Lindsey, et al.'12]
- Strategy 2: Post bag-of-words model inference, visualize topics with n-grams
  - Label topic [Mei et al.'07], TurboTopic [Blei & Lafferty'09], KERT [Danilevsky, et al.'14]
- Strategy 3: Prior bag-of-words model inference, mine phrases and impose on the bag-of-words model
  - ToPMine [El-kishky, et al.'15]

Strategy 1: Simultaneously Inferring Phrases and Topics

- Bigram Topic Model **[Wallach'06]**
  - Probabilistic generative model that conditions on previous word and topic when drawing next word
- Topical N-Grams (TNG) **[Wang, et al.'07]**
  - Probabilistic model that generates words in textual order
  - Create n-grams by concatenating successive bigrams (a generalization of Bigram Topic Model)
- Phrase-Discovering LDA (PDLDA) **[Lindsey, et al.'12]**
  - Viewing each sentence as a time-series of words, PDLDA posits that the generative parameter (topic) changes periodically
  - Each word is drawn based on previous m words (context) and current phrase topic
- High model complexity: Tends to overfitting; High inference cost: Slow

TNG: Experiments on Research Papers

Reinforcement Learning			Human Receptive System		
LDA	n-gram (2+)	n-gram (1)	LDA	n-gram (2+)	n-gram (1)
state	reinforcement learning	action	motion	receptive field	motion
learning	optimal policy	policy	visual	spatial frequency	spatial
policy	dynamic programming	reinforcement	field	temporal frequency	visual
action	optimal control	states	position	visual motion	receptive
reinforcement	function approximator	actions	figure	motion energy	response
states	prioritized sweeping	function	direction	tuning curves	direction
time	finite-state controller	optimal	fields	horizontal cells	cells
optimal	learning system	learning	eye	motion detection	figure
actions	reinforcement learning rl	reward	location	preferred direction	stimulus
function	function approximators	control	retina	visual processing	velocity
algorithm	markov decision problems	agent	receptive	area mt	contrast
reward	markov decision processes	q-learning	velocity	visual cortex	tuning
step	local search	goal	vision	light intensity	moving
dynamic	state-action pair	space	moving	directional selectivity	model
control	markov decision process	step	system	high contrast	temporal
sutton	belief states	environment	flow	motion detectors	responses
rl	stochastic policy	system	edge	spatial phase	orientation
decision	action selection	problem	center	moving stimuli	light
algorithms	upright position	steps	light	decision strategy	stimuli
agent	reinforcement learning methods	transition	local	visual stimuli	cell

TNG: Experiments on Research Papers (2)

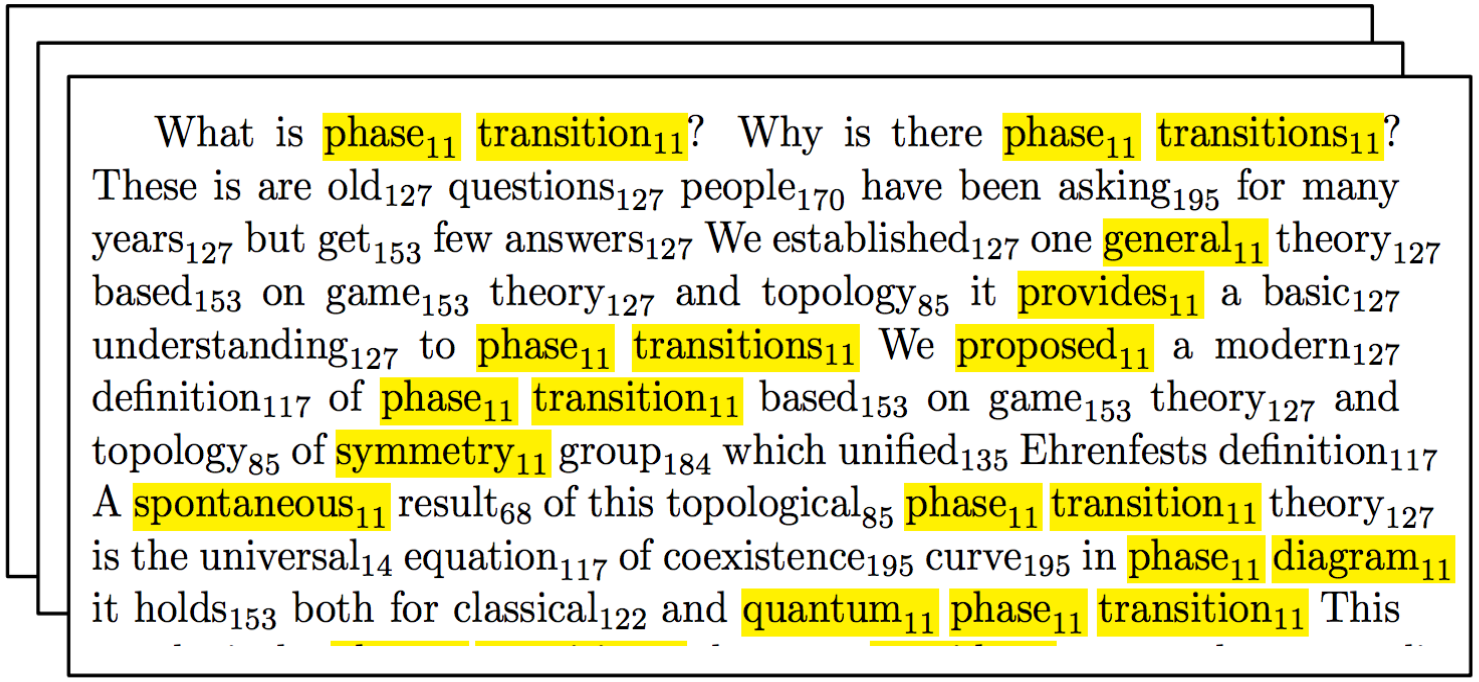
Speech Recognition			Support Vector Machines		
LDA	n-gram (2+)	n-gram (1)	LDA	n-gram (2+)	n-gram (1)
recognition	speech recognition	speech	kernel	support vectors	kernel
system	training data	word	linear	test error	training
word	neural network	training	vector	support vector machines	support
face	error rates	system	support	training error	margin
context	neural net	recognition	set	feature space	svm
character	hidden markov model	hmm	nonlinear	training examples	solution
hmm	feature vectors	speaker	data	decision function	kernels
based	continuous speech	performance	algorithm	cost functions	regularization
frame	training procedure	phoneme	space	test inputs	adaboost
segmentation	continuous speech recognition	acoustic	pca	kkt conditions	test
training	gamma filter	words	function	leave-one-out procedure	data
characters	hidden control	context	problem	soft margin	generalization
set	speech production	systems	margin	bayesian transduction	examples
probabilities	neural nets	frame	vectors	training patterns	cost
features	input representation	trained	solution	training points	convex
faces	output layers	sequence	training	maximum margin	algorithm
words	training algorithm	phonetic	svm	strictly convex	working
frames	test set	speakers	kernels	regularization operators	feature
database	speech frames	mlp	matrix	base classifiers	sv
mlp	speaker dependent	hybrid	machines	convex optimization	functions

Strategy 2: Post Topic Modeling Phrase Construction

- **TurboTopics** [Blei & Lafferty'09] – Phrase construction as a post-processing step to Latent Dirichlet Allocation
  - Perform Latent Dirichlet Allocation on corpus to assign each token a topic label
  - Merge adjacent unigrams with the same topic label by a distribution-free permutation test on arbitrary-length back-off model
  - End recursive merging when all significant adjacent unigrams have been merged
- **KERT** [Danilevsky et al.'14] – Phrase construction as a post-processing step to Latent Dirichlet Allocation
  - Perform **frequent pattern mining** on each topic
  - Perform **phrase ranking** based on four different criteria

Example of TurboTopics

Annotated documents



What is phase$_{11}$ transition$_{11}$? Why is there phase$_{11}$ transitions$_{11}$? These is are old$_{127}$ questions$_{127}$ people$_{170}$ have been asking$_{195}$ for many years$_{127}$ but get$_{153}$ few answers$_{127}$ We established$_{127}$ one general$_{11}$ theory$_{127}$ based$_{153}$ on game$_{153}$ theory$_{127}$ and topology$_{85}$ it provides$_{11}$ a basic$_{127}$ understanding$_{127}$ to phase$_{11}$ transitions$_{11}$ We proposed$_{11}$ a modern$_{127}$ definition$_{117}$ of phase$_{11}$ transition$_{11}$ based$_{153}$ on game$_{153}$ theory$_{127}$ and topology$_{85}$ of symmetry$_{11}$ group$_{184}$ which unified$_{135}$ Ehrenfests definition$_{117}$ A spontaneous$_{11}$ result$_{68}$ of this topological$_{85}$ phase$_{11}$ transition$_{11}$ theory$_{127}$ is the universal$_{14}$ equation$_{117}$ of coexistence$_{195}$ curve$_{195}$ in phase$_{11}$ diagram$_{11}$ it holds$_{153}$ both for classical$_{122}$ and quantum$_{11}$ phase$_{11}$ transition$_{11}$ This

LDA topic #11

phase, transitions, phases, transition, quantum, critical, symmetry, field, point, model, order, diagram, systems, two, theory, system, study, breaking, spin, first

Turbo topic #11

phase transitions, model, symmetry, point, quantum, systems, phase transition, phase diagram, system, order, field, order, parameter, critical, two, transitions in, models, different, symmetry breaking, first order, phenomena

- Perform LDA on corpus to assign each token a topic label
  - E.g., … phase$_{11}$ transition$_{11}$ …. game$_{153}$ theory$_{127}$ …
- Then merge adjacent unigrams with same topic label

KERT: Topical Keyphrase Extraction & Ranking

[Danilevsky, et al. 2014]

knowledge discovery using least squares support vector machine classifiers

support vectors for reinforcement learning

a hybrid approach to feature selection

pseudo conditional random fields

automatic web page classification in a dynamic and hierarchical way

inverse time dependency in convex regularized learning

postprocessing decision trees to extract actionable knowledge

variance minimization least squares support vector machines

…

Unigram topic assignment: Topic 1 & Topic 2

→

learning
support vector machines
reinforcement learning
feature selection
conditional random fields
classification
decision trees
:

Topical keyphrase extraction & ranking

Framework of KERT

1. Run bag-of-words model inference and assign topic label to each token
2. Extract candidate keyphrases within each topic

Frequent pattern mining

3. Rank the keyphrases in each topic
   - Popularity: 'information retrieval' vs. 'cross-language information retrieval'
   - Discriminativeness: only frequent in documents about topic t
   - Concordance: 'active learning' vs. 'learning classification'
   - Completeness: 'vector machine' vs. 'support vector machine'

Comparability property: directly compare phrases of mixed lengths

KERT: Topical Phrases on Machine Learning

Top-Ranked Phrases by Mining Paper Titles in DBLP

kpRel [Zhao et al. 11]	KERT (-popularity)	KERT (-discriminativeness)	KERT (-concordance)	KERT [Danilevsky et al. 14]
learning	effective	support vector machines	learning	learning
classification	text	feature selection	classification	support vector machines
selection	probabilistic	reinforcement learning	selection	reinforcement learning
models	identification	conditional random fields	feature	feature selection
algorithm	mapping	constraint satisfaction	decision	conditional random fields
features	task	decision trees	bayesian	classification
decision	planning	dimensionality reduction	trees	decision trees
:	:	:	:	:

The topic that represents the area of Machine Learning

Outline

- Why Text to Networks?
- Phrase Mining and Topic Modeling from Large Text Corpora
  - Why Phrase Mining
  - Previous Approaches
  - TopMine
  - SegPhrase and AutoPhrase
- Entity Recognition and Typing for Massive Text Data
  - ClusType: Entity Extraction and Typing by Relational Graph Construction and Propagation
  - Refined Entity Typing: PLE and AFET
  - CoType: Joint Typing of Entities and Relations

Strategy 3: First Phrase Mining then Topic Modeling

- **ToPMine** [El-Kishky et al. VLDB'15]
  - First phrase construction, then topic mining
  - Contrast with KERT: topic modeling, then phrase mining
- **The ToPMine Framework:**
  - Perform frequent ***contiguous pattern*** mining to extract candidate phrases and their counts
  - Perform agglomerative merging of adjacent unigrams as guided by a significance score—This segments each document into a ***"bag-of-phrases"***
  - The newly formed bag-of-phrases are passed as input to PhraseLDA, an extension of LDA, that constrains all words in a phrase to each sharing the same latent topic

Why First Phrase Mining then Topic Modeling ?

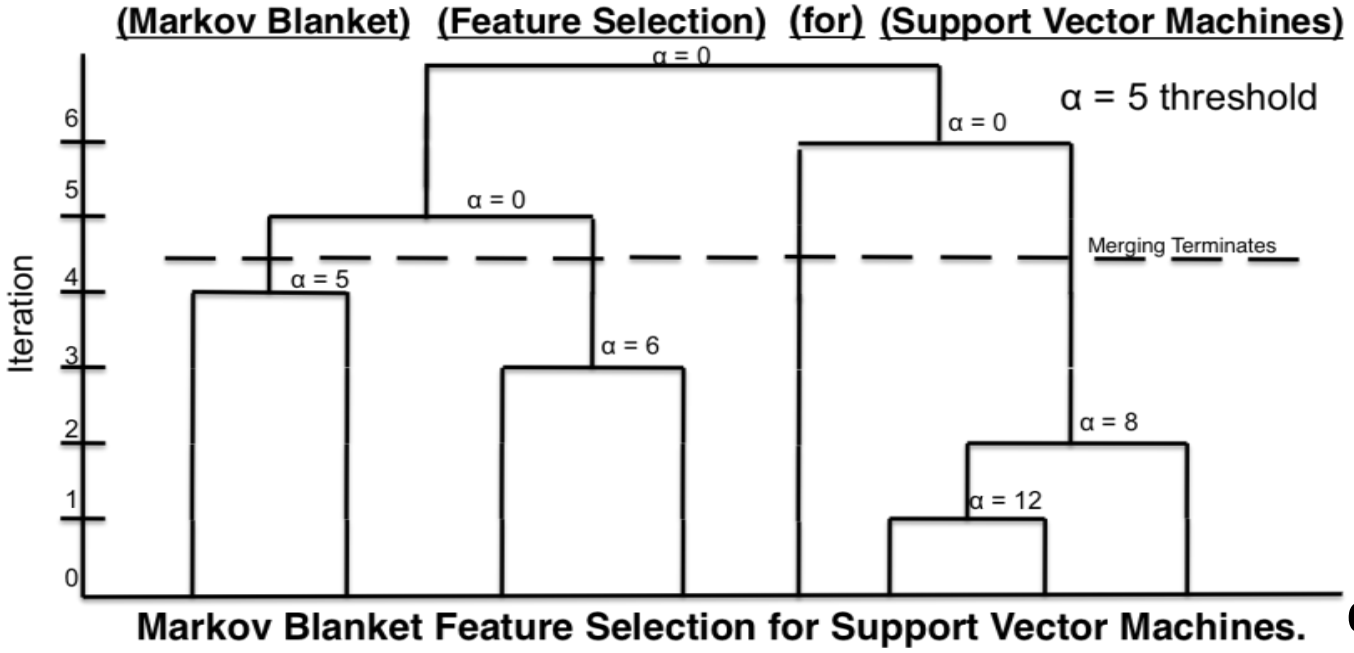
- With Strategy 2, tokens in the same phrase may be assigned to different topics
  - Ex. knowledge discovery using least squares support vector machine classifiers…
  - *Knowledge discovery* and *support vector machine* should have coherent topic labels
- Solution: switch the order of phrase mining and topic model inference

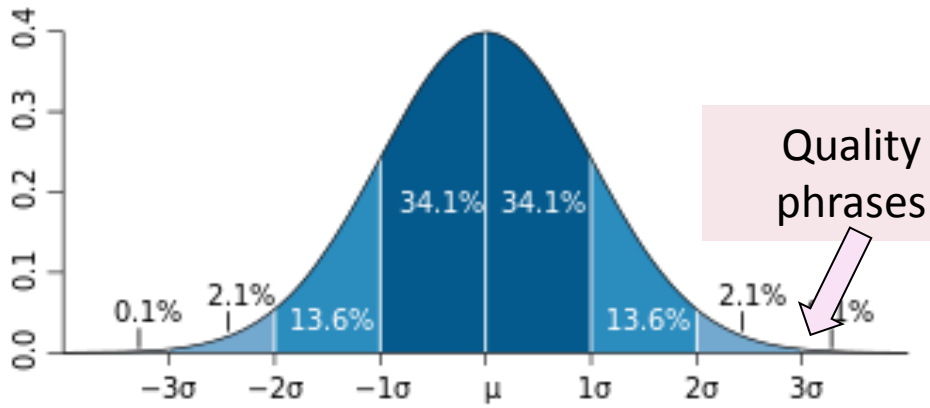
[knowledge discovery] using [least squares] [support vector machine] [classifiers] …

[knowledge discovery] using [least squares] [support vector machine] [classifiers] …

- Techniques
  - Phrase mining and document segmentation
  - Topic model inference with phrase constraint

Phrase Mining: Frequent Pattern Mining + Statistical Analysis





Based on significance score [Church et al.'91]:

$$\alpha(P_1, P_2) \approx (f(P_1 \bullet P_2) - \mu_0(P_1, P_2))/\sqrt{f(P_1 \bullet P_2)}$$

[Markov blanket] [feature selection] for [support vector machines]
[knowledge discovery] using [least squares] [support vector machine] [classifiers]
…[support vector] for [machine learning]…

Phrase	Raw freq.	True freq.
[support vector machine]	90	80
[vector machine]	95	0
[support vector]	100	20

Collocation Mining

- Collocation: A sequence of words that occur more frequently than expected
  - Often "interesting" and due to their non-compositionality, often relay information not portrayed by their constituent terms (e.g., "made an exception", "strong tea")
- Many different measures used to extract collocations from a corpus [Dunning 93, Pederson 96]
  - E.g., mutual information, t-test, z-test, chi-squared test, likelihood ratio

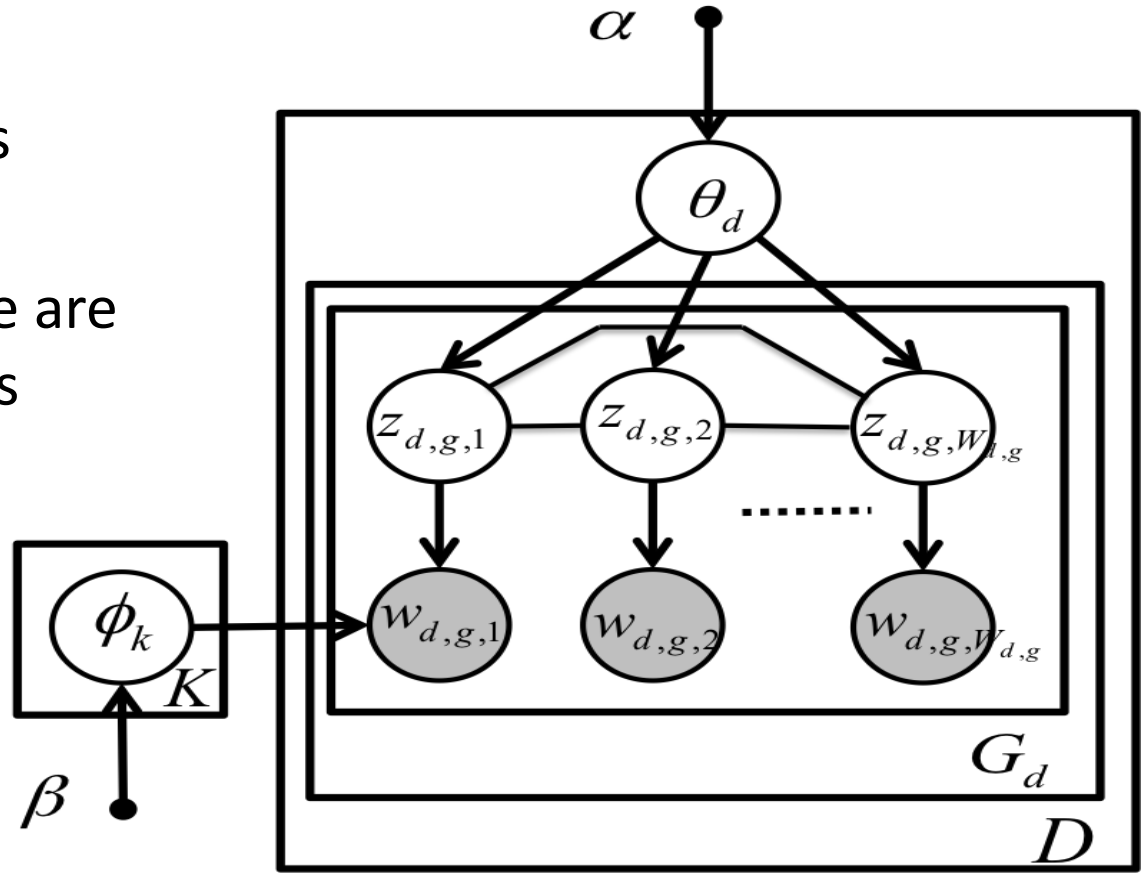
$$\mathrm{PMI}(x,y) = \log \frac{p(x,y)}{p(x)p(y)} \qquad sig = \frac{count(phr_{x+y}) - E[count(phr_{x+y})])}{\sqrt{count(phr_{x+y})}} \qquad \chi^2 = \sum \frac{(O-E)^2}{E}$$

- Many of these measures can be used to guide the agglomerative phrase-segmentation algorithm

ToPMine: Phrase LDA (Constrained Topic Modeling)

- The generative model for PhraseLDA is the same as LDA
- Difference: the model incorporates constraints obtained from the **"bag-of-phrases"** input
  - Chain-graph shows that all words in a phrase are constrained to take on the same topic values

[knowledge discovery] using [least squares] [support vector machine] [classifiers] …



Topic model inference with phrase constraints

Example Topical Phrases: A Comparison

social networks	information retrieval
web search	text classification
time series	machine learning
search engine	support vector machines
management system	information extraction
real time	neural networks
decision trees	text categorization
:	:
Topic 1	Topic 2

PDLDA [Lindsey et al. 12] Strategy 1 (3.72 hours)

information retrieval	feature selection
social networks	machine learning
web search	semi supervised
search engine	large scale
information extraction	support vector machines
question answering	active learning
web pages	face recognition
:	:
Topic 1	Topic 2

ToPMine [El-kishky et al. 14] Strategy 3 (67 seconds)

ToPMine: Experiments on DBLP Abstracts

	Topic 1	*Topic 2*	*Topic 3*	*Topic 4*	*Topic 5*
unigrams	problem	word	data	programming	data
	algorithm	language	method	language	patterns
	optimal	text	algorithm	code	mining
	solution	speech	learning	type	rules
	search	system	clustering	object	set
	solve	recognition	classification	implementation	event
	constraints	character	based	system	time
	programming	translation	features	compiler	association
	heuristic	sentences	proposed	java	stream
	genetic	grammar	classifier	data	large
n-grams	genetic algorithm	natural language	data sets	programming language	data mining
	optimization problem	speech recognition	support vector machine	source code	data sets
	solve this problem	language model	learning algorithm	object oriented	data streams
	optimal solution	natural language processing	machine learning	type system	association rules
	evolutionary algorithm	machine translation	feature selection	data structure	data collection
	local search	recognition system	paper we propose	program execution	time series
	search space	context free grammars	clustering algorithm	run time	data analysis
	optimization algorithm	sign language	decision tree	code generation	mining algorithms
	search algorithm	recognition rate	proposed method	object oriented programming	spatio temporal
	objective function	character recognition	training data	java programs	frequent itemsets

ToPMine: Topics on Associate Press News (1989)

