Data-Driven Behavioral Analytics:
Observations, Representations and Models

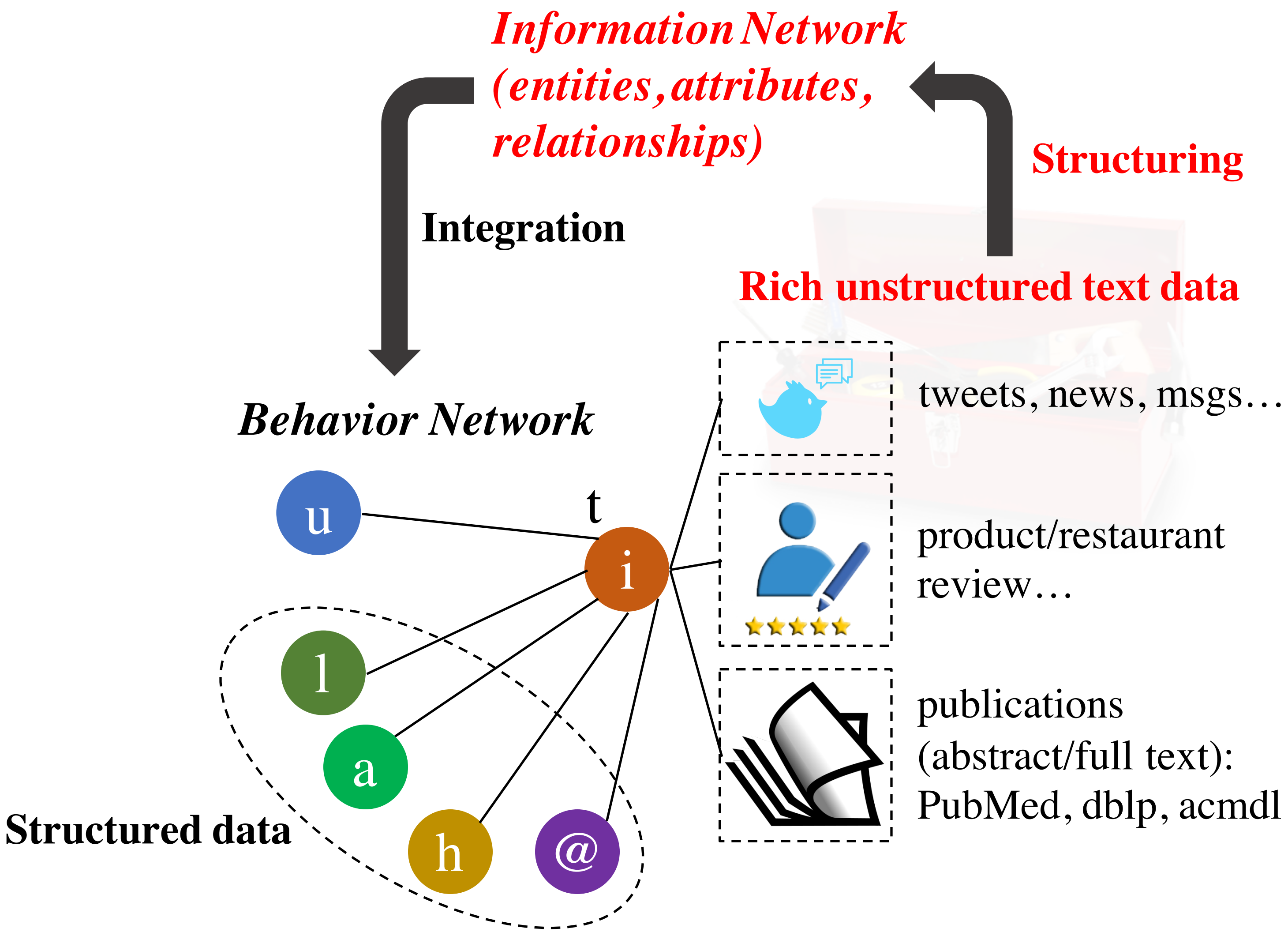
Meng Jiang (UIUC)
Peng Cui (Tsinghua)
Jiawei Han (UIUC)

http://www.meng-jiang.com/tutorial-cikm16.html

Tutorial in CIKM 2016, October 24, Indianapolis, IN

II. Structuring behavioral content and integrating behavioral analysis with information networks

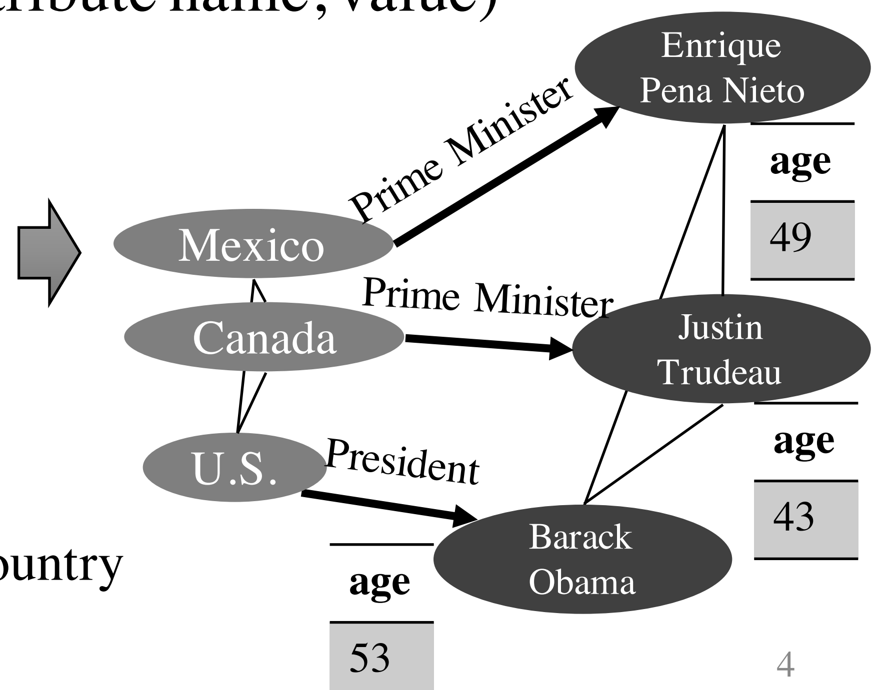
Data to Network to Knowledge

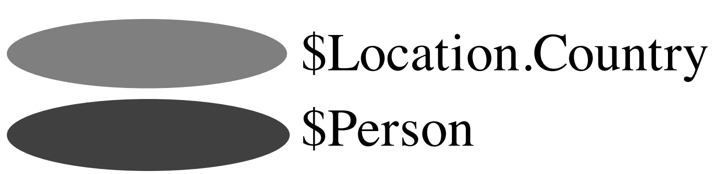


Construction of Heterogeneous Information Networks from Text

- Philosophy: Not extensive "labeling" but exploring the power of massive text corpora!
  - Mining phrases (the minimal semantic units)
  - Entity recognition and typing
  - Attribute discovery (entity, attribute name, value)

…here by **Canada Prime Minister Justin Trudeau, 43**, the so-called #APEChottie…of **Mexico's Enrique Pena Nieto, 49**, … **United States President Barack Obama, 53**, who…