	Topic 1	*Topic 2*	*Topic 3*	*Topic 4*	*Topic 5*
unigrams	plant	church	palestinian	bush	drug
	nuclear	catholic	israeli	house	aid
	environmental	religious	israel	senate	health
	energy	bishop	arab	year	hospital
	year	pope	plo	bill	medical
	waste	roman	army	president	patients
	department	jewish	reported	congress	research
	power	rev	west	tax	test
	state	john	bank	budget	study
	chemical	christian	state	committee	disease
n-grams	energy department	roman catholic	gaza strip	president bush	health care
	environmental protection agency	pope john paul	west bank	white house	medical center
	nuclear weapons	john paul	palestine liberation prganization	bush administration	united states
	acid rain	catholic church	united states	house and senate	aids virus
	nuclear power plant	anti semitism	arab reports	members of congress	drug abuse
	hazardous waste	baptist church	prime minister	defense secretary	food and drug administration
	savannah river	united states	yitzhak shamir	capital gains tax	aids patient
	rocky flats	lutheran church	israel radio	pay raise	centers for disease control
	nuclear power	episcopal church	occupied territories	house members	heart disease
	natural gas	church members	occupied west bank	committee chairman	drug testing

ToPMine: Experiments on Yelp Reviews

	Topic 1	*Topic 2*	*Topic 3*	*Topic 4*	*Topic 5*
unigrams	coffee	food	room	store	good
	ice	good	parking	shop	food
	cream	place	hotel	prices	place
	flavor	ordered	stay	find	burger
	egg	chicken	time	place	ordered
	chocolate	roll	nice	buy	fries
	breakfast	sushi	place	selection	chicken
	tea	restaurant	great	items	tacos
	cake	dish	area	love	cheese
	sweet	rice	pool	great	time
n-grams	ice cream	spring rolls	parking lot	grocery store	mexican food
	iced tea	food was good	front desk	great selection	chips and salsa
	french toast	fried rice	spring training	farmer's market	food was good
	hash browns	egg rolls	staying at the hotel	great prices	hot dog
	frozen yogurt	chinese food	dog park	parking lot	rice and beans
	eggs benedict	pad thai	room was clean	wal mart	sweet potato fries
	peanut butter	dim sum	pool area	shopping center	pretty good
	cup of coffee	thai food	great place	great place	carne asada
	iced coffee	pretty good	staff is friendly	prices are reasonable	mac and cheese
	scrambled eggs	lunch specials	free wifi	love this place	fish tacos

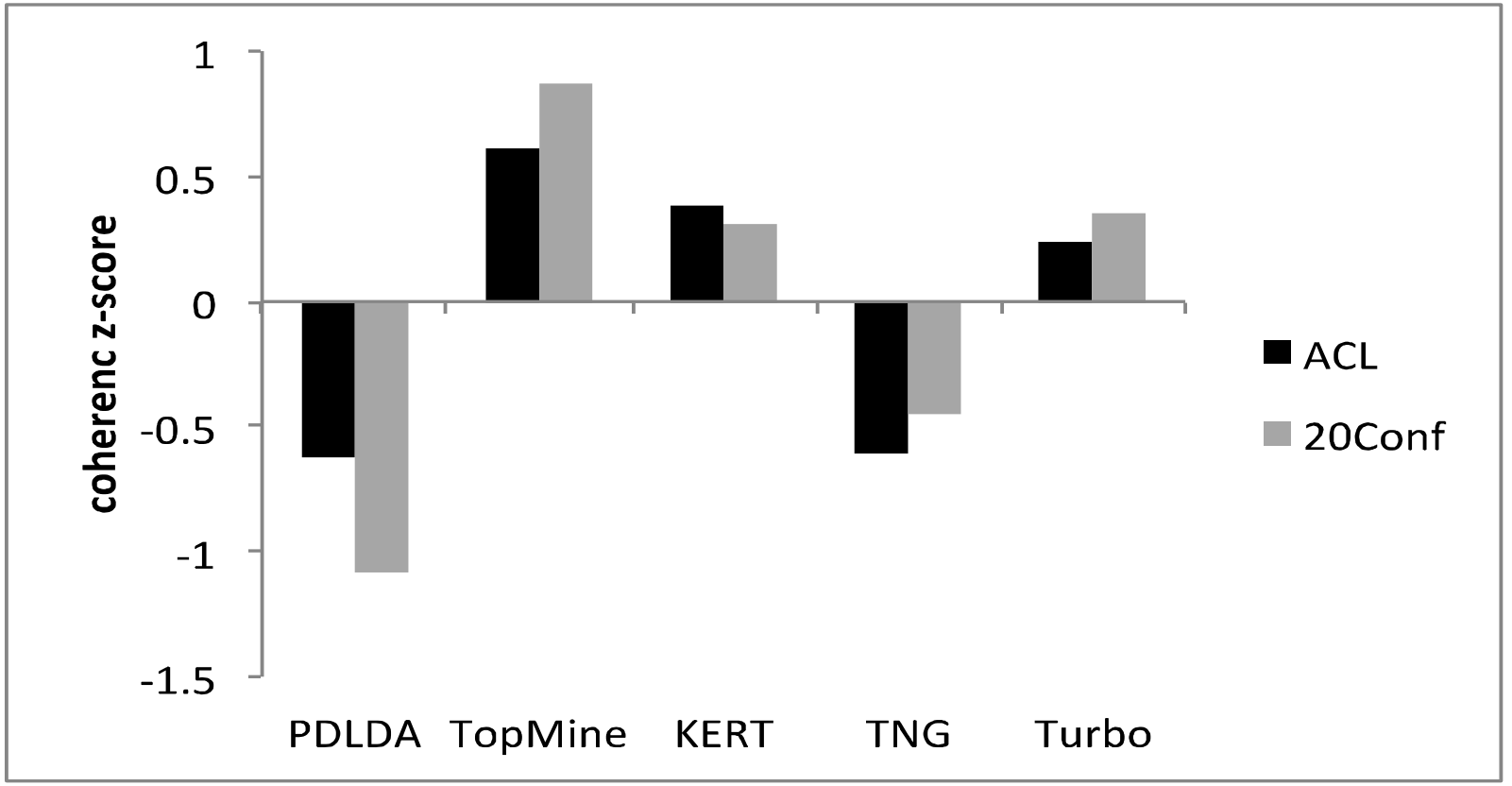
Efficiency: Running Time of Different Strategies

Method	*sampled dblp titles (k=5)*	*dblp titles (k=30)*	*sampled dblp abstracts*	*dblp abstracts*
PDLDA	3.72(hrs)	~20.44(days)	1.12(days)	~95.9(days)
Turbo Topics	6.68(hrs)	>30(days)*	>10(days)*	>50(days)*
TNG	146(s)	5.57 (hrs)	853(s)	NA†
LDA	**65(s)**	3.04 (hrs)	353(s)	13.84(hours)
KERT	68(s)	3.08(hrs)	1215(s)	NA†
ToP-Mine	67(s)	**2.45(hrs)**	**340(s)**	**10.88(hrs)**

Running time: strategy 3 > strategy 2 > strategy 1 (“>” means outperforms)

- Strategy 1: Generate bag-of-words → generate sequence of tokens
- Strategy 2: Post bag-of-words model inference, visualize topics with n-grams
- Strategy 3: Prior bag-of-words model inference, mine phrases and impose to the bag-of-words model

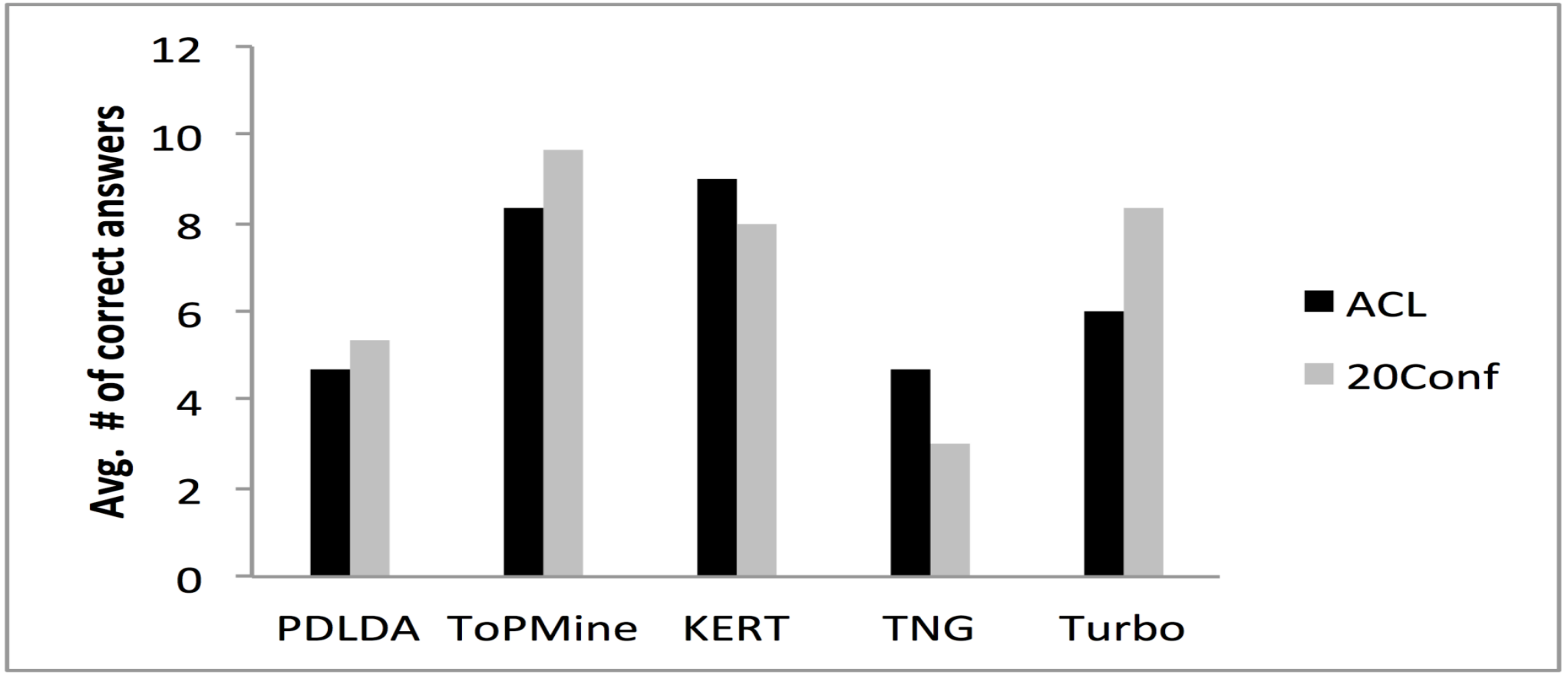
Coherence of Topics: Comparison of Strategies



Coherence measured by z-score: strategy 3 > strategy 2 > strategy 1

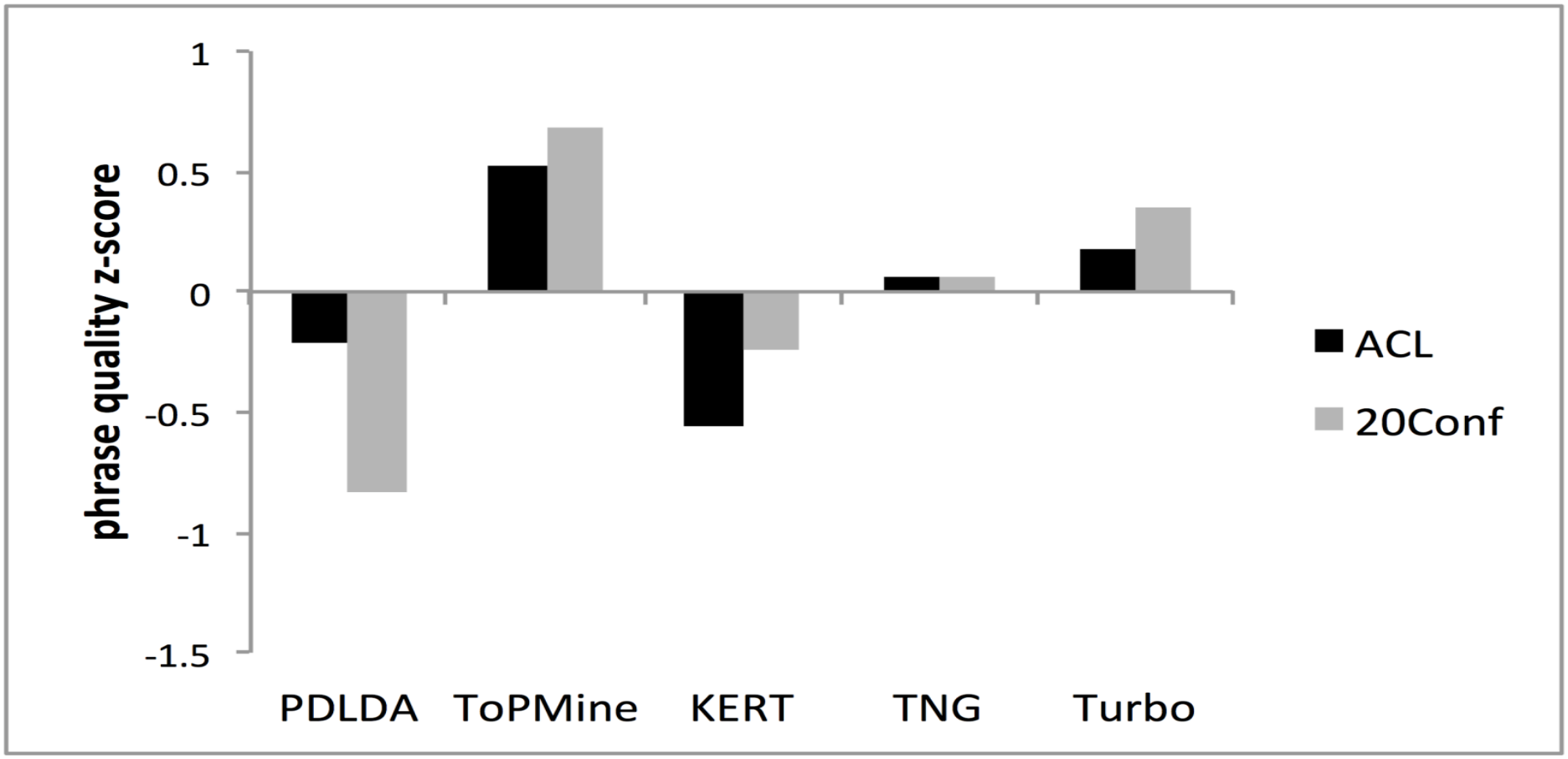
- Strategy 1: Generate bag-of-words → generate sequence of tokens
- Strategy 2: Post bag-of-words model inference, visualize topics with n-grams
- Strategy 3: Prior bag-of-words model inference, mine phrases and impose to the bag-of-words model

Phrase Intrusion: Comparison of Strategies



Phrase intrusion measured by average number of correct answers:
strategy 3 > strategy 2 > strategy 1

Phrase Quality: Comparison of Strategies



Phrase quality measured by z-score:
strategy 3 > strategy 2 > strategy 1

Summary: Strategies on Topical Phrase Mining

- Strategy 1: Generate bag-of-words → generate sequence of tokens
  - Integrated complex model; phrase quality and topic inference rely on each other
  - Slow and overfitting
- Strategy 2: Post bag-of-words model inference, visualize topics with n-grams
  - Phrase quality relies on topic labels for unigrams
  - Can be fast; generally high-quality topics and phrases
- Strategy 3: Prior bag-of-words model inference, mine phrases and impose to the bag-of-words model
  - Topic inference relies on correct segmentation of documents, but not sensitive
  - Can be fast; generally high-quality topics and phrases

Outline

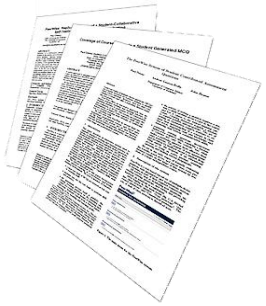
- Why Text to Networks?
- Phrase Mining and Topic Modeling from Large Text Corpora
  - Why Phrase Mining
  - Previous Approaches
  - TopMine
  - SegPhrase and AutoPhrase
- Entity Recognition and Typing for Massive Text Data
  - ClusType: Entity Extraction and Typing by Relational Graph Construction and Propagation
  - Refined Entity Typing: PLE and AFET
  - CoType: Joint Typing of Entities and Relations

Mining Phrases: Why Not Use Raw Frequency Based Methods?

- Traditional data-driven approaches
  - Frequent pattern mining
    - If AB is frequent, likely AB could be a phrase
- Raw frequency could NOT reflect the quality of phrases
  - E.g., freq(vector machine) ≥ freq(support vector machine)
  - Need to rectify the frequency based on segmentation results
- Phrasal segmentation will tell
  - Some words should be treated as a whole phrase whereas others are still unigrams

SegPhrase: From Raw Corpus to Quality Phrases and Segmented Corpus

Raw Corpus



Quality Phrases

Segmented Corpus

Document 1
Citation recommendation is an interesting but challenging research problem in data mining area.

Document 2
In this study, we investigate the problem in the context of heterogeneous information networks using data mining technique.

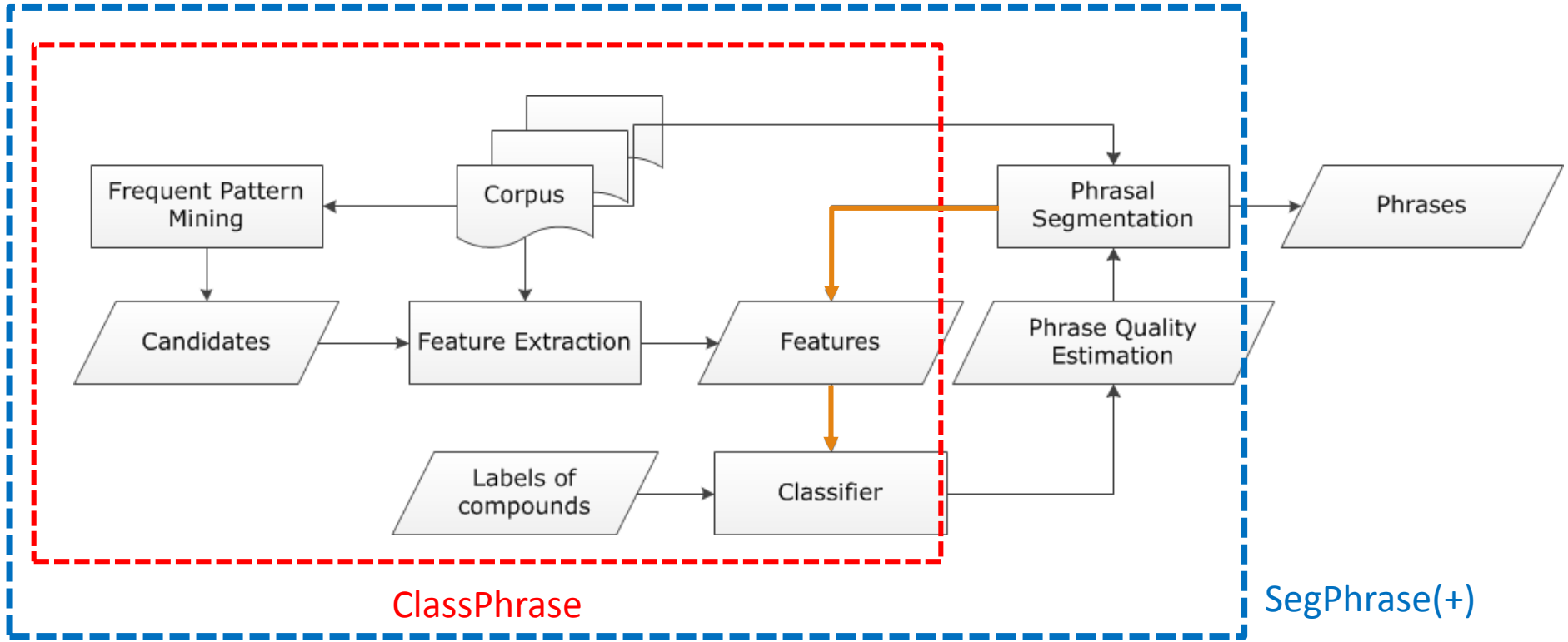
Document 3
Principal Component Analysis is a linear dimensionality reduction technique commonly used in machine learning applications.

Input Raw Corpus → **Quality Phrases** ⟷ **Segmented Corpus**

Phrase Mining **Phrasal Segmentation**

SegPhrase: The Overall Framework

- ClassPhrase: Frequent pattern mining, feature extraction, classification
- SegPhrase: Phrasal segmentation and phrase quality estimation
- SegPhrase+: One more round to enhance mined phrase quality



What Kind of Phrases Are of "High Quality"?

- Judging the quality of phrases
  - **Popularity**
    - "information retrieval" vs. "cross-language information retrieval"
  - **Concordance**
    - "powerful tea" vs. "strong tea"
    - "active learning" vs. "learning classification"
  - **Informativeness**
    - "this paper" (frequent but not discriminative, not informative)
  - **Completeness**
    - "vector machine" vs. "support vector machine"

ClassPhrase I: Pattern Mining for Candidate Set

- Build a candidate phrases set by frequent pattern mining
  - Mining frequent *k*-grams
    - *k* is typically small, e.g. 6 in our experiments
- **Popularity** measured by *raw* frequent words and phrases mined from the corpus

ClassPhrase II: Feature Extraction: Concordance

- Partition a phrase into two parts to check whether the co-occurrence is significantly higher than pure random
  - support vector | machine
    u_l u_r
  - this paper | demonstrates
    u_l u_r

$$\langle u_l, u_r \rangle = \arg \min_{u_l \oplus u_r = v} \log \frac{p(v)}{p(u_l)p(u_r)}$$

- Pointwise mutual information:

$$PMI(u_l, u_r) = \log \frac{p(v)}{p(u_l)p(u_r)}$$

- Pointwise KL divergence:

$$PKL(v \| \langle u_l, u_r \rangle) = p(v) \log \frac{p(v)}{p(u_l)p(u_r)}$$

- The additional $p(v)$ is multiplied with pointwise mutual information, leading to less bias towards rare-occurred phrases

ClassPhrase II: Feature Extraction: Informativeness

- Deriving Informativeness
  - Quality phrases typically start and end with a non-stopword
    - "machine learning is" vs. "machine learning"
  - Use average IDF over words in the phrase to measure the semantics
  - Usually, the probabilities of a quality phrase in quotes, brackets, or connected by dash should be higher (punctuations information)
    - "state-of-the-art"
- We can also incorporate features using some NLP techniques, such as POS tagging, chunking, and semantic parsing

ClassPhrase III: Classifier

- **Limited Training**
  - Labels: Whether a phrase is a quality one or not
    - "support vector machine": 1
    - "the experiment shows": 0
  - For ~1GB corpus, only 300 labels
- Random Forest as our classifier
  - Predicted phrase quality scores lie in [0, 1]
  - Bootstrap many different datasets from limited labels

SegPhrase: Why Do We Need Phrasal Segmentation in Corpus?

- Phrasal segmentation can tell which phrase is more appropriate
  - Ex: A standard [feature vector] [machine learning] setup is used to describe...

Not counted towards the rectified frequency

- Rectified phrase frequency (expected influence)
  - Example:

sequence	frequency	phrase?	rectified
support vector machine	100	yes	80
support vector	160	yes	50
vector machine	150	no	6
support	500	N/A	150
vector	1000	N/A	200
machine	1000	N/A	150

SegPhrase: Segmentation of Phrases

- Partition a sequence of word by maximizing the likelihood
  - Considering
    - Phrase quality score
      - ClassPhrase assigns a **quality score** for each phrase
    - Probability in corpus
    - Length penalty
      - **length penalty** α: when $\alpha > 1$, it favors shorter phrases
- Filter out phrases with low rectified frequency
  - Bad phrases are expected to rarely occur in the segmentation results

SegPhrase+: Enhancing Phrasal Segmentation

- SegPhrase+: One more round for enhanced phrasal segmentation
- **Feedback**
  - Using rectified frequency, re-compute those features previously computing based on raw frequency
- Process
  - Classification → Phrasal segmentation // SegPhrase
    - → Classification → Phrasal segmentation // SegPhrase+
- **Effects** on computing quality scores
  - np hard in the strong sense ↗
  - ~~np hard in the strong~~ ↘
  - data base management system ↗

Performance Study: Methods to Be Compared

- Other phase mining methods: Methods to be compared
  - NLP chunking based methods
    - Chunks as candidates
    - Sorted by **TF-IDF** and **C-value** (K. Frantzi et al., 2000)
  - Unsupervised raw frequency based methods
    - **ConExtr** (A. Parameswaran et al., VLDB 2010)
    - **ToPMine** (A. El-Kishky et al., VLDB 2015)
  - Supervised method
    - **KEA**, designed for single document keyphrases (O. Medelyan & I. H. Witten, 2006)

Performance Study: Experimental Setting

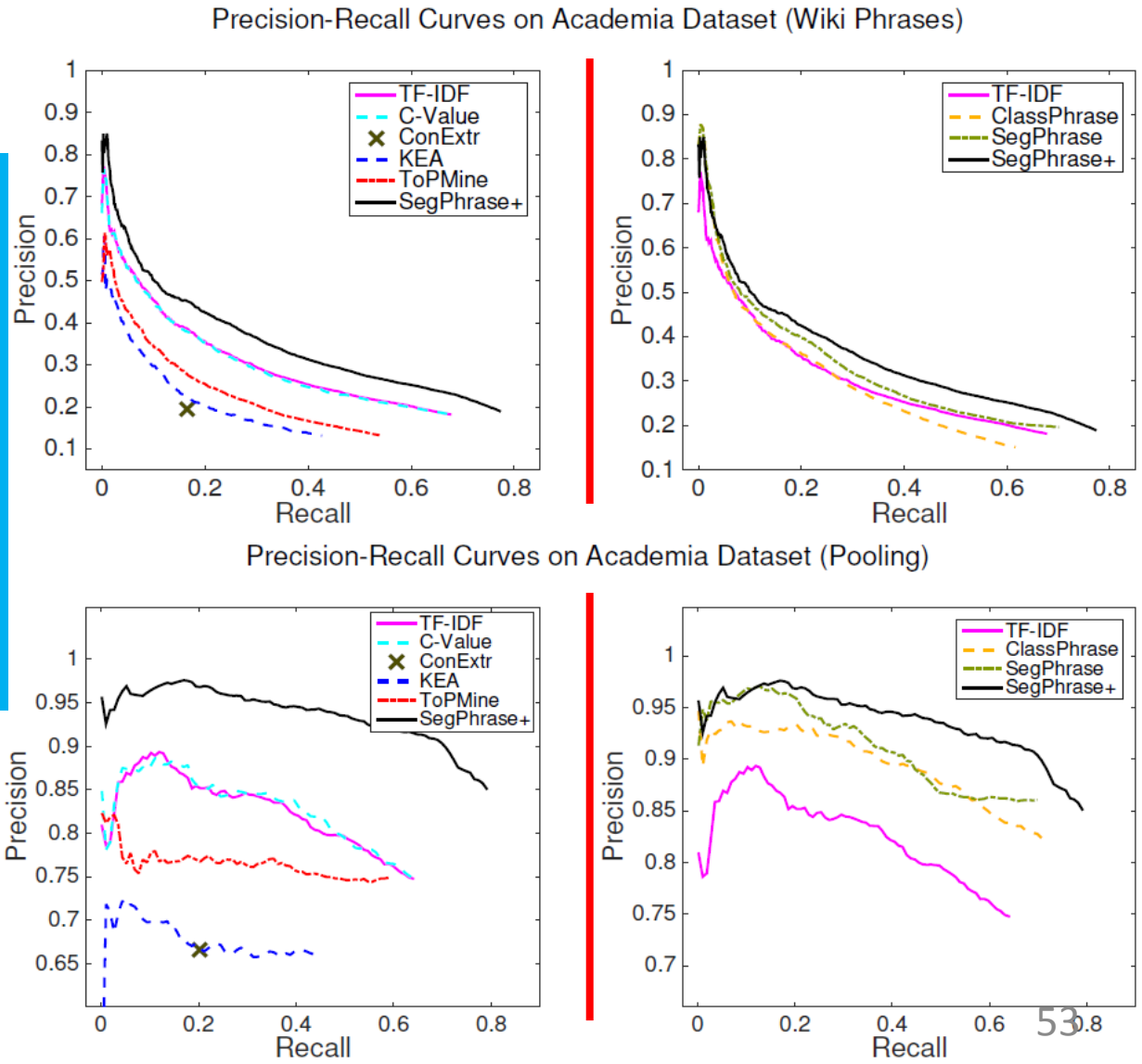
- Datasets

Dataset	#docs	#words	#labels
DBLP	2.77M	91.6M	300
Yelp	4.75M	145.1M	300

- Popular Wiki Phrases
  - Based on internal links
  - ~7K high quality phrases
- Pooling
  - Sampled 500 * 7 **Wiki-uncovered** phrases
  - Evaluated by 3 reviewers independently

Performance: Precision Recall Curves on DBLP

Compare with other baselines
TF-IDF
C-Value
ConExtr
KEA
ToPMine
SegPhrase+

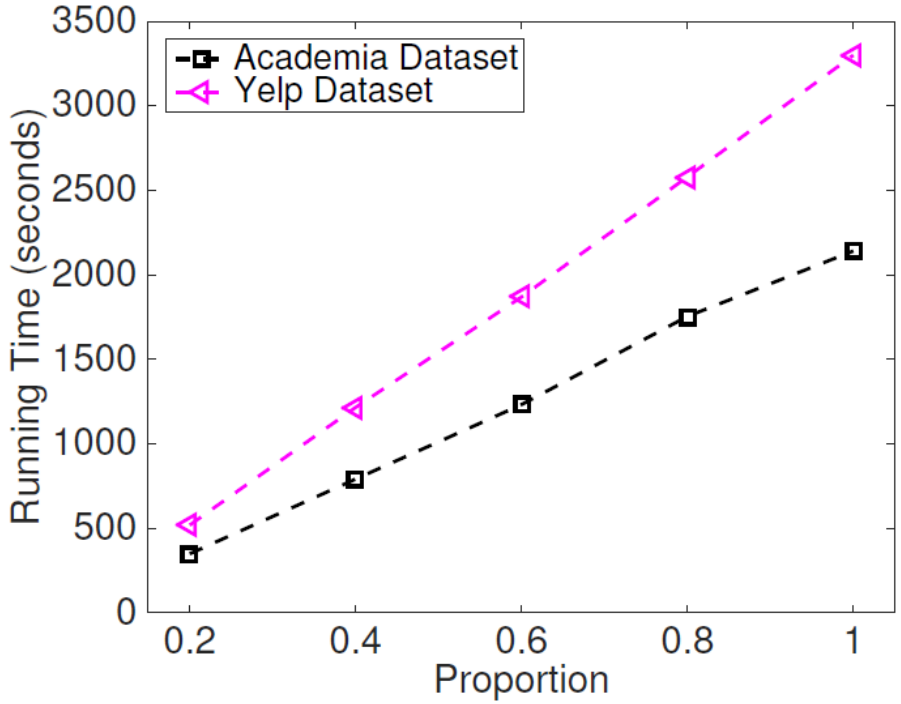


Compare with our 3 variations
TF-IDF
ClassPhrase
SegPhrase
SegPhrase+

53

Performance Study: Processing Efficiency

- SegPhrase+ is linear to the size of corpus!



dataset	file size	#words	time
Academia	613MB	91.6M	0.595h
Yelp	750MB	145.1M	0.917h
Wikipedia	20.23GB	3.26G	28.08h

Experimental Results: Interesting Phrases Generated (From the Titles and Abstracts of SIGMOD)

Query	SIGMOD	
Method	SegPhrase+	Chunking (TF-IDF & C-Value)
1	data base	data base
2	database system	database system
3	relational database	query processing
4	query optimization	query optimization
5	query processing	relational database
…	…	…
51	sql server	database technology
52	**relational data**	database server
53	data structure	**large volume**
54	**join query**	performance study
55	web service	web service
…	…	…
201	**high dimensional data**	**efficient implementation**
202	location based service	sensor network
203	xml schema	**large collection**
204	**two phase locking**	**important issue**
205	deep web	frequent itemset
…	…	…

Only in SegPhrase+

Only in Chunking

Experimental Results: Interesting Phrases Generated (From the Titles and Abstracts of SIGKDD)

Query	SIGKDD	
Method	SegPhrase+	Chunking (TF-IDF & C-Value)
1	data mining	data mining
2	data set	association rule
3	association rule	knowledge discovery
4	knowledge discovery	frequent itemset
5	**time series**	decision tree
…	…	…
51	association rule mining	search space
52	rule set	domain knowledge
53	concept drift	**important problem**
54	knowledge acquisition	concurrency control
55	**gene expression data**	conceptual graph
…	…	…
201	web content	optimal solution
202	**frequent subgraph**	semantic relationship
203	intrusion detection	**effective way**
204	**categorical attribute**	space complexity
205	user preference	**small set**
…	…	…

Only in SegPhrase+

Only in Chunking

Experimental Results: Similarity Search

- Find high-quality similar phrases based on user's phrase query
  - In response to a user's phrase query, SegPhrase+ generates high quality, semantically similar phrases
  - In DBLP, query on "data mining" and "OLAP"
  - In Yelp, query on "blu-ray", "noodle", and "valet parking"

Query	data mining		olap	
Method	SegPhrase+	Chunking	SegPhrase+	Chunking
1	knowledge discovery	driven methodologies	data warehouse	warehouses
2	text mining	text mining	online analytical processing	clustcube
3	web mining	financial investment	data cube	rolap
4	machine learning	knowledge discovery	olap queries	online analytical processing
5	data mining techniques	building knowledge	multidimensional databases	analytical processing

Query	blu-ray		noodle		valet parking	
Method	SegPhrase+	Chunking	SegPhrase+	Chunking	SegPhrase+	Chunking
1	dvd	new microwave	ramen	noodle soup	valet	huge lot
2	vhs	lifetime warranty	noodle soup	asian noodle	self-parking	private lot
3	cd	recliner	rice noodle	beef noodle	valet service	self-parking
4	new release	battery	egg noodle	stir fry	free valet parking	valet
5	sony	new battery	pasta	fish ball	covered parking	front lot

Mining Quality Phrases in Multiple Languages

- Both ToPMine and SegPhrase+ are extensible to mining quality phrases in multiple languages
- SegPhrase+ on Chinese (From Chinese Wikipedia)
- ToPMine on Arabic (From Quran (Fus7a Arabic)(no preprocessing)
  - Experimental results of Arabic phrases:

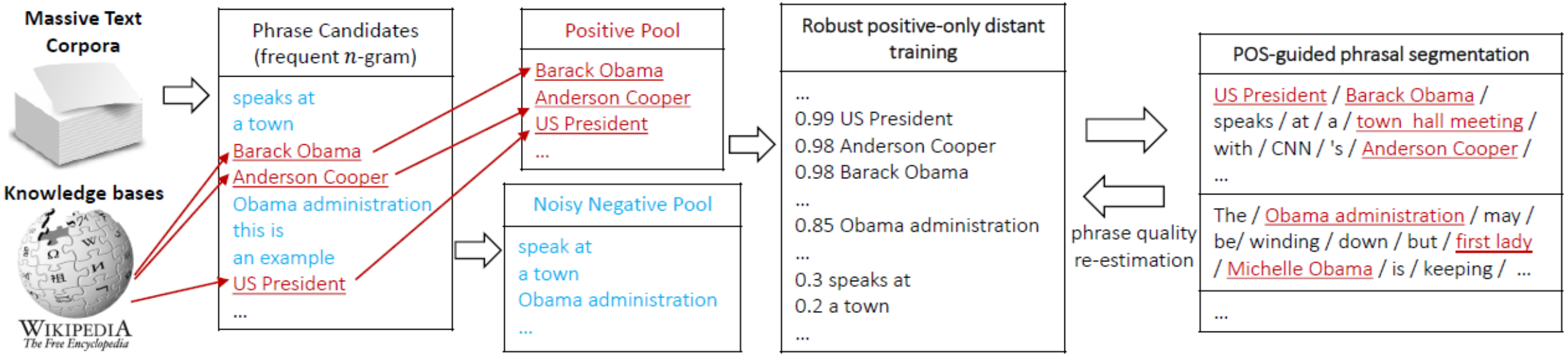
كفروا→ Those who disbelieve

بسم الله الرحمن الرحيم→ In the name of God the Gracious and Merciful

Rank	Phrase	In English
…	…	…
62	首席_执行官	CEO
63	中间_偏右	Middle-right
…	…	…
84	百度_百科	Baidu Pedia
85	热带_气旋	Tropical cyclone
86	中国科学院_院士	Fellow of Chinese Academy of Sciences
…	…	…
1001	十大_中文_金曲	Top-10 Chinese Songs
1002	全球_资讯网	Global News Website
1003	天一阁_藏_明代_科举_录_选刊	A Chinese book name
…	…	…
9934	国家_戏剧_院	National Theater
9935	谢谢_你	Thank you
…	…	…

AutoPhrase: Automated Phrase Mining

- Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R. Voss, Jiawei Han, "**AutoPhrase:** Automated Phrase Mining from Massive Text Corpora" submitted for pub. 2017
- Automatic extraction of high-quality phrases (e.g., scientific terms and general entity names) in a given corpus (e.g., research papers and news)
  - No human efforts
  - Multiple languages
  - High performance—precision, recall, efficiency



AutoPhrase: Label Generation by Distant Supervision

- Completely remove the human effort for labeling phrases
- **Distant training**: Utilize high-quality phrases in KBs (e.g., Wiki) as positive phrase labels
- **Method: Sampling-based label generation**
  - **Positive Labels**
    - Wikipedia contains many high-quality phrases in titles, keywords, and internal links
      - E.g., in Chinese, more than 20,000
    - Uniformly draw 100 samples as positive labels for single-word and multi-word phrases respectively
  - **Negative Labels**
    - Phrase candidates meeting the popularity requirement is huge in size and the majority of them are actually poor in quality (e.g., “francisco opera and”).
      - Ex. A small corpus in Chinese has about **4 million frequent phrase candidates**, while **more than 90%** are not in good quality

Generating High-Quality Phrases in Multi-Languages

- Complicated pre-processing models, such as dependency parsing, heavily rely on human efforts and thus cannot be smoothly applied to multiple languages
- Minimum Language Dependency = **Tokenization + POS tagging**
- Drawbacks of Frequency-based signals only: Over-decomposition & Under-decomposition
  - "Sophia Smith" vs. "Sophia" and "Smith"
  - "Great Firewall" vs. "firewall software"

#1:	⌈Sophia	Smith⌋	was	born	in	England	.
	NNP	NNP	VBD	VBN	IN	NNP	.
#2:	...	the	⌈Great	Firewall⌋	is	...	
	...	DT	NNP	NNP	VBZ	...	
#3:	This	is	a	great	⌈firewall	software⌋	.
	DT	VBZ	DT	JJ	NN	NN	.

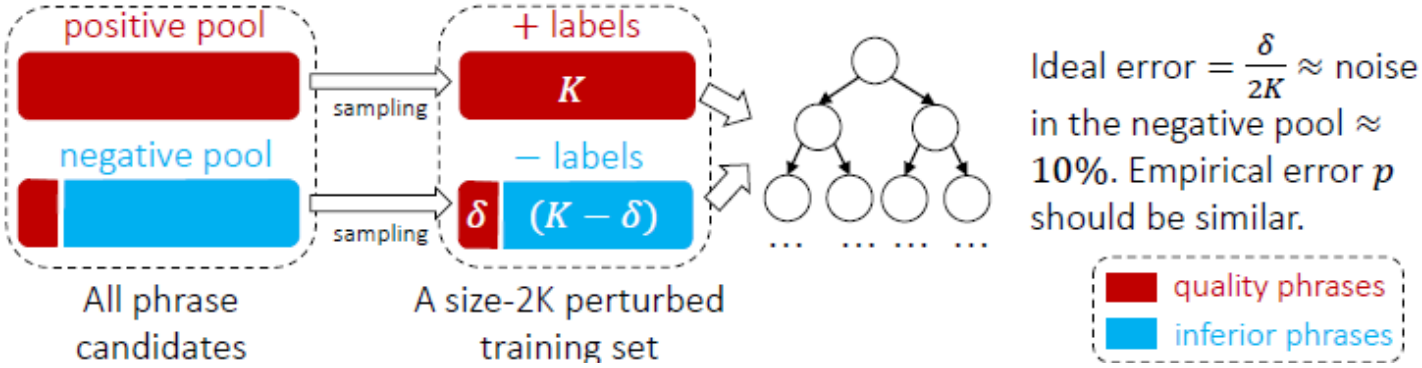
- Drawbacks of POS only:
  - "classifier SVM" vs. "discriminative classifier" and "SVM"

#4:	The	⌈discriminative	classifier⌋	⌈SVM⌋	is	...
	DT	JJ	NN	NN	VBZ	...

- Context-aware phrasal segmentation

Robust Positive-Only Distant Training

- In each base classifier, randomly sample K positive (e.g., wiki titles, keywords, links) and K noisy negative labels from the pools
- Noisy negative pool: δ quality phrases among the K negative labels



- Perturbed training set: size-2K subset of the full set of all phrase where the labels of some quality phrases are switched from positive to negative
- For each base classifier, we randomly draw K phrase candidates with replacement from the positive pool and the negative pool respectively
- We grow an unpruned decision tree to the point of separating all phrases to meet this requirement
- Use an ensemble classifier that averages the results of T independently trained base classifiers

Single-Word Modeling: Enhancing Recall

- AutoPhrase: Simultaneously model single-word and multi-word phrases
- A phrase is not only a group of multiple words, but also possibly a single word, as long as it functions as a constituent in the syntax of a sentence, e.g., “UIUC”, “Illinois”
  - Based on our experiments: 10%~30% quality phrases are single-word phrases
- Modeling single-word phrases: Examining requirements of quality multi-word phrase
  - Popularity: sufficient frequent in the given corpus
  - Concordance: the collocation of tokens in such frequency that is significantly higher than random
  - Informativeness: indicative of a specific topic or concept
  - Completeness: Complete semantic unit
- Only **concordance** cannot be defined in single-word phrases
- **Independence**: A quality single-word phrase is more likely a complete semantic unit in the given documents

Experiments and Performance Comparison

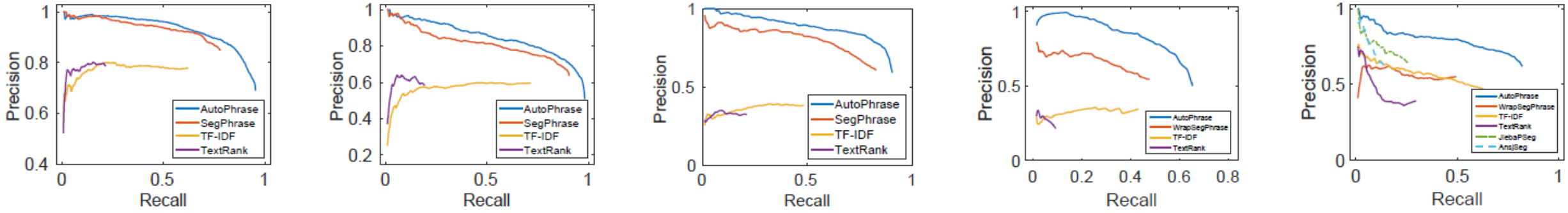
- Datasets:

Dataset	Domain	Language	$\|\Omega\|$	File size	$size_p$
DBLP	Scientific Paper	English	91.6M	618MB	29K
Yelp	Business Review	English	145.1M	749MB	22K
EN	Wikipedia Article	English	808.0M	3.94GB	184K
ES	Wikipedia Article	Spanish	791.2M	4.06GB	65K
CN	Wikipedia Article	Chinese	371.9M	1.56GB	29K

Phrase Mining Results

	EN	*CN*	
Rank	Phrase	Phrase	Translation (Explanation)
1	Elf Aquitaine	江苏 舜 天	(the name of a soccer team)
2	Arnold Sommerfeld	苦 艾 酒	Absinthe
3	Eugene Wigner	白发 魔 女	(the name of a novel/TV-series)
4	Tarpon Springs	笔记 型 电脑	notebook computer, laptop
5	Sean Astin	党委 书记	Secretary of Party Committee
...	...	...	...
20,001	ECAC Hockey	非洲 国家	African countries
20,002	Sacramento Bee	左翼 党	The Left (German: Die Linke)
20,003	Bering Strait	菲 沙 河谷	Fraser Valley
20,004	Jacknife Lee	海马 体	Hippocampus
20,005	WXYZ-TV	斋 贺光希	Mitsuki Saiga (a voice actress)
...	...	...	...
99,994	John Gregson	计算机 科学技术	Computer Science and Technology
99,995	white-tailed eagle	恒 天然	Fonterra (a company)
99,996	rhombic dodecahedron	中国 作家 协会 副 主席	The Vice President of Writers Association of China
99,997	great spotted woodpecker	维他命 b	Vitamin B
99,998	David Manners	舆论 导向	controlled guidance of the media
...	...	...	...

- Comparing methods
  - SegPhrase/WrapSegPhrae (encoding preprocessing for handling non-English)
  - TF-IDF/TextRank



(a) DBLP (b) Yelp (c) EN (d) ES (e) CN

References on Phrase Mining

- D. M. Blei and J. D. Lafferty. Visualizing Topics with Multi-Word Expressions, arXiv:0907.1013, 2009
- M. Danilevsky, C. Wang, N. Desai, X. Ren, J. Guo, J. Han. Automatic Construction and Ranking of Topical Keyphrases on Collections of Short Documents“, SDM’14
- A. El-Kishky, Y. Song, C. Wang, C. R. Voss, and J. Han. Scalable Topical Phrase Mining from Text Corpora. *VLDB*’15
- K. Frantzi, S. Ananiadou, and H. Mima, Automatic Recognition of Multi-Word Terms: the c-value/nc-value Method. Int. Journal on Digital Libraries, 3(2), 2000
- R. V. Lindsey, W. P. Headden, III, M. J. Stipicevic. A phrase-discovering topic model using hierarchical pitman-yor processes, EMNLP-CoNLL’12.
- J. Liu, J. Shang, C. Wang, X. Ren, J. Han, Mining Quality Phrases from Massive Text Corpora. *SIGMOD’*15
- Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R. Voss, Jiawei Han, “**AutoPhrase:** Automated Phrase Mining from Massive Text Corpora” submitted for pub. 2017
- O. Medelyan and I. H. Witten, Thesaurus Based Automatic Keyphrase Indexing. IJCDL’06
- Q. Mei, X. Shen, C. Zhai. Automatic Labeling of Multinomial Topic Models, KDD’07
- A. Parameswaran, H. Garcia-Molina, and A. Rajaraman. Towards the Web of Concepts: Extracting Concepts from Large Datasets. *VLDB*’10
- X. Wang, A. McCallum, X. Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval, ICDM’07

Outline

- Why Text to Networks?
- Phrase Mining and Topic Modeling from Large Text Corpora
  - Why Phrase Mining
  - Previous Approaches
  - TopMine
  - SegPhrase and AutoPhrase
- Entity Recognition and Typing for Massive Text Data
  - ClusType: Entity Extraction and Typing by Relational Graph Construction and Propagation
  - Refined Entity Typing: PLE and AFET
  - CoType: Joint Typing of Entities and Relations

Why Entity Recognition and Typing from Massive Corpora?