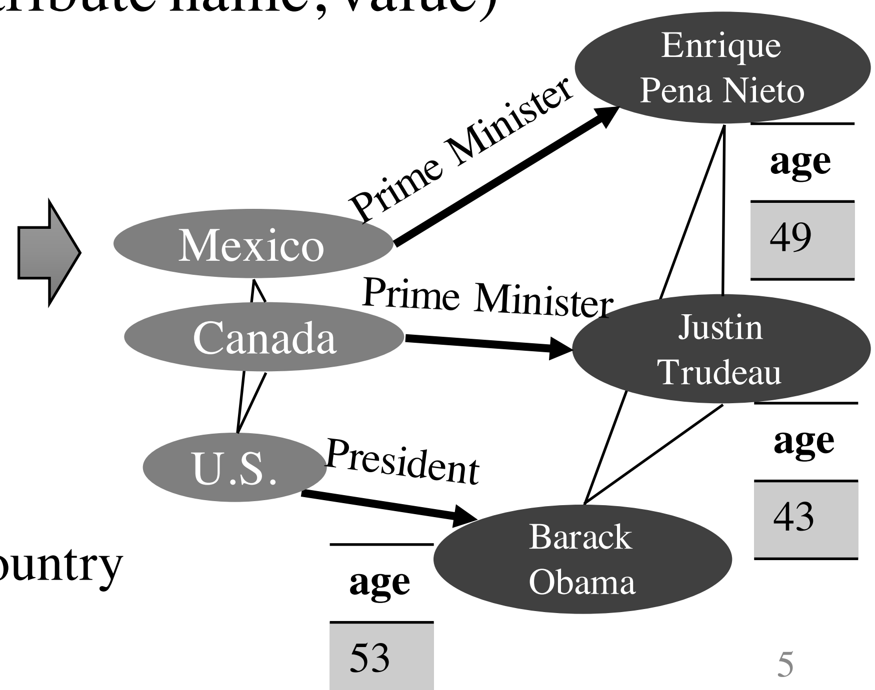


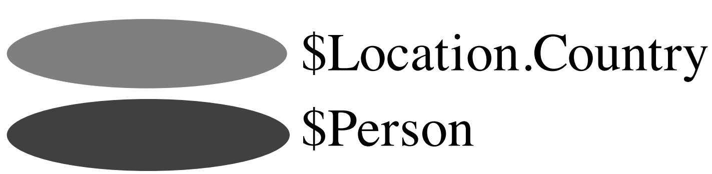


Construction of Heterogeneous Information Networks from Text

- Philosophy: Not extensive "labeling" but exploring the power of massive text corpora!
  - **Mining phrases (the minimal semantic units)**
  - Entity recognition and typing
  - Attribute discovery (entity, attribute name, value)

…here by **Canada Prime Minister Justin Trudeau, 43**, the so-called #APEChottie…of **Mexico's Enrique Pena Nieto, 49**, … **United States President Barack Obama, 53**, who…





Why Mining Phrases?

- **Unigrams** are *ambiguous but* **phrases** are natural, *unambiguous* semantic units
  - Ex.: “United” vs. United States, United Airline, United Parcel Service
- Mining semantically meaningful phrases
  - Transform text data from *word granularity* to *phrase granularity*
  - Enhance the power at manipulating unstructured data using information networks
- Phrase mining: Most NLP methods may need annotation and training
  - Annotate hundreds of documents as training data
  - Train a supervised model based on part-of-speech features
  - Limitations: High annotation cost
  - May not be scalable to domain-specific, dynamic, emerging applications
    - Scientific domains, query logs, or social media, e.g., Yelp, Twitter
- Minimal/no training but making good use of massing corpora

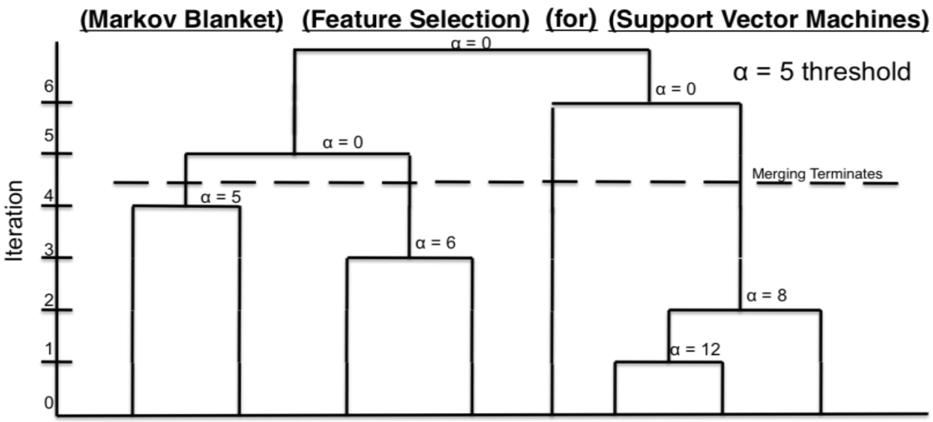
Strategies for Phrase Mining

- Strategy 1: Simultaneously inferring phrases and topics
  - Bigram topical model [Wallach'06], topical n-gram model [Wang, et al.'07], phrase discovering topic model [Lindsey, et al.'12]
  - High model complexity: Tends to overfitting; High inference cost: Slow
- Strategy 2: Post topic modeling phrase construction
  - Label topic [Mei et al.'07], TurboTopic [Blei & Lafferty'09], KERT [Danilevsky, et al.'14]
  - Words in the same phrase may be assigned to different topics
    - Ex. … knowledge discovery using least squares support vector machine …
- **Our solution 1: ToPMine** [El-kishky, et al., VLDB'15]
  - First Phrase Mining then Topic Modeling (No training data at all)
- **Our solution 2: SegPhrase+** [Liu, et al., SIGMOD'15]
  - Integrating phrase mining and document segmentation (with minimal training data)