- Traditional named entity recognition systems are designed for major types (e.g., PER, LOC, ORG) and general domains (e.g., news)
  - Require additional steps to adapt to new domains/types
  - Expensive human labor on annotation
    - 500 documents for entity extraction; 20,000 queries for entity linking
  - Unsatisfying agreement due to various granularity levels and scopes of types
- Entities obtained by entity linking techniques have *limited* coverage and freshness
  - > 50% unlinkable entity mentions in Web corpus [Lin et al., EMNLP'12]
  - > 90% in our experiment corpora: tweets, Yelp reviews, …
- A new approach: ClusType: Entity Recognition and Typing by Relation Phrase-Based Clustering [Ren, et al., KDD 2015]
  - Recognizing entity mentions of target types with minimal/no human supervision and with no requirement that entities can be found in a KB (distant supervision)

Recognizing Typed Entities

Identifying token span as entity mentions in documents and labeling their types

Target Types

- FOOD
- LOCATION
- JOB_TITLE
- EVENT
- ORGANIZATION
- …

The best BBQ I've tasted in Phoenix! I had the pulled pork sandwich with coleslaw and baked beans for lunch. ... The owner is very nice. …

The best BBQ:Food I've tasted in Phoenix:LOC ! I had the [pulled pork sandwich]:Food with coleslaw:Food and [baked beans]:Food for lunch. … The owner:JOB_TITLE is very nice. …

Plain text

Text with typed entities

FOOD

LOCATION

EVENT

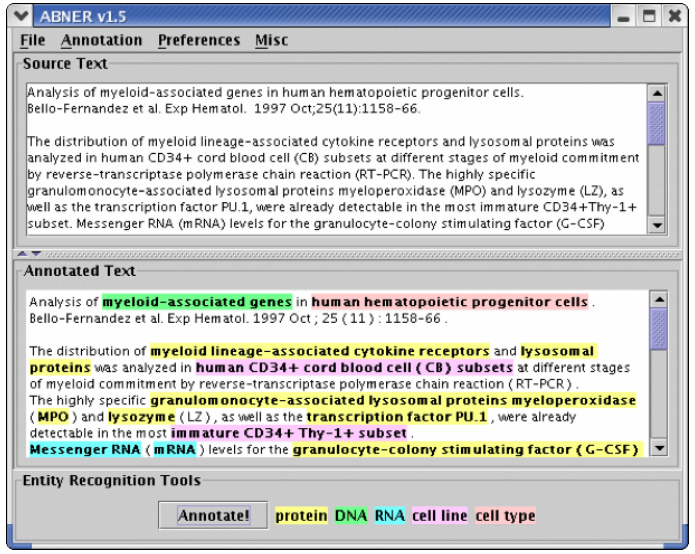
Enabling structured analysis of unstructured text corpus

Traditional NLP Approach: Feature Engineering

- Typical Entity Extraction Features (Li et al., 2012)
  - N-gram: Unigram, bigram and trigram token sequences in the context window
  - Part-of-Speech: POS tags of the context words
  - Gazetteers: person names, organizations, countries and cities, titles, idioms, etc.
  - Word clusters: word clusters / embeddings
  - Case and Shape: Capitalization and morphology analysis based features
  - Chunking: NP and VP Chunking tags
  - Global feature: Sentence level and document level structure/position features

Traditional Named Entity Recognition (NER) Systems

- Heavy reliance on human annotated data
- Training sequence models is slow



A manual annotation interface

The best [BBQ] I've tasted in [Phoenix].

The	best	BBQ	I've	tasted	in	Phoenix
O	O	Food	O	O	O	Location

Sequence model training

Systems:
Stanford NER
Illinois Name Tagger
IBM Alchemy APIs
…

Finkel et al., *Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling*, ACL 2005

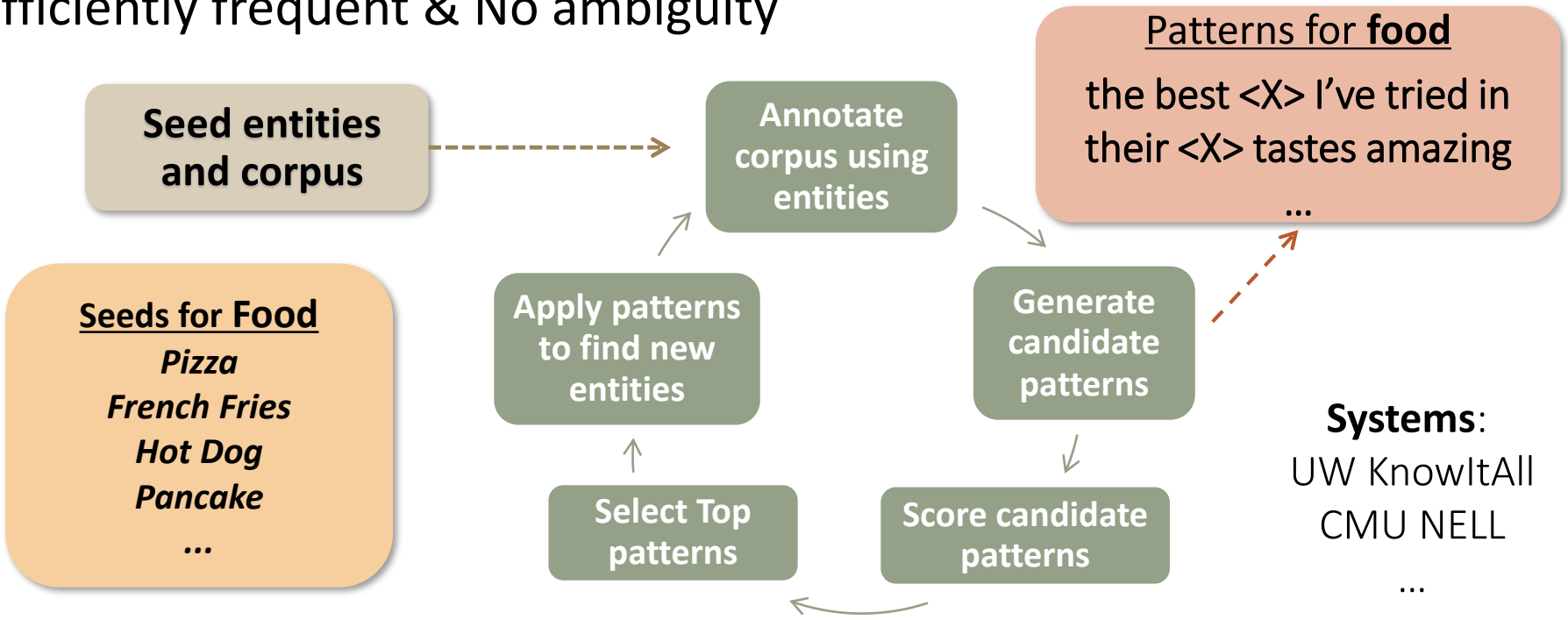
Traditional NLP Approach: Feature Engineering

- Typical Entity Linking Features (Ji et al., 2011)

Mention/Concept Attribute		Description
Name	Spelling match	Exact string match, acronym match, alias match, string matching…
	KB link mining	Name pairs mined from KB text redirect and disambiguation pages
	Name Gazetteer	Organization and geo-political entity abbreviation gazetteers
Document surface	Lexical	Words in KB facts, KB text, mention name, mention text.
		Tf.idf of words and ngrams
	Position	Mention name appears early in KB text
	Genre	Genre of the mention text (newswire, blog, …)
	Local Context	Lexical and part-of-speech tags of context words
Entity Context	Type	Mention concept type, subtype
	Relation/Event	Concepts co-occurred, attributes/relations/events with mention
	Coreference	Co-reference links between the source document and the KB text
Profiling		Slot fills of the mention, concept attributes stored in KB infobox
Concept		Ontology extracted from KB text
Topic		Topics (identity and lexical similarity) for the mention text and KB text
KB Link Mining		Attributes extracted from hyperlink graphs of the KB text
Popularity	Web	Top KB text ranked by search engine and its length
	Frequency	Frequency in KB texts

Entity Extraction with Minimal Human Supervision (I) Weak Supervision

- Weak supervision: relies on *manually selected seed entities* in applying pattern-based bootstrapping methods or label propagation methods to identify more entities
- Pattern-Based Bootstrapping: Requires manual seed selection & mid-point checking
  - Sufficiently frequent & No ambiguity



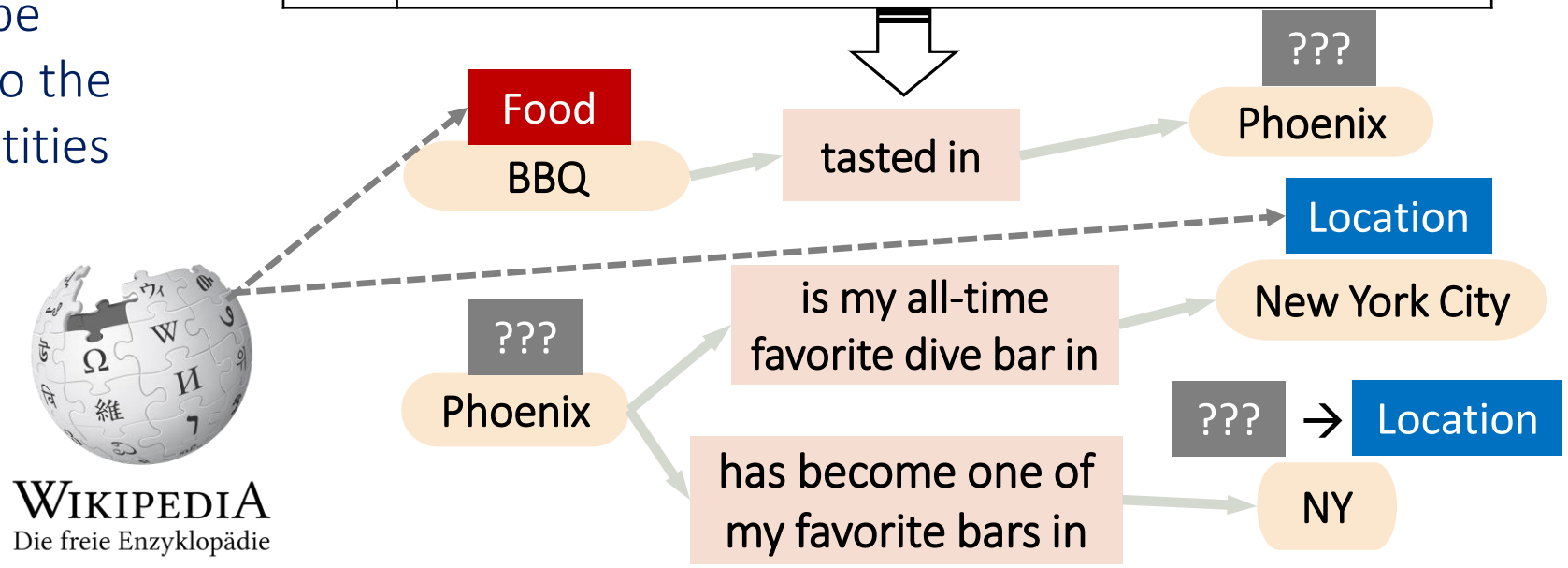
Etzioni et al., *Unsupervised named-entity extraction from the web: An experimental study*, Artificial Intelligence 2005. Mitchell et al. Never-ending Learning, AAAI, 2015

Entity Extraction with Minimal Human Supervision (II)
Distant Supervision

- Distant supervision: leverages entity information in KBs to reduce human supervision

1. Detect entity names from text
2. Link entity names to KB entities
3. Propagate type information to the unlinkable entities

ID	Sentence
S1	*Phoenix* is my all-time favorite dive bar in *New York City* .
S2	The best *BBQ* I've tasted in *Phoenix*.
S3	*Phoenix* has become one of my favorite bars in *NY* .



Lin et al. *No noun phrase left behind: detecting and typing unlinkable entities*. EMNLP, 2012

Previous Methods: Limitation I: Domain Restriction

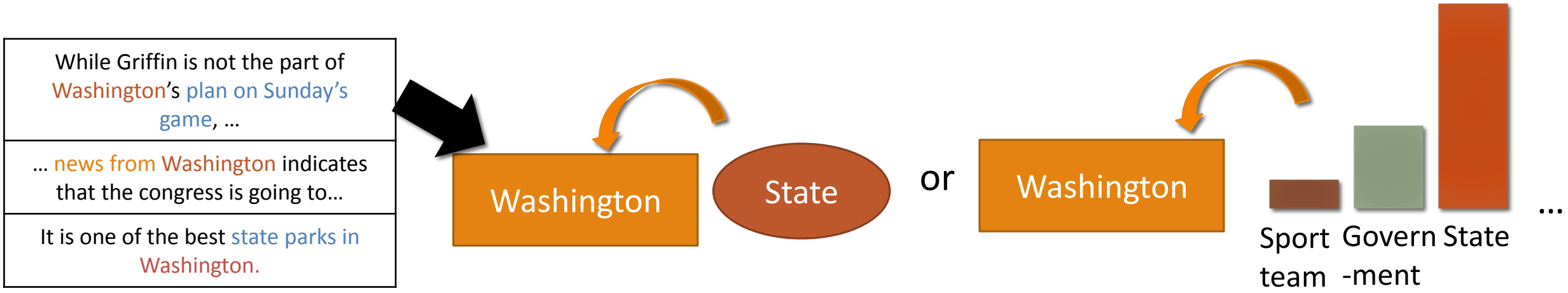
- Most existing work assume entity mentions are already extracted by existing entity detection tools, *e.g.*, noun phrase chunkers
  - Usually trained on general-domain corpora like news articles (clean, grammatical)
  - Make use of various linguistics features (e.g., semantic parsing structures)
  - Do not work well on specific, dynamic or emerging domains (*e.g.*, tweets, Yelp reviews)
    - E.g, “in-and-out” from Yelp review may not be properly detected

Previous Methods: Limitation II: Name Ambiguity

- Multiple entities may share the same surface name

While Griffin is not the part of Washington's plan on Sunday's game, …	Sport team
…has concern that Kabul is an ally of Washington.	U.S. government
He has office in Washington, Boston and San Francisco	U.S. capital city

- Previous methods simply output a single type/type distribution for each surface name, instead of an exact type for each entity mention



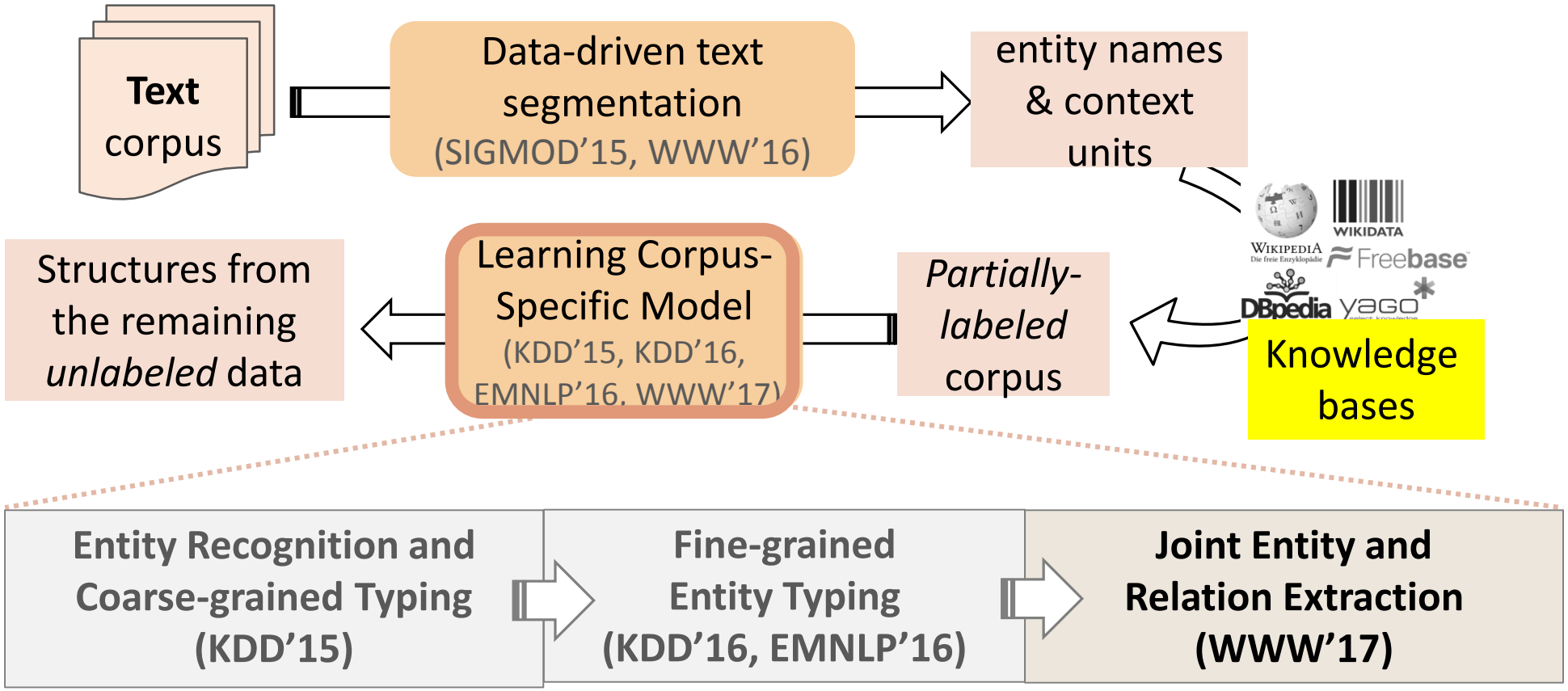
Previous Methods: Limitation III: Context Sparsity

- A variety of contextual clues are leveraged to find sources of shared semantics across different entities
  - Keywords, Wiki concepts, linguistic patterns, textual relations, …
- There are often many ways to describe even the same relation between two entities

ID	Sentence	Freq
1	The magnitude 9.0 quake **caused widespread devastation in [Kesennuma city]**	12
2	… tsunami that **ravaged [northeastern Japan]** last Friday	31
3	The resulting tsunami **devastate [Japan]**'s northeast	244

- Previous methods have difficulties in handling entity mention with sparse (infrequent) context

Corpus to Structured Text: Our Roadmap



The ClusType Solution

Domain-agnostic phrase mining algorithm: Extracts candidate entity mentions with minimal linguistic assumption → address domain restriction

- E.g., part-of-speech (POS) tagging << semantic parsing

Do not simply merge entity mentions with *identical surface names*

- Model each mention based on its surface name and context, in a scalable way

  → address name ambiguity

Mine *relation phrase* co-occurring with entity mentions; infer synonymous relation phrases

- Helps form connecting bridges among entities that do not share identical context, but share synonymous relation phrases → **address context sparsity**

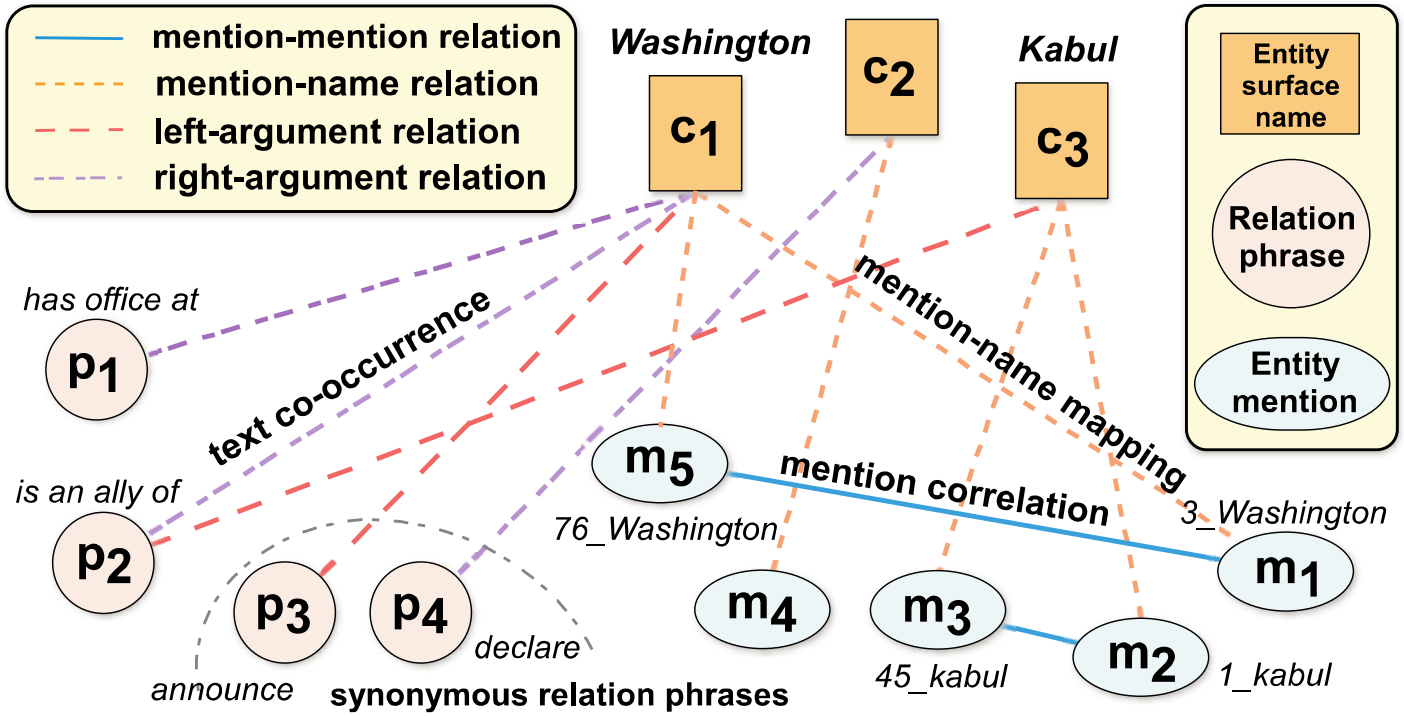
The ClusType Framework: Phrase Segmentation and Heterogeneous Graph Construction

- POS-constrained phrase segmentation for mining candidate entity mentions and relation phrases, simultaneously
- Construct a heterogeneous graph to represent available information in a unified form

Entity mentions are kept as individual objects **to be disambiguated**

Linked to entity surface names & relation phrases

Weight assignment: The more two objects are likely to share the same label, the larger the weight will be associated with their connecting edge

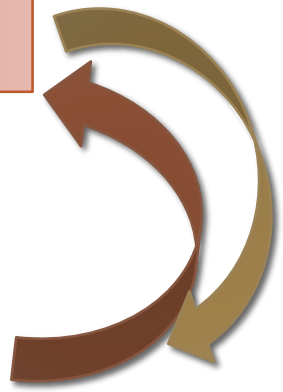


The ClusType Framework: Mutual Enhancement of Type Propagation and Relation Phrase Clustering

- With the constructed graph, formulate a graph-based semi-supervised learning of two tasks jointly:

Type propagation on heterogeneous graph

Derived entity argument types serve as **good feature** for clustering relation phrases



Multi-view relation phrase clustering

Propagate type information among entities bridges via synonymous relation phrases

Mutually enhancing each other; leads to quality recognition of unlinkable entity mentions

ClusType: A General Framework Overview

- **Candidate Generation**
  - Perform phrase mining on a POS-tagged corpus to extract candidate entity mentions and relation phrases
- **Construction of Heterogeneous Graphs**
  - Construct a heterogeneous graph to encode our insights on modeling the type for each entity mention
  - Collect seed entity mentions as labels by linking extracted mentions to the KB
- **Relation Phrase Clustering**
  - Estimate type indicator for unlinkable candidate mentions with the proposed type propagation integrated with relation phrase clustering on the constructed graph

Step 1: Candidate Generation

- An efficient phrase mining algorithm incorporating both *corpus-level statistics* and *syntactic constraints*
  - **Global significance score**: Filter low-quality candidates; **generic POS tag patterns**: remove phrases with improper syntactic structure
  - By extending TopMine, the algorithm partitions corpus into segments which meet both significance threshold and POS patterns → candidate entity mentions & relation phrases

Algorithm workflow:

1. Mine frequent contiguous patterns
2. Performs greedy-agglomerative merging while enforcing our syntactic constraints
   - Entity mention: consecutive nouns
   - Relation phrases: shown in the table
3. Terminates when the next highest-score merging does not meet a pre-defined significance threshold

Relation phrase: Phrase that denotes a unary or binary relation in a sentence

Pattern	Example
V	disperse; hit; struck; knock;
P	in; at; of; from; to;
V P	locate in; come from; talk to;
VW*(P)	caused major damage on; come lately

V-verb; P-prep; W-{adv | adj | noun | det | pron}
W* denotes multiple W; (P) denotes optional.

Candidate Generation: Example and Performance

- Example output of candidate generation on NYT news articles

> Over:RP the weekend the system:EP dropped:RP nearly inches of snow in:RP western Oklahoma:EP and at:RP [Dallas Fort Worth International Airport]:EP sleet and ice caused:RP hundreds of [flight cancellations]:EP and delays. ..... It is forecast:RP to reach:RP [northern Georgia]:EP by:RP [Tuesday afternoon]:EP, Washington:EP and [New York]:EP by:RP [Wednesday afternoon]:EP, meteorologists:EP said:RP.

`EP: entity mention candidate; RP: relation phrase`

- Entity detection performance comparison with an NP chunker

Method	NYT		Yelp		Tweet	
	Prec	Recall	Prec	Recall	Prec	Recall
Our method	**0.469**	**0.956**	**0.306**	**0.849**	0.226	**0.751**
NP chunker	0.220	0.609	0.296	0.247	**0.287**	0.181

Recall is most critical for this step since later we cannot detect the misses (i.e., false negatives)

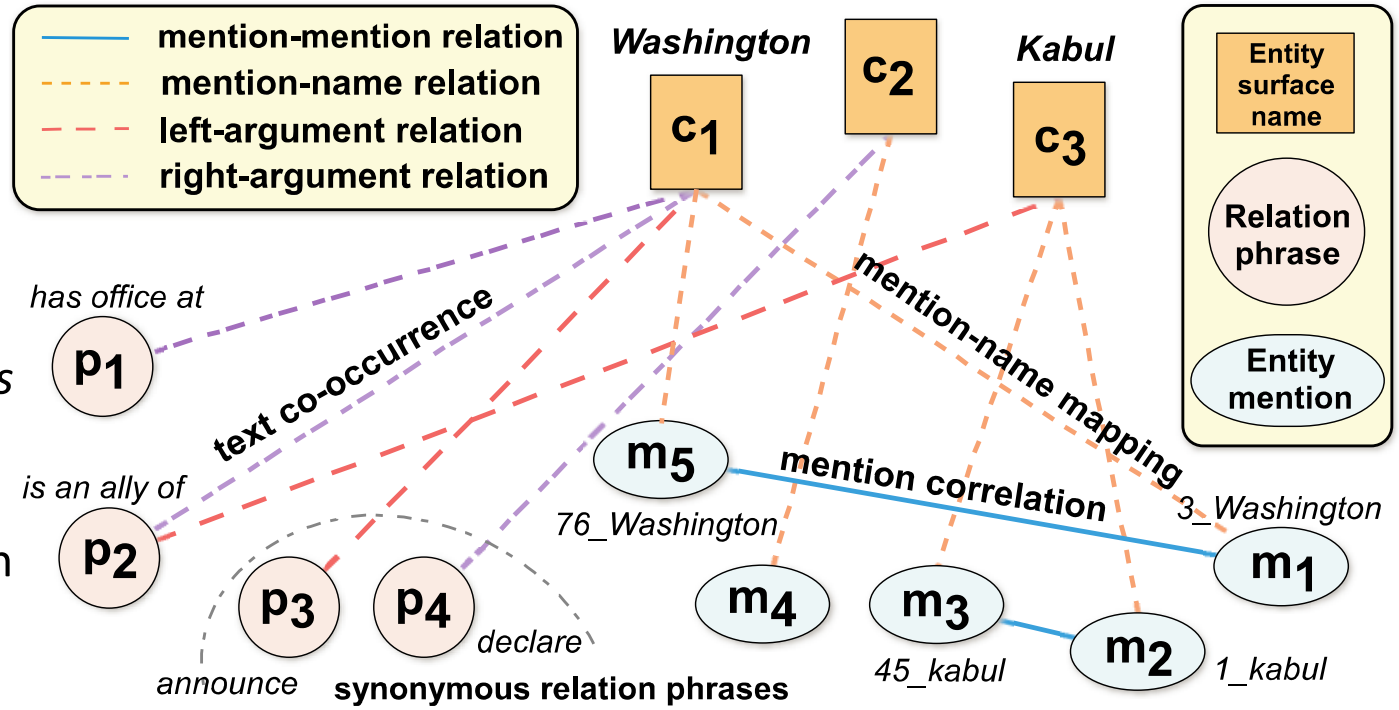
Step 2: Construction of Heterogeneous Graphs

- With three types of objects extracted from corpus: candidate entity mentions, entity surface names, and relation phrases
  - We can construct a heterogeneous graph to **enforce several hypotheses for modeling type of each entity mention** (introduced in the following slides)

Basic idea for constructing the graph:
the more two objects are likely to share the same label, the larger the weight will be associated with their connecting edge

Three types of links:

1. **Mention-name link**: *(many-to-one) mappings* between entity mention and surface names
2. **Name-relation phrase links**: *corpus-level co-occurrence* between surface names and relation phrases
3. **Mention correlation links**: *distributional similarity* between entity mentions



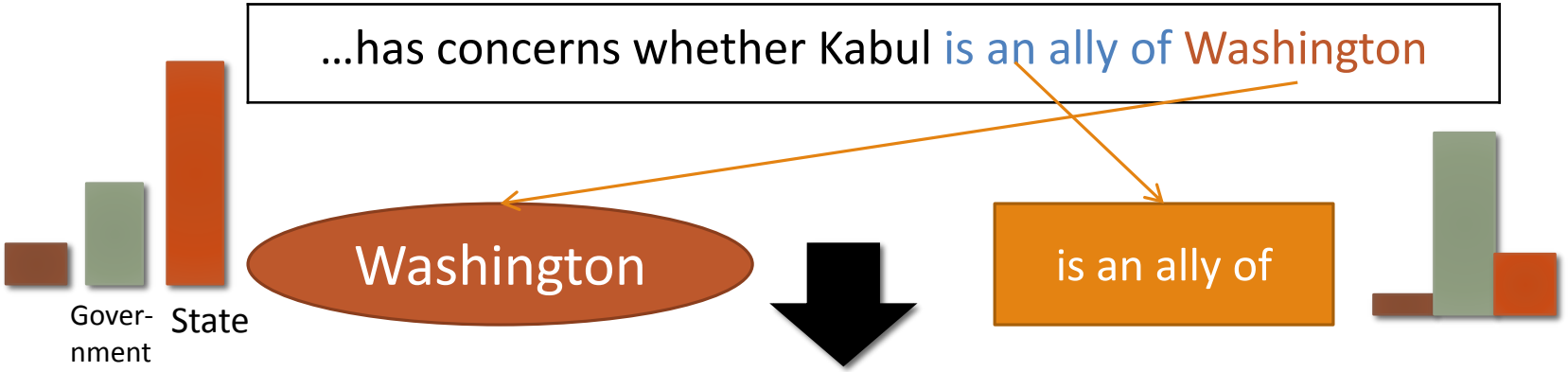
Entity Mention-Surface Name Subgraph

- Directly modeling type indicator of each entity mention in label propagation

  → Intractable size of parameter space

- Both the entity name and the surrounding relation phrases provide strong cues on the types of a candidate entity mention

  → Model the type of each entity mention by (1) type indicator of its surface name; (2) the type signatures of its surrounding relation phrases (more details in the following slides)



M candidate mentions;
n surface names

Use a bi-adjacency matrix to represent the mapping

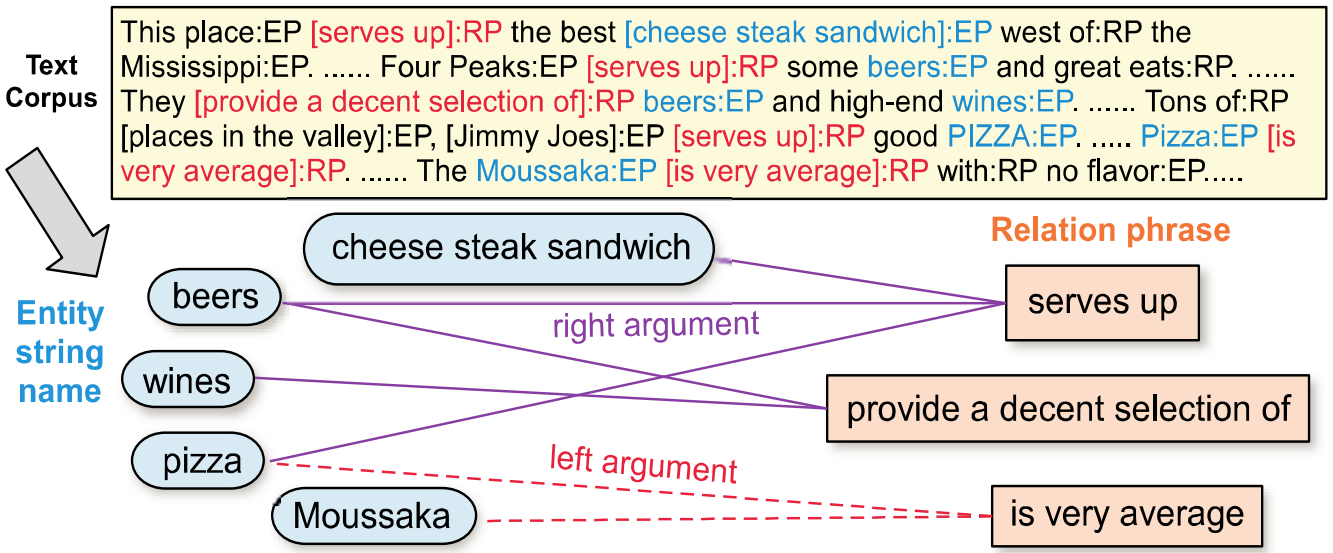
$$\Pi_{\mathcal{C}} \in \{0,1\}^{M \times n}$$

Entity Name-Relation Phrase Subgraph

- Aggregated co-occurrences between entity surface names and relation phrases across corpus → weight importance of different relation phrases for surface names

  → use connected edges as bridges to propagate type information

  - **Left/right entity argument of relation phrase**: for each mention, assign it as the left (right) argument to the closest relation phrase on its right (left) in a sentence
  - **Type signature of relation phrase**: Two type indicators for its left and right arguments



Hypothesis 1 (Entity-Relation Co-occurrences): *If surface name c often appears as the left (right) argument of relation phrase p, then c's type indicator tends to be similar to the corresponding type indicator in p's type signature.*

l different relation phrases, mapping between mentions and relation phrases: $\Pi_L, \Pi_R \in \{0,1\}^{M \times l}$

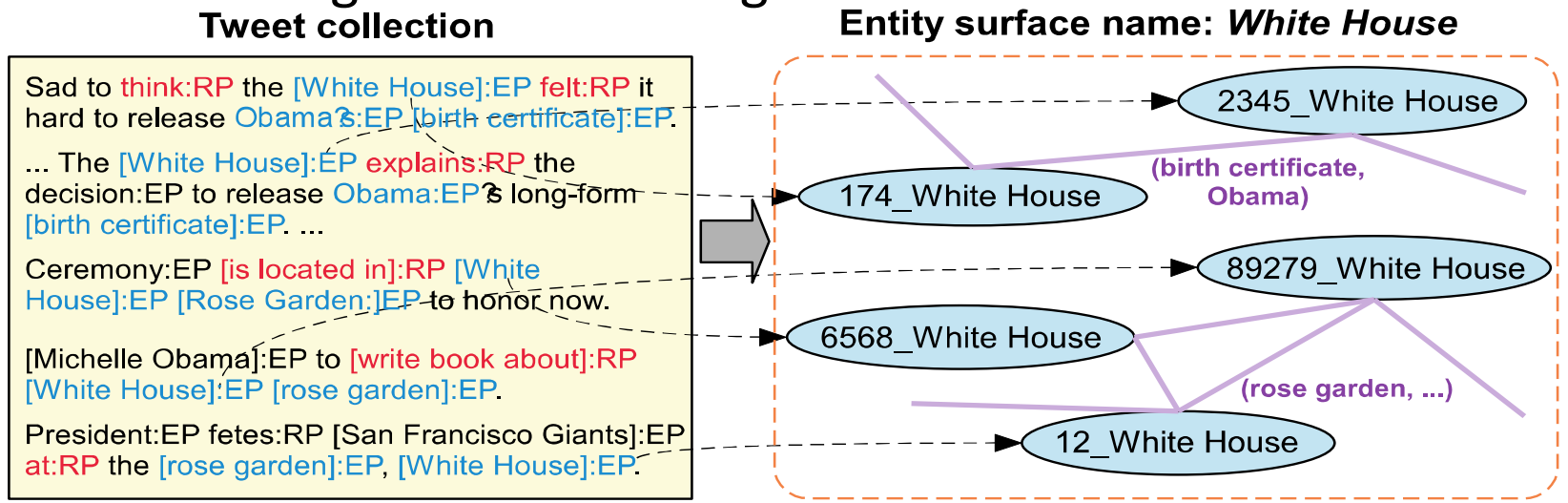
Two bi-adjacency matrices for the subgraph

$$\mathbf{W}_L = \Pi_C^T \Pi_L \quad \text{and} \quad \mathbf{W}_R = \Pi_C^T \Pi_R;$$

Mention Correlation Subgraph

- An entity mention may have ambiguous name and ambiguous relation phrases
  - E.g., “White house” and “felt” in the first sentence of Figure
- Other co-occurring mentions may provide good hints to the type of an entity mention
  - E.g., “birth certificate” and “rose garden” in the Figure

→ Propagate type information between candidate mentions of *each* surface name, based on following hypothesis:



Hypothesis 2 (Mention correlation):

If there exists a strong correlation (i.e., within sentence, common neighbor mentions) between two candidate mentions that share the same name, then their type indicators tend to be similar.

Construct **KNN graph** based on the feature vector **f**-surface names of co-occurring entity mentions $\mathbf{W}_{\mathcal{M}} \in \mathbb{R}^{M \times M}$

$$W_{\mathcal{M},ij} = \begin{cases} \text{sim}(\mathbf{f}^{(i)}, \mathbf{f}^{(j)}), & \text{if } \mathbf{f}^{(i)} \in N_k(\mathbf{f}^{(j)}) \text{ or } \mathbf{f}^{(j)} \in N_k(\mathbf{f}^{(i)}) \\ & \text{and } c(m_i) = c(m_j); \\ 0, & \text{otherwise.} \end{cases}$$

Step 3: Relation Phrase Clustering

- The type signatures of frequent relation phrases can help infer the type signatures of infrequent (sparse) ones that have similar cluster memberships
- Existing work on relation phrase clustering utilizes strings; context words; entity argument to cluster synonymous relation phrases
  - String similarity and distribution similarity may be insufficient to resolve two relation phrases; type information is particular helpful in such case
- We propose to leverage type signature of relation phrase, and propose a general relation phrase clustering method to incorporate different features
  - Two relation phrases tend to have similar cluster memberships, if they have similar (1) strings; (2) context words; and (3) left and right argument type indicators
- Relation phrase clustering is further integrated with the graph-based type propagation in a mutually enhancing framework, based on following hypothesis

RP Clustering: Type Signature Consistency

- Observation: many relation phrases have very few occurrences in the corpus
  - ~37% relation phrases have < 3 unique entity surface names (in right or left arguments)

  → Hard to model their type signature based on aggregated co-occurrences with entity surface names (i.e., Hypothesis 1)

- Softly clustering synonymous relation phrases:

  → the type signatures of frequent relation phrases can help infer the type signatures of infrequent (sparse) ones that have similar cluster memberships

Hypothesis 3 (Type signature consistency):

If two relation phrases have similar cluster memberships, the type indicators of their left and right arguments (type signature) tend to be similar, respectively.

RP Clustering: Relation Phrase Similarity

- Existing work on relation phrase clustering utilizes strings; context words; entity argument to cluster synonymous relation phrases
- String similarity and distribution similarity may be insufficient to resolve two relation phrases; type information is particular helpful in such case
- We propose to leverage type signature of relation phrase, and proposed a general relation phrase clustering method to incorporate different features

  → further integrated with the graph-based type propagation in a mutually enhancing framework, based on following hypothesis

Hypothesis 4 (Relation phrase similarity):
Two relation phrases tend to have similar cluster memberships, if they have similar (1) strings; (2) context words; and (3) left and right argument type indicators

Type signatures $\mathbf{P}_L, \mathbf{P}_R \in \mathbb{R}^{l \times T}$
String features $\mathbf{F}_s \in \mathbb{R}^{l \times n_s}$
Context features $\mathbf{F}_c \in \mathbb{R}^{l \times n_c}$

Type Inference: A Joint Optimization Problem

$$\mathcal{O}_{\alpha,\gamma,\mu} = \mathcal{F}(\mathbf{C}, \mathbf{P}_L, \mathbf{P}_R) + \mathcal{L}_\alpha\big(\mathbf{P}_L, \mathbf{P}_R, \{\mathbf{U}^{(v)}, \mathbf{V}^{(v)}\}, \mathbf{U}^*\big) + \Omega_{\gamma,\mu}(\mathbf{Y}, \mathbf{C}, \mathbf{P}_L, \mathbf{P}_R). \quad (2)$$

$$\mathcal{F}(\mathbf{C}, \mathbf{P}_L, \mathbf{P}_R) = \sum_{i=1}^{n}\sum_{j=1}^{l} W_{L,ij}\left\|\frac{\mathbf{C}_i}{\sqrt{D^{(\mathcal{C})}_{L,ii}}} - \frac{\mathbf{P}_{L,j}}{\sqrt{D^{(\mathcal{P})}_{L,jj}}}\right\|_2^2 + \sum_{i=1}^{n}\sum_{j=1}^{l} W_{R,ij}\left\|\frac{\mathbf{C}_i}{\sqrt{D^{(\mathcal{C})}_{R,ii}}} - \frac{\mathbf{P}_{R,j}}{\sqrt{D^{(\mathcal{P})}_{R,jj}}}\right\|_2^2$$

Type propagation between entity surface names & relation phrases (Hypo 1)

Mention modeling & mention correlation (Hypo 2)

$$\Omega_{\gamma,\mu}(\mathbf{Y}, \mathbf{C}, \mathbf{P}_L, \mathbf{P}_R) = \big\|\mathbf{Y} - f(\Pi_{\mathcal{C}}\mathbf{C}, \Pi_L\mathbf{P}_L, \Pi_R\mathbf{P}_R)\big\|_F^2 + \frac{\gamma}{2}\sum_{c\in\mathcal{C}}\sum_{i,j=1}^{M_c} W^{(c)}_{ij}\left\|\frac{\mathbf{Y}_i}{\sqrt{D^{(c)}_{ii}}} - \frac{\mathbf{Y}_j}{\sqrt{D^{(c)}_{jj}}}\right\|_2^2 + \mu\|\mathbf{Y} - \mathbf{Y}_0\|_F^2$$

$$\mathcal{L}_\alpha\big(\mathbf{P}_L, \mathbf{P}_R, \{\mathbf{U}^{(v)}, \mathbf{V}^{(v)}\}, \mathbf{U}^*\big) = \sum_{v=0}^{d}\beta^{(v)}\big(\|\mathbf{F}^{(v)} - \mathbf{U}^{(v)}\mathbf{V}^{(v)T}\|_F^2 + \alpha\|\mathbf{U}^{(v)}\mathbf{Q}^{(v)} - \mathbf{U}^*\|_F^2\big). \quad (3)$$

Multi-view relation phrases clustering) (Hypo 3 & 4)

The ClusType Algorithm

$$\min_{\substack{\mathbf{Y},\mathbf{C},\mathbf{P},\mathbf{P}_R,\mathbf{U}^* \\ \{\mathbf{U}^{(v)},\mathbf{V}^{(v)},\beta^{(v)}\}}} \mathcal{O}_{\alpha,\gamma,\mu,\lambda_L,\lambda_\Omega}$$

$$\text{s.t.} \quad \mathbf{Y} \in \{0,1\}^{M\times T}, \quad \mathbf{Y1} = \mathbf{1};$$
$$\mathbf{U}^* \geq 0, \ \mathbf{U}^{(v)} \geq 0, \ \mathbf{V}^{(v)} \geq 0;$$
$$\sum_{v=0}^{d} \exp(-\beta^{(v)}) = 1, \ \forall 0 \leq v \leq d.$$

- Can be efficiently solved by alternate minimization based on block coordinate descent algorithm
- Algorithm complexity is linear to #entity mentions, #relation phrases, #cluster, #clustering features and #target types

The ClusType algorithm:

Update type indicators and type signatures

$$\mathbf{Y}^{(c)} = \left[(1+\gamma+\mu)\mathbf{I}_c - \gamma\mathbf{S}_{\mathcal{M}}^{(c)}\right]^{-1}\left(\mathbf{\Theta}^{(c)} + \mu\mathbf{Y}_0^{(c)}\right), \quad \forall c \in \mathcal{C}, \quad (7)$$

$$\mathbf{C} = \frac{1}{2}\left[\mathbf{S}_L\mathbf{P}_L + \mathbf{S}_R\mathbf{P}_R + \Pi_{\mathcal{C}}^T(\mathbf{Y} - \Pi_L\mathbf{P}_L - \Pi_R\mathbf{P}_R)\right]; \quad (8)$$

$$\mathbf{P}_L = \mathbf{X}_0^{-1}\left[\mathbf{S}_L^T\mathbf{C} + \Pi_L^T(\mathbf{Y} - \Pi_{\mathcal{C}}\mathbf{C} - \Pi_R\mathbf{P}_R) + \beta^{(0)}\mathbf{U}^{(0)}\mathbf{V}^{(0)T}\right];$$

$$\mathbf{P}_R = \mathbf{X}_1^{-1}\left[\mathbf{S}_R^T\mathbf{C} + \Pi_R^T(\mathbf{Y} - \Pi_{\mathcal{C}}\mathbf{C} - \Pi_L\mathbf{P}_L) + \beta^{(1)}\mathbf{U}^{(1)}\mathbf{V}^{(1)T}\right];$$

For each view, performs single-view NMF until converges

$$V_{jk}^{(v)} = V_{jk}^{(v)}\frac{[\mathbf{F}^{(v)T}\mathbf{U}^{(v)}]_{jk} + \alpha\sum_{i=1}^{l} U_{ik}^* U_{ik}^{(v)}}{\mathbf{\Delta}_{jk}^{(v)} + \alpha\left(\sum_{i=1}^{l} U_{ik}^{(v)2}\right)\left(\sum_{i=1}^{T} V_{ik}^{(v)}\right)}, \quad (9)$$

$$U_{ik}^{(v)} = U_{ik}^{(v)}\frac{[\mathbf{F}^{(v)+}\mathbf{V}^{(v)} + \alpha\mathbf{U}^*]_{ik}}{[\mathbf{U}^{(v)}\mathbf{V}^{(v)T}\mathbf{V}^{(v)} + \mathbf{F}^{(v)-}\mathbf{V}^{(v)} + \alpha\mathbf{U}^{(v)}]_{ik}}. \quad (10)$$

Update consensus matrix and relative weights of different views

$$\mathbf{U}^* = \frac{\sum_{v=0}^{d}\beta^{(v)}\mathbf{U}^{(v)}\mathbf{Q}^{(v)}}{\sum_{v=0}^{d}\beta^{(v)}}; \quad \beta^{(v)} = -\log\left(\frac{\delta^{(v)}}{\sum_{i=0}^{d}\delta^{(i)}}\right). \quad (12)$$

Until the objective converges

ClusType: Experiment Setting

- Datasets: 2013 New York Times news (~110k docs) [event, PER, LOC, ORG]; Yelp Reviews (~230k) [Food, Job, …]; 2011 Tweets (~300k) [event, product, PER, LOC, …]
- Seed mention sets: < 7% extracted mentions are mapped to Freebase entities
- Evaluation sets: manually annotate mentions of target types for subsets of the corpora
- Evaluation metrics: Follows named entity recognition evaluation (Precision, Recall, F1)
- Compared methods
  - **Pattern**: (Stanford, CONLL'14) Stanford pattern-based learning;
  - **SemTagger**: (U. Utah, ACL'10) bootstrapping method which trains contextual classifier based on seed mentions
  - **FIGER** (UW, AAAI'12): distantly-supervised sequence labeling method trained on Wiki corpus
  - **NNPLB** (UW, EMNLP'12) label propagation using ReVerb assertion and seed mention
  - **APOLLO** (THU, CIKM'12): mention-level label propagation using Wiki concepts and KB entities

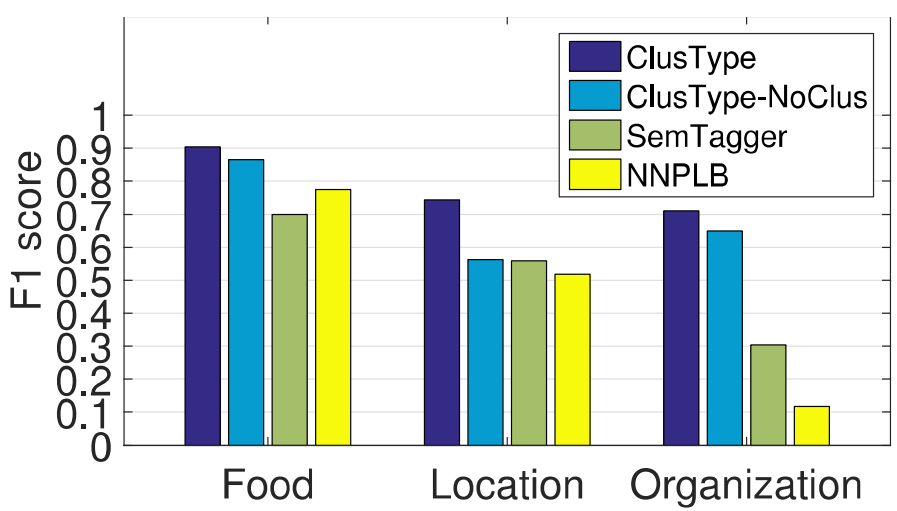
ClusType: Comparing with the State-of-the-Art Systems

Data sets	NYT			Yelp			Tweet		
Method	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Pattern [7]	0.4576	0.2247	0.3014	0.3790	0.1354	0.1996	0.2107	0.2368	0.2230
FIGER [12]	0.8668	0.8964	0.8814	0.5010	0.1237	0.1983	**0.7354**	0.1951	0.3084
SemTagger [9]	0.8667	0.2658	0.4069	0.3769	0.2440	0.2963	0.4225	0.1632	0.2355
APOLLO [22]	0.9257	0.6972	0.7954	0.3534	0.2366	0.2834	0.1471	0.2635	0.1883
NNPLB [11]	0.7487	0.5538	0.6367	0.4248	0.6397	0.5106	0.3327	0.1951	0.2459
ClusType-NoClus	0.9130	0.8685	0.8902	0.7629	0.7581	0.7605	0.3466	0.4920	0.4067
ClusType-NoWm	0.9244	0.9015	0.9128	0.7812	0.7634	0.7722	0.3539	**0.5434**	0.4286
ClusType-TwoStep	0.9257	0.9033	0.9143	0.8025	0.7629	0.7821	0.3748	0.5230	0.4367
ClusType	**0.9550**	**0.9243**	**0.9394**	**0.8333**	**0.7849**	**0.8084**	0.3956	0.5230	**0.4505**

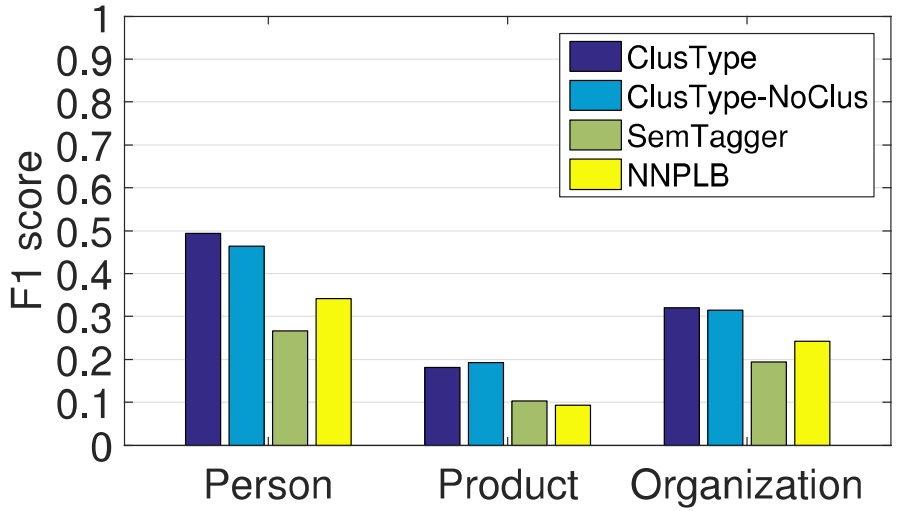
- **Pattern** (Stanford, CONLL'14): explicit textual pattern; semantic drift
- **NNPLB** (UW, EMNLP'12)**:** type propagation on surface name level (name ambiguity)
- **APOLLO** (THU, CIKM'12): **context sparsity** in type propagation
- **FIGER** (UW, AAAI'12): reliance on complex linguistic features (domain restriction)
- **ClusType-NoWm**: ignore mention correlation; **ClusType-NoClus**: conducts only type propagation; **ClusType-TwoStep**: first performs hard clustering then type propagation

Precision (***P***) = $\frac{\#Correctly-typed\ mentions}{\#System-recognized\ mentions}$, Recall (***R***) = $\frac{\#Correctly-typed\ mentions}{\#ground-truth\ mentions}$, F1 score = $\frac{2(P\times R)}{(P+R)}$

Comparing on Different Entity Types



(a) Yelp



(b) Tweet

Obtains larger gain on *organization* and *person* (more entities with ambiguous surface names)

Modeling types on entity mention level is critical for name disambiguation

Superior performance on product and food mainly comes from the domain independence of our method

Both NNPLB and SemTagger require sophisticated linguistic feature generation which is hard to adapt to new types

Comparing on Trained NER System

- Compare with Stanford NER,which is trained on general-domain corpora including ACE corpus and MUC corpus, on three types: PER, LOC, ORG

F1 score comparison with trained NER

Method	NYT	Yelp	Tweet
Stanford NER *	0.6819	0.2403	0.4383
ClusType-NoClus	0.9031	0.4522	0.4167
ClusType	**0.9419**	**0.5943**	**0.4717**

- ClusType and its variants outperform Stanford NER on both dynamic corpus (NYT) and domain-specific corpus (Yelp)
- ClusType has lower precision but higher Recall and F1 score on Tweet → Superior recall of ClusType mainly come from domain-independent candidate generation

* J. R. Finkel, T. Grenager and C. Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In ACL'05.

Example Output and Relation Phrase Clusters

Example output of ClusType and the compared methods on the Yelp dataset

ClusType	SemTagger	NNPLB
The best **BBQ:Food** I've tasted in **Phoenix:LOC** ! I had the **[pulled pork sandwich]:Food** with **coleslaw:Food** and **[baked beans]:Food** for lunch. ...	The best BBQ I've tasted in **Phoenix:LOC** ! I had the pulled **[pork sandwich]:LOC** with **coleslaw:Food** and **[baked beans]:LOC** for lunch. ...	The best **BBQ:Loc** I've tasted in **Phoenix:LOC** ! I had the pulled pork **sandwich:Food** with coleslaw and baked **beans:Food** for **lunch:Food**. ...
I only go to **ihop:LOC** for **pancakes:Food** because I don't really like anything else on the menu. Ordered **[chocolate chip pancakes]:Food** and a **[hot chocolate]:Food**.	I only go to ihop for pancakes because I don't really like anything else on the menu. Ordered **[chocolate chip pancakes]:LOC** and a **[hot chocolate]:LOC**.	I only go to ihop for pancakes because I don't really like anything else on the menu. Ordered chocolate chip pancakes and a hot chocolate.

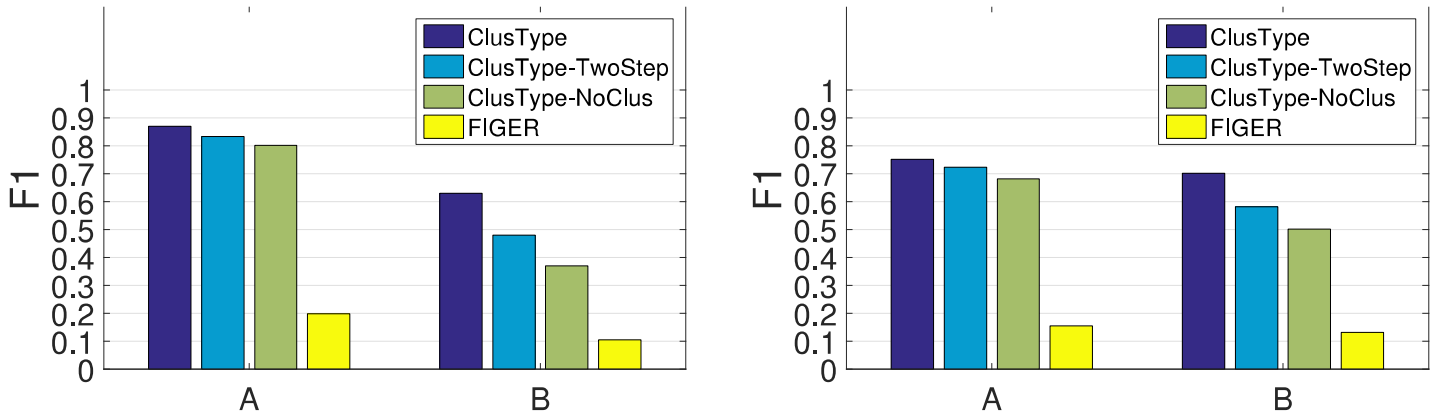
- Extracts more mentions and predicts types with higher accuracy

Example relation phrase clusters and corpus-wide frequency from the NYT dataset

ID	Relation phrase
1	recruited by (5.1k); employed by (3.4k); want hire by (264)
2	go against (2.4k); struggling so much against (54); run for re-election against (112); campaigned against (1.3k)
3	looking at ways around (105); pitched around (1.9k); echo around (844); present at (5.5k);

- Not only synonymous relation phrases, but also both sparse and frequent relation phrase can be clustered together
- → boosts sparse relation phrases with type information of frequent relation phrases

Testing on Context Sparsity and Surface Name popularity



(a) Context sparsity (b) Surface name popularity

Figure 8: Case studies on context sparsity and surface name popularity on the Tweet dataset.

- **Surface name popularity:**
  - Group A: high frequency surface name
  - Group B: infrequent surface name
- ClusType outperforms its variants on Group B
  - → Handles well mentions with insufficient corpus statistics

- **Context sparsity:**
  - Group A: frequent relation phrases
  - Group B: sparse relation phrases
- ClusType obtains superior performance over its variants on Group B
  - → clustering relation phrase is critical for sparse relation phrases

ClusType: Conclusions and Future Work

- Study distantly-supervised entity recognition for domain-specific corpora and propose a novel relation phrase-based framework
  - A data-driven, domain-agnostic phrase mining algorithm for candidate entity mentions and relation phrase generation
  - Integrate relation phrase clustering with type propagation on heterogeneous graphs, and solve it by a joint optimization problem.

Ongoing:

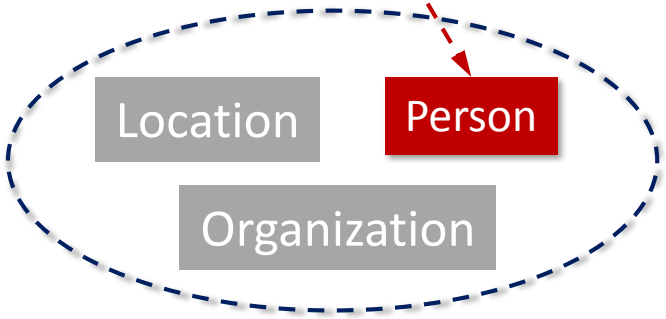
- Extend to role discovery for scientific concepts → paper profiling (research/demo)
- Study of relation phrase clustering, such as
  - joint entity/relation clustering
  - synonymous relation phrase canonicalization
- Study of joint entity and relation phrase extraction with phrase mining

Outline

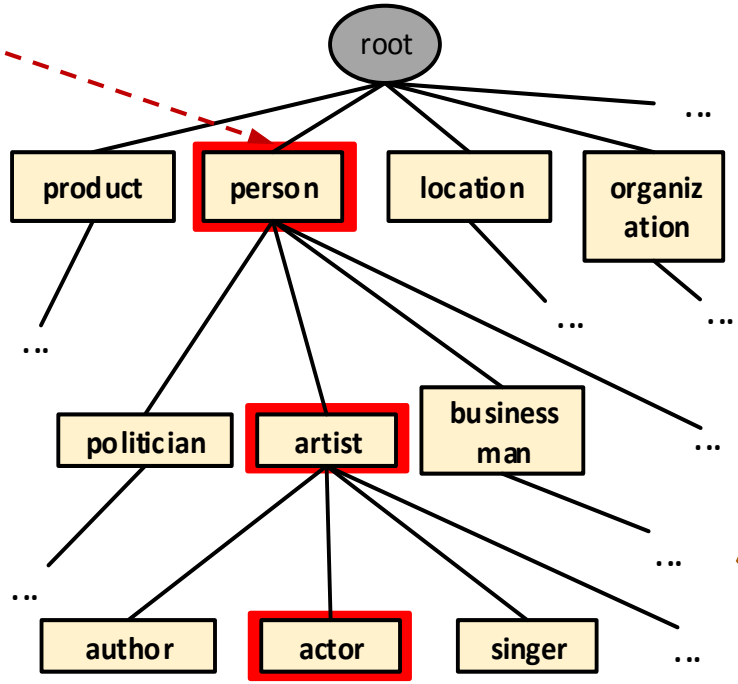
- Why Text to Networks?
- Phrase Mining and Topic Modeling from Large Text Corpora
  - Why Phrase Mining
  - Previous Approaches
  - TopMine
  - SegPhrase and AutoPhrase
- Entity Recognition and Typing for Massive Text Data 
  - ClusType: Entity Extraction and Typing by Relational Graph Construction and Propagation
  - Refined Entity Typing: PLE and AFET 
  - CoType: Joint Typing of Entities and Relations

Fine-Grained Entity Typing

ID	Sentence
S1	*Donald Trump* spent 14 television seasons presiding over a business-themed game show, NBC's The Apprentice



A few common types



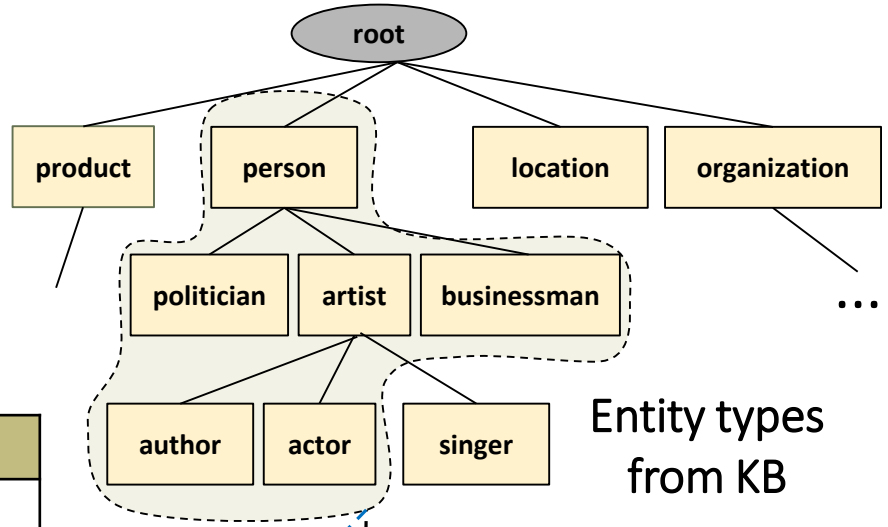
A type hierarchy with 100+ types

The need of fine grained typing

- **Features for deeper NLP tasks**
  - **Relation extraction (Ling & Weld, 2012)**
- **Assists downstream applications**
  - **Question answering**

Current Distant Supervision: The "Context-Agnostic Label" Challenge

- "Context-agnostic" type assignment in **training data**
- Existing work: All labels are "perfect" training labels

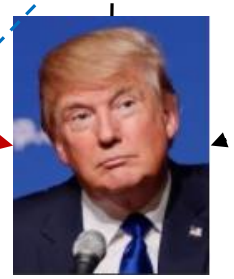


Entity types from KB

ID	Sentence
S1	*Donald Trump* spent 14 television seasons presiding over a business-themed game show, NBC's The Apprentice

S1: *Donald Trump*

Entity Types: person, artist, actor, ~~author~~, ~~businessman~~, ~~politician~~



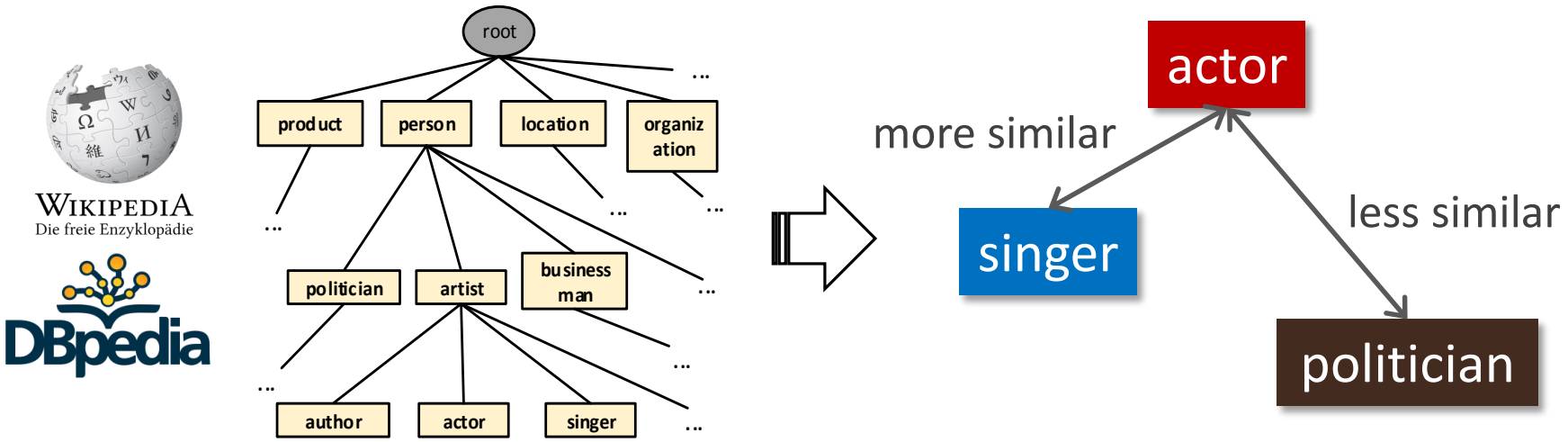
Entity: ***Donald Trump***

WIKIPEDIA
Die freie Enzyklopädie

DBpedia

Current Distant Supervision: Label-Independence Assumption

- Entity types are not independent → **correlated**



- Existing studies ignore such correlation information

How to deal with infrequent entity types?
→ "data sparsity" on entity type level

106

Partial Label Embedding (Ren et al., KDD'16)

Extract Text Features

Label Noise Reduction

Train Classifiers on De-noised Data

Prediction on New Data

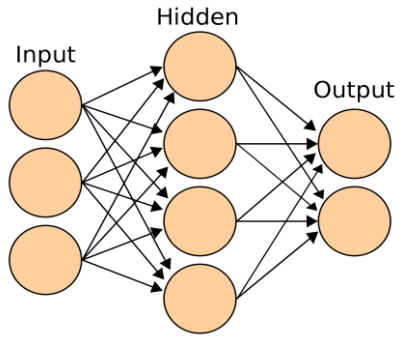
ID	Sentence
S1	*Donald Trump* spent 14 television seasons presiding over a business-themed game show, NBC's The Apprentice

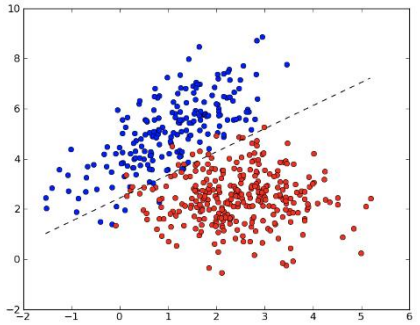
Text features: HEAD_Donald, CXT_A: television, CXT_A: season, POS: NN, TKN_trump, SHAPE: AA

S1: *Donald Trump*

Entity Types: person, artist, actor, ~~author~~, ~~businessman~~, ~~politician~~

De-noised labeled data





"Robust" classifier

X. Ren et al. *Label Noise Reduction in Entity Typing by Heterogeneous Partial-Label Embedding*. KDD, 2016

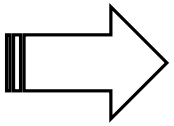
PLE: Modeling Clean and Noisy Mentions Separately

For a clean mention, its "*positive types*" should be **ranked higher** than all its "*negative types*"

ID	Noisy Entity Mention
S1	***Donald Trump*** spent 14 television seasons presiding over a business-themed game show, NBC's The Apprentice

S1: *Donald Trump*

Entity Types: person, artist, actor, author, businessman, politician



Types ranked

(+) actor
(-) singer
(-) coach
(-) doctor
(-) location
(-) organization

"**Best**" candidate type

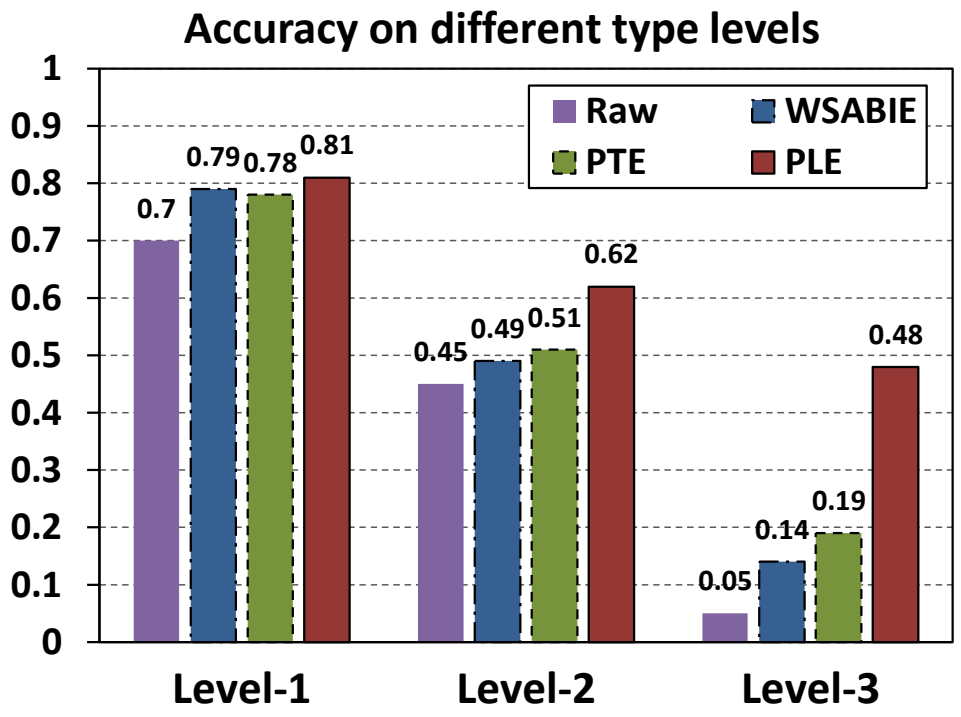
(+) actor 0.88
(+) artist 0.74
(+) person 0.55
(+) author 0.41
(+) politician 0.33
(+) business 0.31

Measured based on currently estimated embedding space

For a noisy mention, its "*best candidate type*" should be **ranked higher** than all its "*non-candidate types*"

PLE: Performance of Fine-Grained Entity Typing

$$\text{Accuracy} = \frac{\#\ mentions\ with\ all\ types\ correctly\ predicted}{\#\ mentions\ in\ the\ test\ set}$$

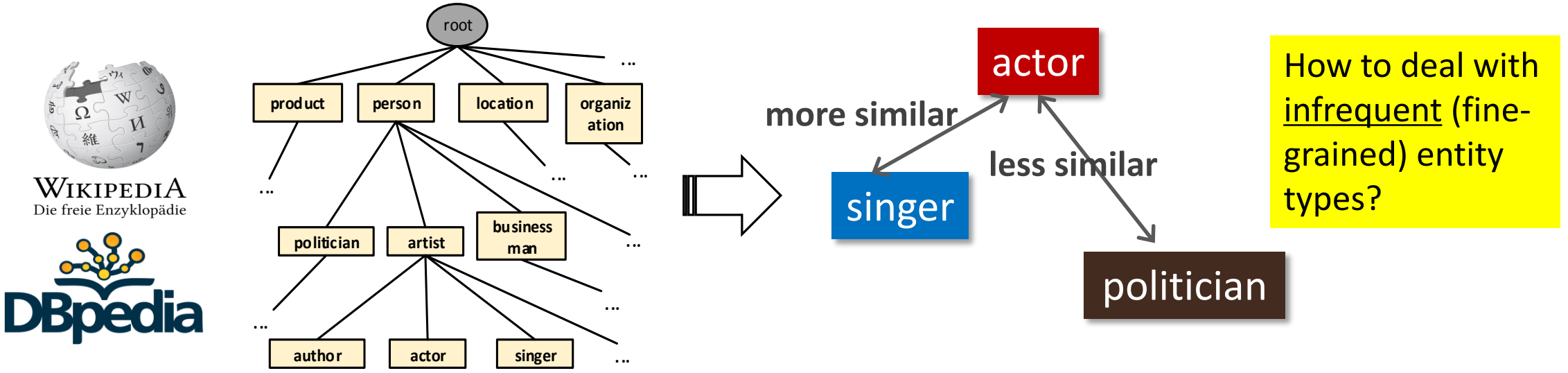


OntoNotes dataset (Weischedel et al. 2011, Gillick et al., 2014): 13,109 news articles, 77 annotated documents, 89 types
https://catalog.ldc.upenn.edu/LDC2013T19

- **Raw**: candidate types from distant supervision
- **WASBIE** (Google, ACL'14): joint feature and type embedding
- **PTE** (MSR, WWW'15): joint mention, feature and type embedding
  - Both WASBIE and PTE suffer from context-agnostic labels
- **PLE** (KDD'16): partial-label loss + type correlation modeling

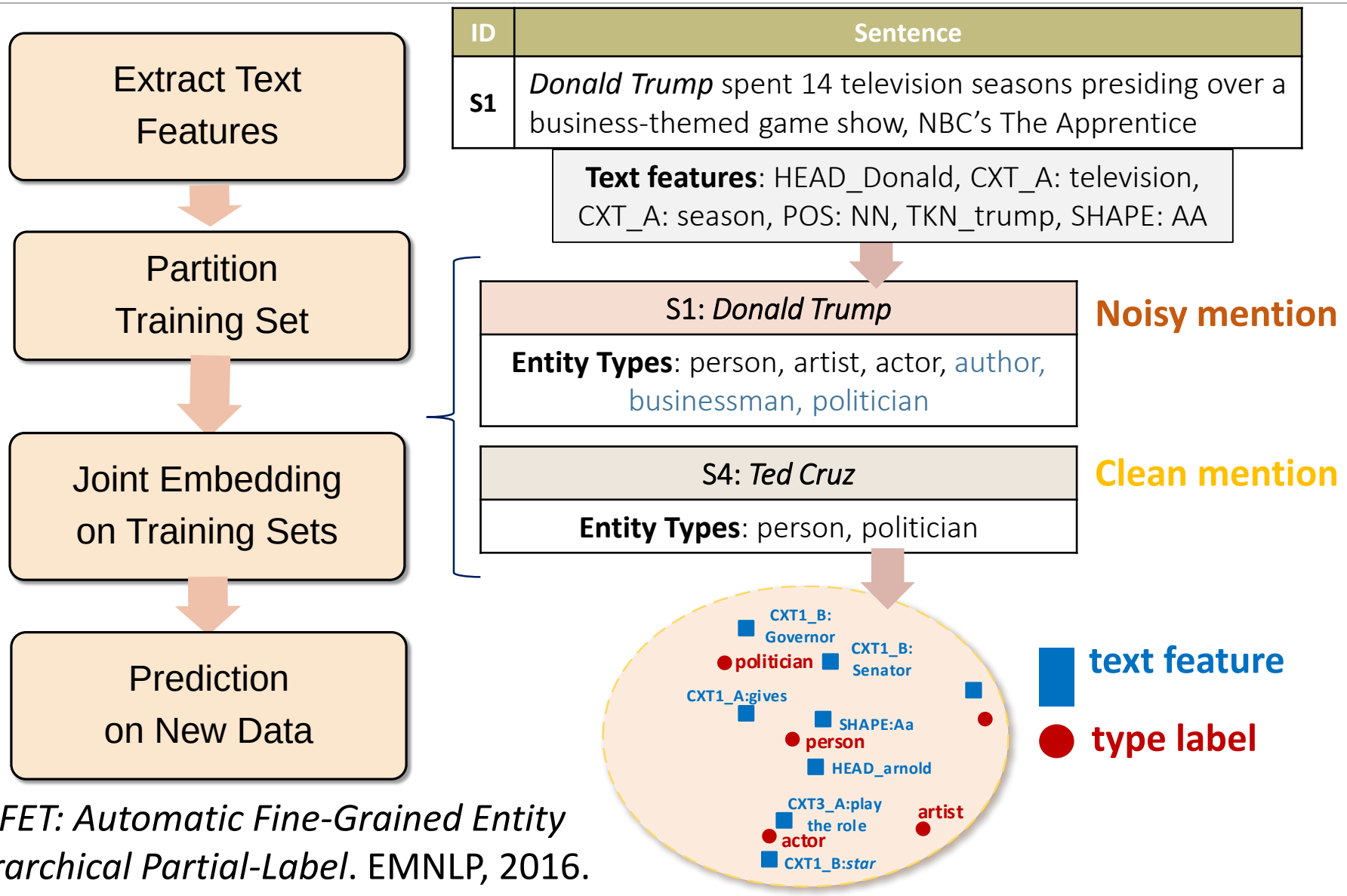
AFET (EMNLP16) and "Type Correlation" Challenge

- Distant supervision: Existing studies ignore type correlations
- In reality, entity types are not independent → **correlated**
- AFET: Automatic Fine-Grained Entity Typing by Hierarchical Partial-Label (EMNLP'16)



- Jointly embed entity mentions and type labels into a low-dimensional vector space (to capture type semantics)
- Design a noise-robust loss function to model "false positive" type labels in noisy training data
- Enforce adaptive margin on entity mentions, to encode type correlation

AFET (EMNLP'16): Framework Overview



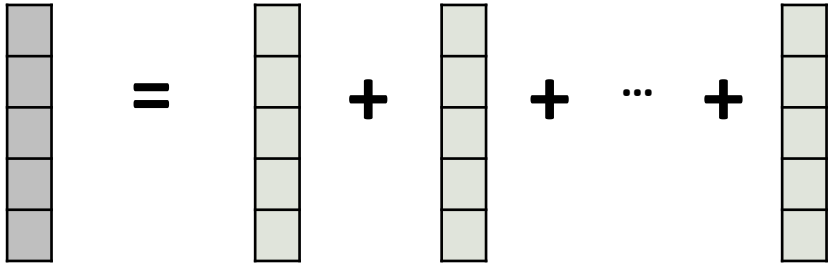
X. Ren et al. *AFET: Automatic Fine-Grained Entity Typing by Hierarchical Partial-Label*. EMNLP, 2016.

Type Inference in AFET

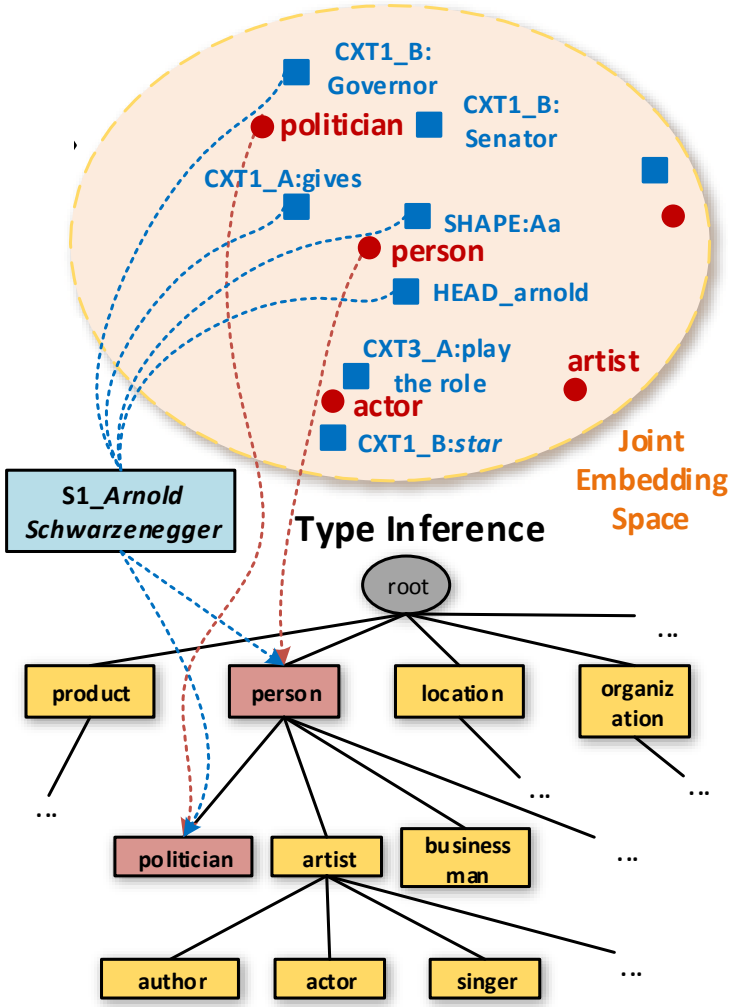
- Top-down nearest neighbor search in the given type hierarchy

ID	Sentence
S2	Governor ***Arnold Schwarzenegger*** gives a speech at Mission Serve's service project on Veterans Day 2010.

Test mention: ***S1_Arnold Schwarzenegger*** **Governor** **gives** **speech**



Learned embeddings for text features



AFET: Performance of Fine-Grained Entity Typing

$$\text{Accuracy} = \frac{\#\ mentions\ with\ all\ types\ correctly\ predicted}{\#\ unseen\ entity\ mentions\ in\ the\ test\ set}$$

	Methods	Wikipedia	OntoNotes	BBN
Fine-grained classifiers	**FIGER** (UW, AAAI'12)	0.474	0.369	0.467
	HYENA (Max-Planck, COLING'12)	0.288	0.249	0.523
Embedding-based Methods	**WASABIE** (Google, ACL'14)	0.480	0.404	0.619
	HNM (IBM, EMNLP'15)	0.237	0.122	0.551
	PTE (MSR, WWW'15)	0.405	0.436	0.604
Partial-label Learning	**PL-SVM** (Cornell, KDD'08)	0.428	0.225	0.465
	AFET (EMNLP'16)	0.533	0.551	0.670

- **Partial-label loss for modeling noisy labels (vs. fine-grained classifier, embedding methods)**
- **Adaptive margins for capturing type correlation (vs. PL-SVM, all)**

- Wikipedia dataset (Ling & Weld, 2012): 1.5M sentences, 113 types
- OntoNotes dataset (Weischedel et al. 2011, Gillick et al., 2014): 13,109 news articles, 89 types
- BBN dataset (Weischedel & Brunstein, 2005): 2,311 news articles, 93 types

Outline

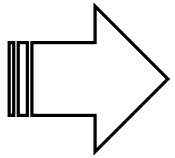
- Why Text to Networks?
- Phrase Mining and Topic Modeling from Large Text Corpora
  - Why Phrase Mining
  - Previous Approaches
  - TopMine
  - SegPhrase and AutoPhrase
- Entity Recognition and Typing for Massive Text Data 
  - ClusType: Entity Extraction and Typing by Relational Graph Construction and Propagation
  - Refined Entity Typing: PLE and AFET
  - CoType: Joint Typing of Entities and Relations 

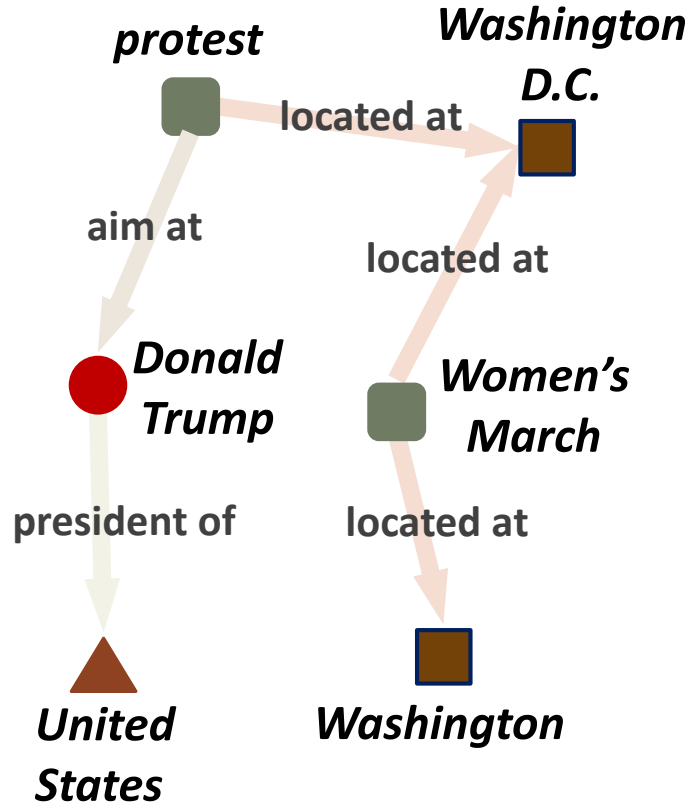
Outline

- Why Text to Networks?
- Phrase Mining and Topic Modeling from Large Text Corpora
  - Why Phrase Mining
  - Previous Approaches
  - TopMine
  - SegPhrase and AutoPhrase
- Entity Recognition and Typing for Massive Text Data
  - ClusType: Entity Extraction and Typing by Relational Graph Construction and Propagation
  - Refined Entity Typing: PLE and AFET
  - CoType: Joint Typing of Entities and Relations

Joint Extraction of Typed Entities and Relations

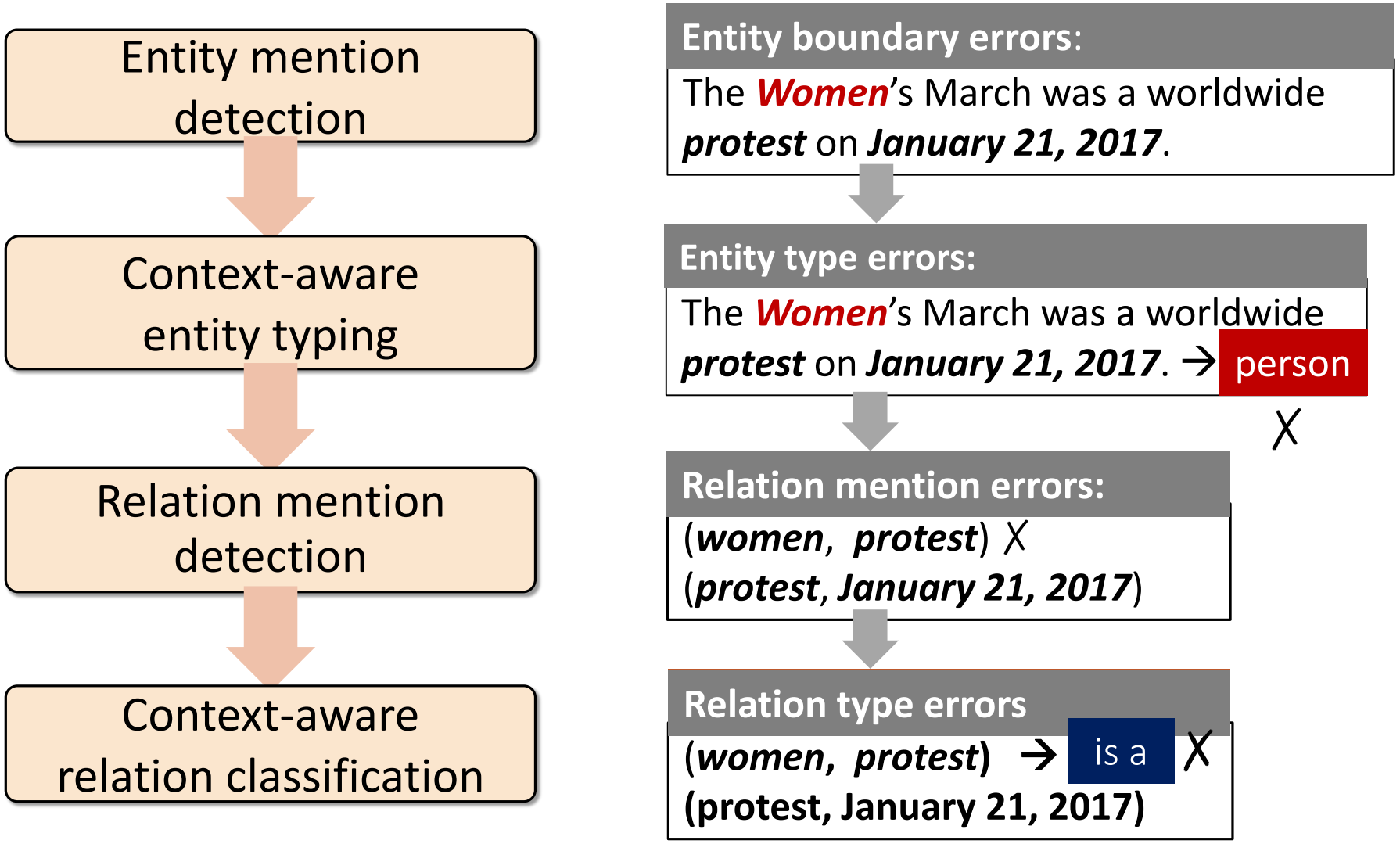
The **Women's March** was a worldwide **protest** on January 21, 2017. The **protest** was aimed at **Donald Trump**, the recently inaugurated president of the **United States**. The first **protest** was planned in **Washington, D.C.**, and was known as the **Women's March** on **Washington**.



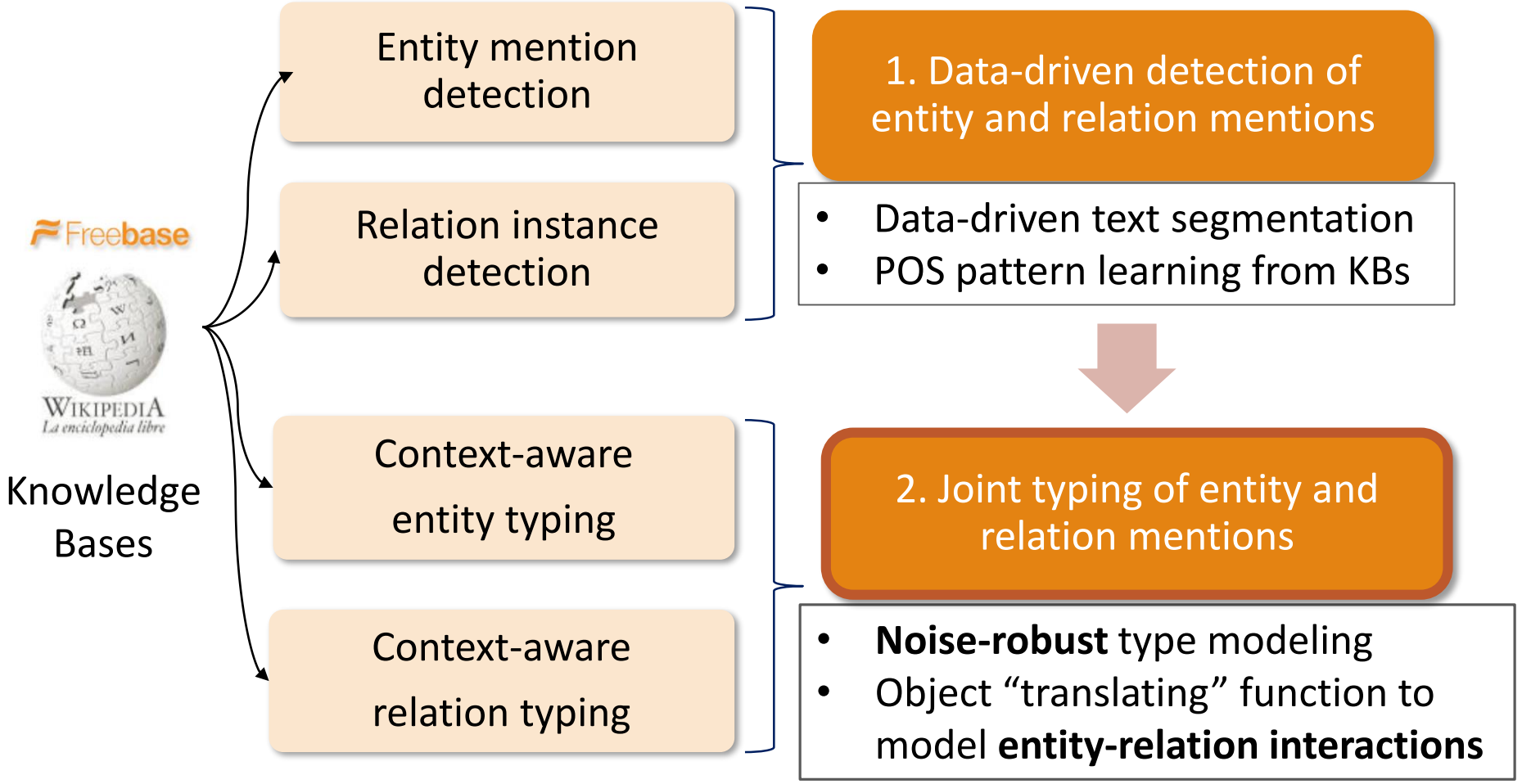


Prior Work: An "Incremental" System Pipeline

Error propagation cascading down the pipeline

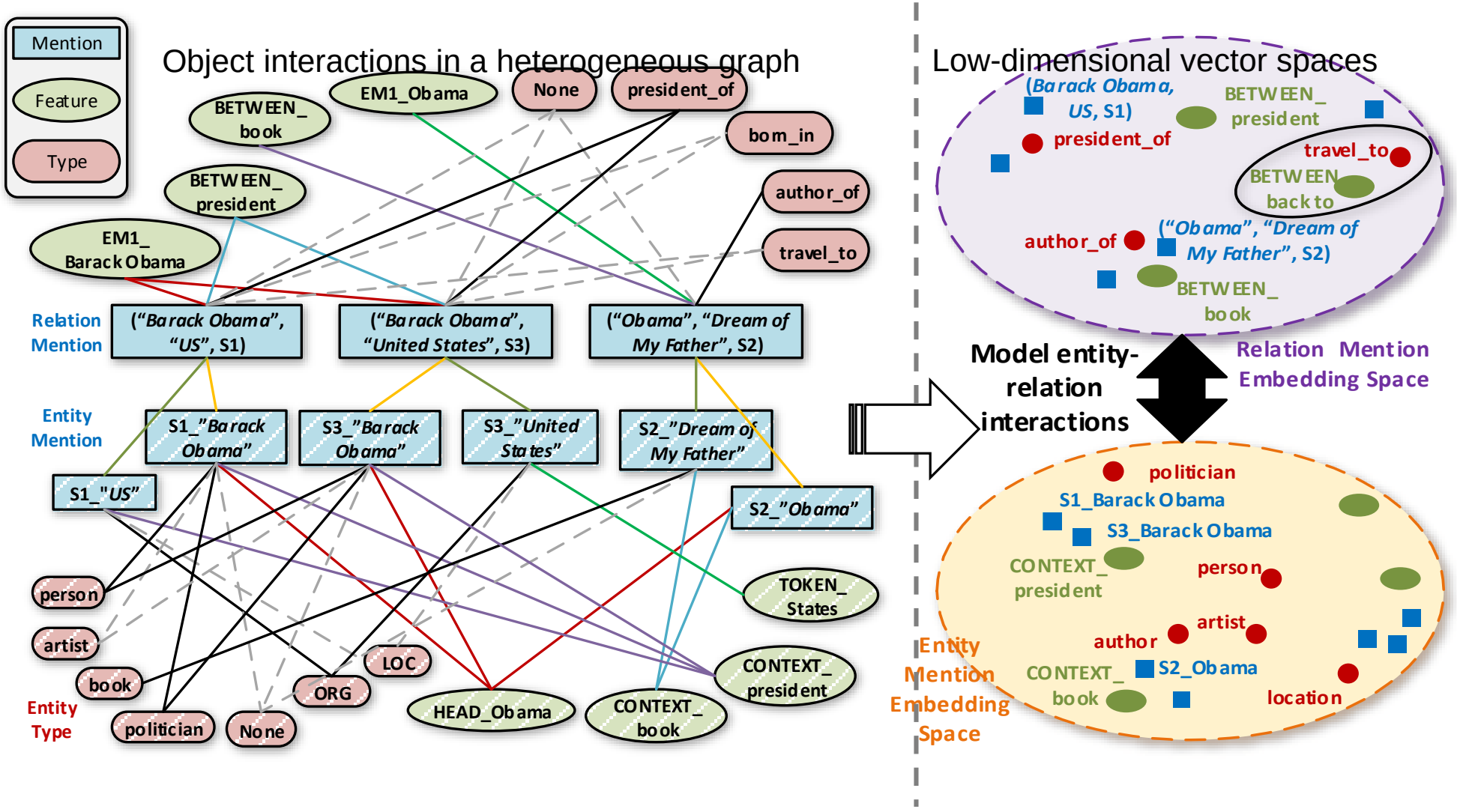


CoType: Joint Extraction of Typed Entities and Relations with Knowledge Bases (WWW'17)



X. Ren et al. *CoType: Joint Extraction of Typed Entities and Relations with Knowledge Bases*. WWW, 2017.

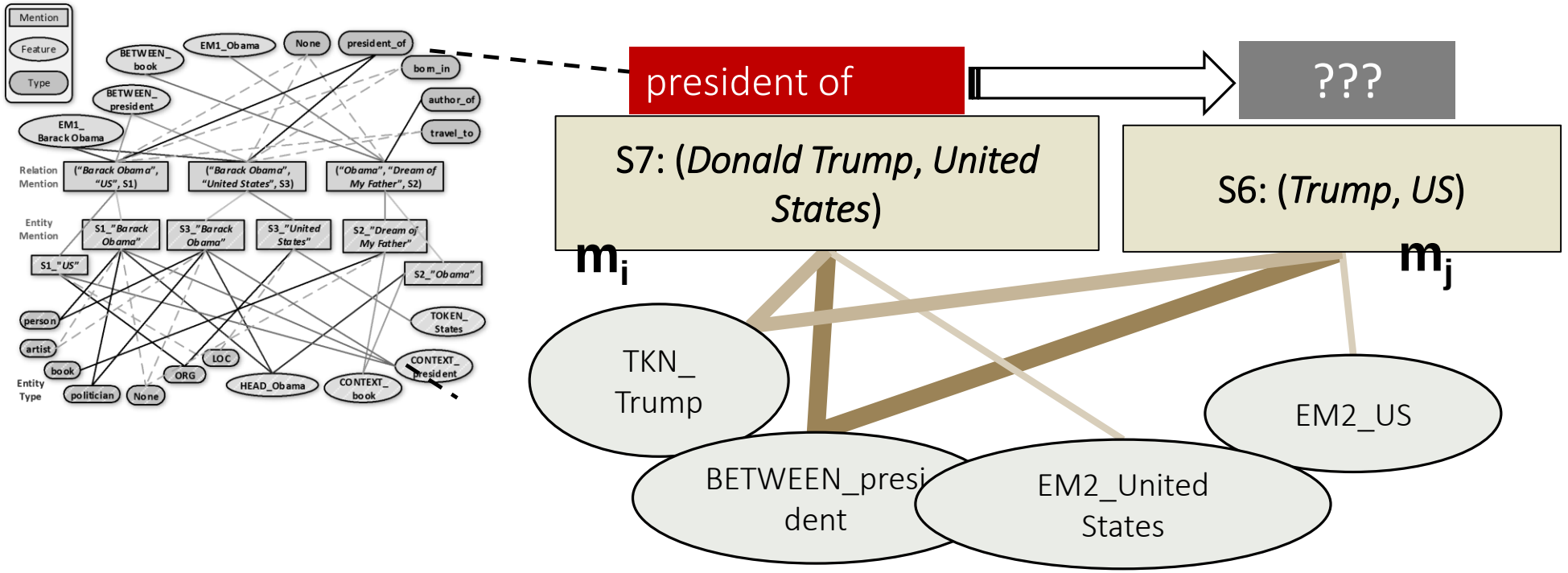
CoType: Co-Embedding for Typing Entities and Relations



Modeling Mention-Feature Co-Occurrences

- Second-order Proximity
  - Mentions with similar distributions over text features should have similar types

Vertex $\mathbf{m_i}$ and $\mathbf{m_j}$ have a large second-order proximity



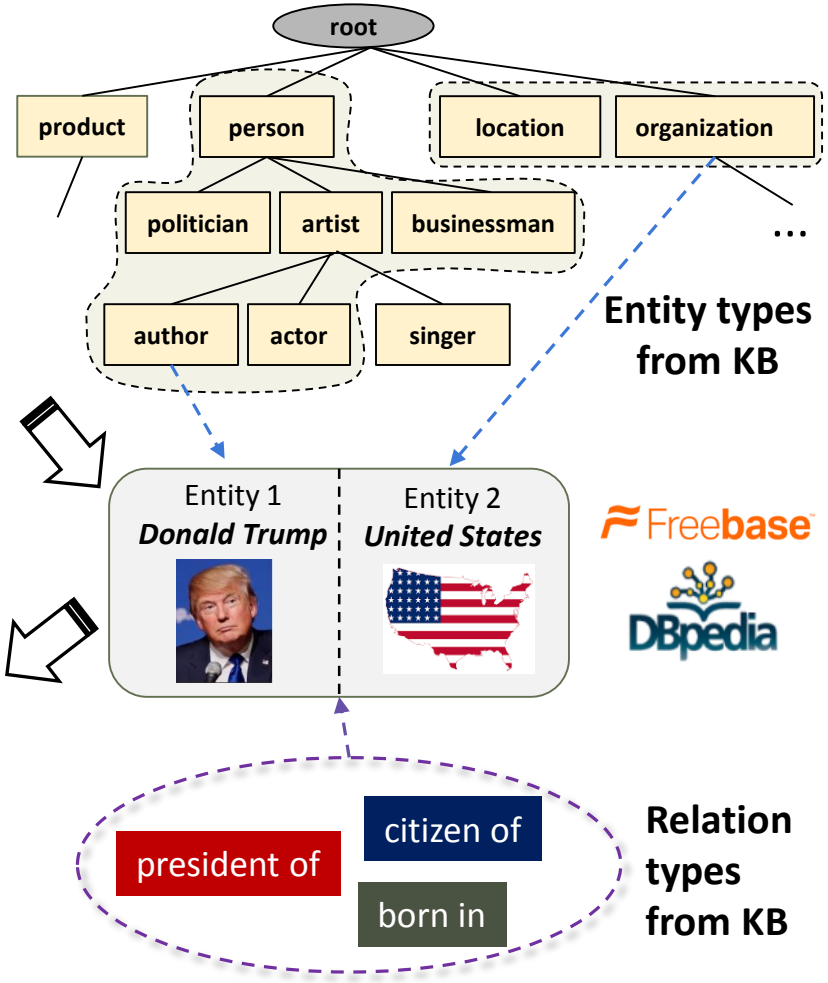
(Tang et al., WWW'15), (Ren et al. WWW'17)

Challenge: "Context-Agnostic Label"

ID	Sentence
S1	*Donald Trump* was born in Queens, New York, *USA* on June 14,1946.
S2	The protest was aimed at *Donald Trump*, the recently inaugurated president of the *United States*.
S3	There is a method to *Donald Trump*'s madness and he laid it all out in his book, "*The Art of the Deal*."

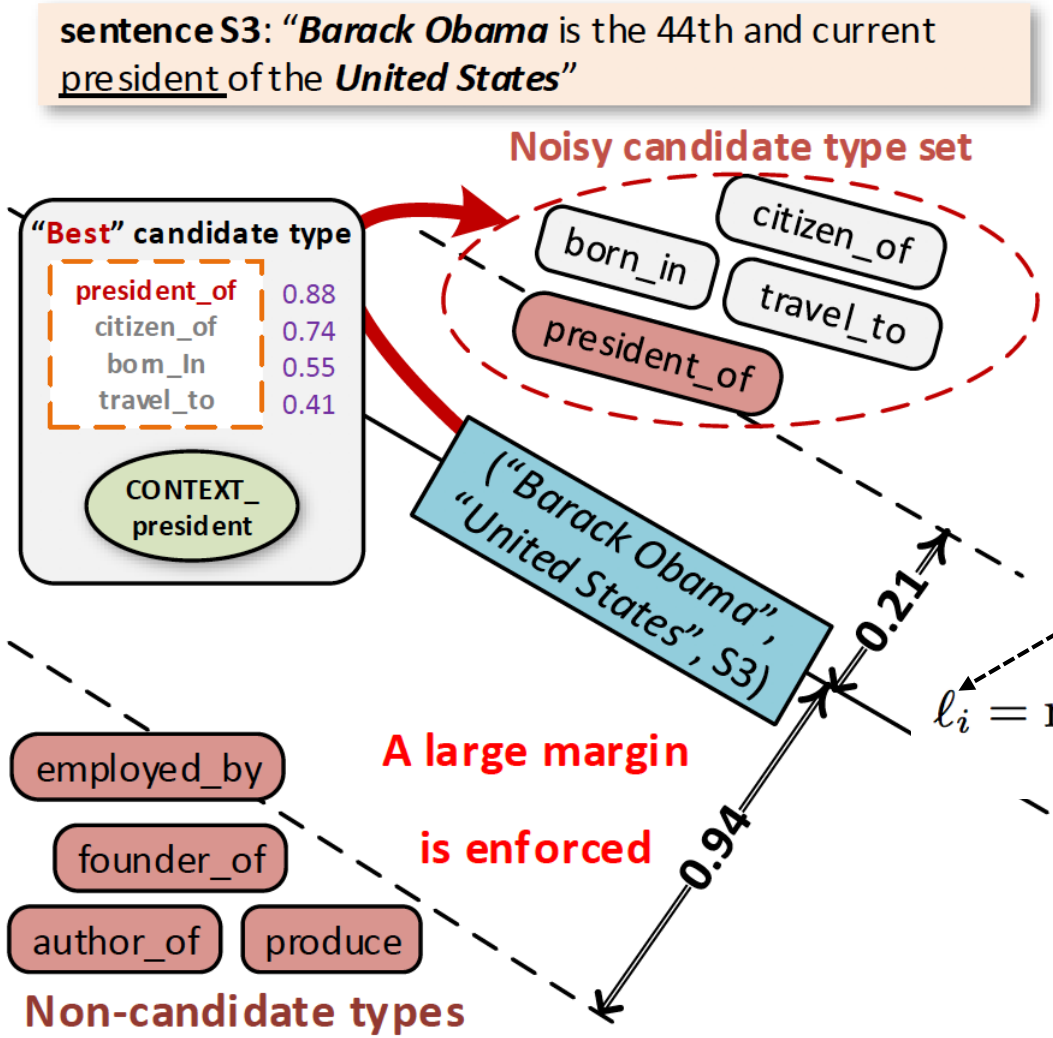
Type labels for relation mention in S2:

E1: *Donald J. Trump*	**E2**: *United States*
E1 Types: person, politician, ~~businessman~~, ~~author~~, ~~actor~~	**E2 Types**: location, ~~organization~~
Relations between E1, E2: president of, ~~citizen of~~, ~~born in~~	



Surdeanu et al. *Multi-instance multi-label learning for relation extraction*. EMNLP, 2012.

Context-Aware Type Modeling



Partial-label Loss Function

- A **relation mention** should be more similar to its "**most relevant**" **candidate type**, than to any other **non-candidate type**

Enforce: mention is more similar to its "most relevant" candidate type

$$\ell_i = \max\left\{0, 1 - \left[\max_{y \in \mathcal{Y}_i} s(m_i, y) - \max_{y' \in \overline{\mathcal{Y}}_i} s(m_i, y')\right]\right\}$$

Score for "most relevant" type

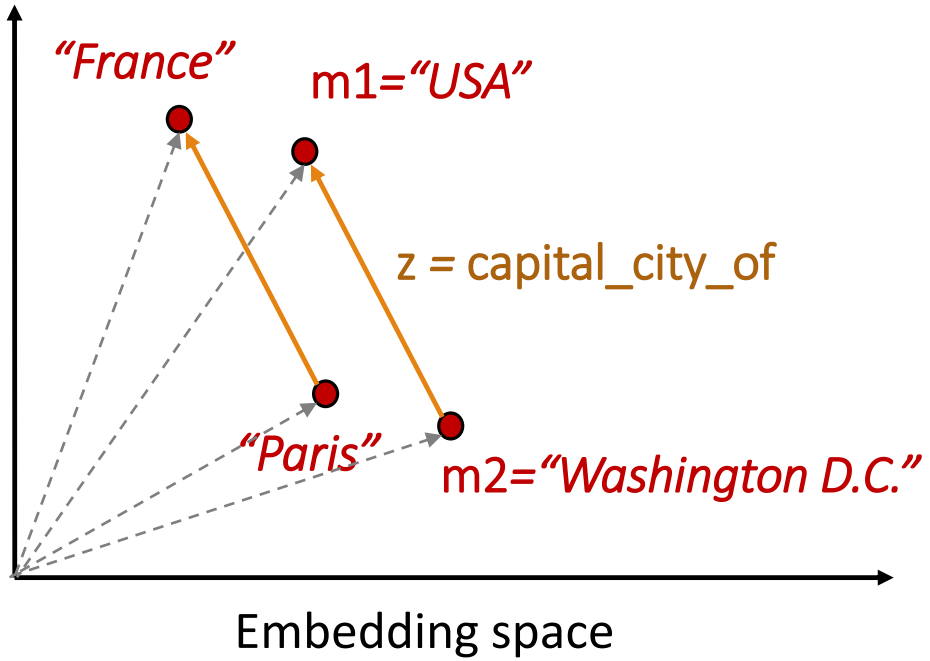
Maximal score for non-candidate types

Modeling Entity-Relation Interactions

Object "Translating" Assumption

For a relation mention **z** of entity mentions **m1** and **m2**,

vec(m1) ≈ **vec**(m2) + **vec**(z)



- Error on an entity-relation triple (z, m1, m2):

$$\tau(z) = \|\mathbf{m}_1 + \mathbf{z} - \mathbf{m}_2\|_2^2$$

Enforce: error on a positive triple should be smaller than error on a negative triple

$$\sum_{z_i \in \mathcal{Z}_L} \sum_{v=1}^{V} \max\{0, 1 + \tau(z_i) - \tau(z_v)\}$$

positive relation triple

negative relation triple

Reducing Error Propagation: A Joint Optimization Framework

Modeling **Entity-relation** interactions

$$O_{ZM} = \sum_{z_i \in \mathcal{Z}_L} \sum_{v=1}^{V} \max\{0, 1 + \tau(z_i) - \tau(z_v)\}$$

$$\min \mathcal{O} = \mathcal{O}_M + \mathcal{O}_Z + \mathcal{O}_{ZM}$$

Modeling **relation** mentions

$$O_Z = \mathcal{L}_{ZF} + \sum_{i=1}^{N_L} \ell_i + \frac{\lambda}{2} \sum_{i=1}^{N_L} \|\mathbf{z}_i\|_2^2 + \frac{\lambda}{2} \sum_{k=1}^{K_r} \|\mathbf{r}_k\|_2^2$$

Modeling **entity** mentions

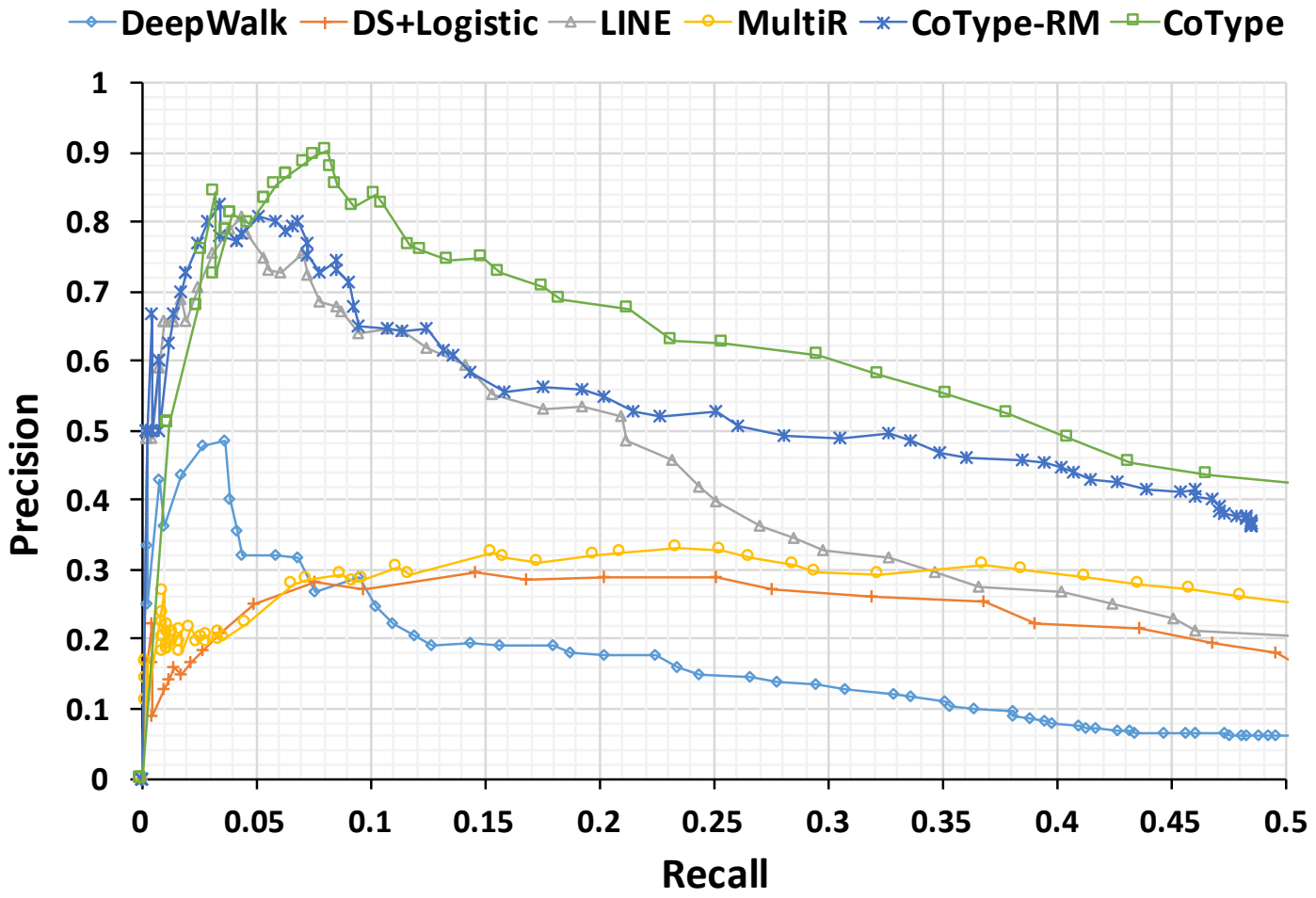
$$O_M = \mathcal{L}_{MF} + \sum_{i=1}^{N'_L} \ell'_i + \frac{\lambda}{2} \sum_{i=1}^{N'_L} \|\mathbf{m}_i\|_2^2 + \frac{\lambda}{2} \sum_{k=1}^{K_y} \|\mathbf{y}_k\|_2^2$$

Please refer the paper for details:

X. Ren et al. *CoType: Joint Extraction of Typed Entities and Relations with Knowledge Bases*. WWW2017.

CoType: Comparing with State-of-the-Arts RE Systems

- Given candidate relation mentions, predict its relation type if it expresses a relation of interest; otherwise, output "None"



- DeepWalk (StonyBrook, KDD'14): homogeneous graph embedding
- DS+Logistic (Stanford, ACL'09): trains logistic classifier on DS
- LINE (MSR, WWW'15): joint feature and type embedding
- MultiR (UW, ACL'11): distantly-supervised, models noisy labels
- CoType-RM (WWW'17): only models relation mentions
- CoType (WWW'17): models entity-relation interactions

References on Entity Recognition and Typing

- X. Ren, A. El-Kishky, C. Wang, F. Tao, C. R. Voss, H. Ji and J. Han. ClusType: Effective Entity Recognition and Typing by Relation Phrase-Based Clustering. KDD'15
- Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare Voss, Heng Ji, Tarek Abdelzaher and Jiawei Han, "CoType: Joint Extraction of Typed Entities and Relations with Knowledge Bases", WWW'17
- Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han, "AFET: Automatic Fine-Grained Entity Typing by Hierarchical Partial-Label Embedding", EMNLP'16
- Xiang Ren, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, Jiawei Han, "Label Noise Reduction in Entity Typing by Heterogeneous Partial-Label Embedding", KDD'16
- H. Huang, Y. Cao, X. Huang, H. Ji and C. Lin. Collective Tweet Wikification based on Semi-supervised Graph Regularization. ACL'14
- T. Lin, O. Etzioni, et al. No noun phrase left behind: detecting and typing unlinkable entities. EMNLP'12
- N. Nakashole, T. Tylenda, and G. Weikum. Fine-grained semantic typing of emerging entities. ACL'13
- R. Huang and E. Riloff. Inducing domain-specific semantic class taggers from (almost) nothing. ACL'10
- X. Ling and D. S. Weld. Fine-grained entity recognition. AAAI'12
- W. Shen, J. Wang, P. Luo, and M. Wang. A graph-based approach for ontology population with named entities. CIKM'12
- S. Gupta and C. D. Manning. Improved pattern learning for bootstrapped entity extraction. CONLL'14
- P. P. Talukdar and F. Pereira. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. ACL'10
- Z. Kozareva and E. Hovy. Not all seeds are equal: Measuring the quality of text mining seeds. NAACL'10
- L. Gal´arraga, G. Heitz, K. Murphy, and F. M. Suchanek. Canonicalizing open knowledge bases. CIKM'14