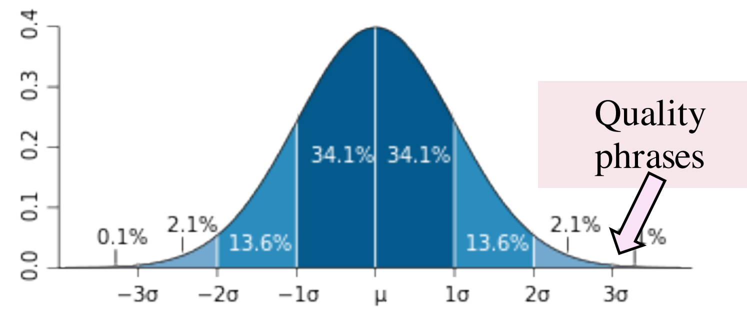
ToPMine: The Overall Phrase Mining Framework

- **ToPMine** [El-Kishky et al. VLDB'15]
  - First phrase construction, then topic mining
  - Contrast with KERT: First 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

Phrase Mining: Frequent Pattern Mining + Statistical Analysis



Markov Blanket Feature Selection for Support Vector Machines.



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

$$\alpha(P_1, P_2) \approx (f(P_1 \bullet P_2) - \mu_0(P_1, P_2)) \,/\, f(P_1 \bullet P_2)^{1/2}$$

[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

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"

ToPMine: Experiments on Yelp Reviews

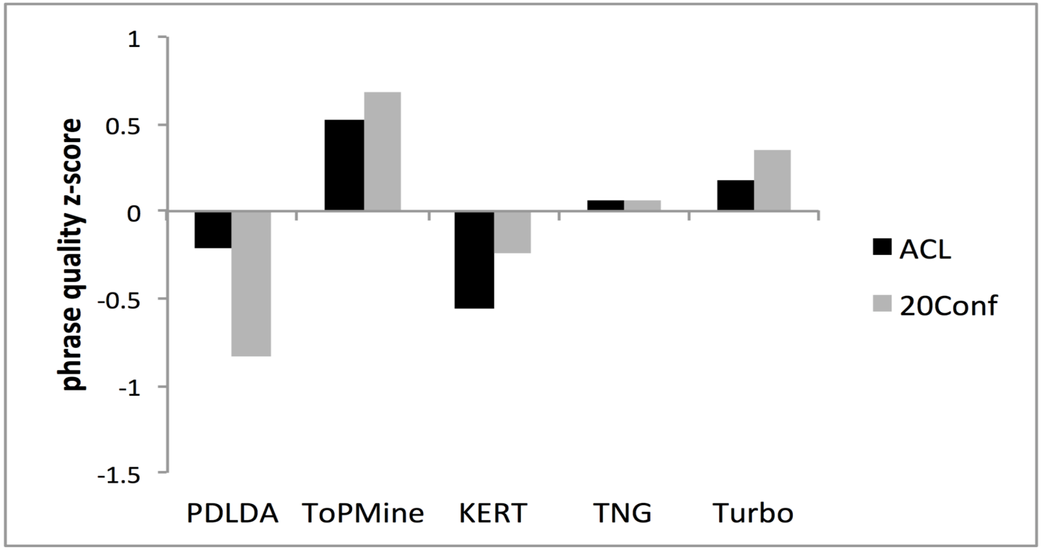
	Topic 1	*Topic 2*	*Topic 3*	*Topic 4*	*Topic 5*
unigrams	coffee	food	room	store	good
	ice	good	parking	shop	food
	cream	place	hotel	prices	place
	flavor	ordered	stay	find	burger
	egg	chicken	time	place	ordered
	chocolate	roll	nice	buy	fries
	breakfast	sushi	place	selection	chicken
	tea	restaurant	great	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

ToPMine: Faster and Generating Better Quality Phrases

Running time of different algorithms

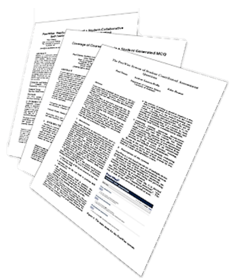
Method	*sampled dblp titles (k=5)*	*dblp titles (k=30)*	*sampled dblp abstracts*	*dblp abstracts*
PDLDA	3.72(hrs)	~20.44(days)	1.12(days)	~95.9(days)
Turbo Topics	6.68(hrs)	>30(days)*	>10(days)*	>50(days)*
TNG	146(s)	5.57 (hrs)	853(s)	NA†
LDA	**65(s)**	3.04 (hrs)	353(s)	13.84(hours)
KERT	68(s)	3.08(hrs)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

Phrase quality measured by z-score



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

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.

+ A small set of labels or a general KB

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

Phrase Mining **Phrasal Segmentation**

Integrating phrase mining with phrasal segmentation

J. Liu et al. **Mining Quality Phrases from Massive Text Corpora.** SIGMOD, 2015 (**won Grand Prize in Yelp Dataset Challenge**).

Experiments: 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

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

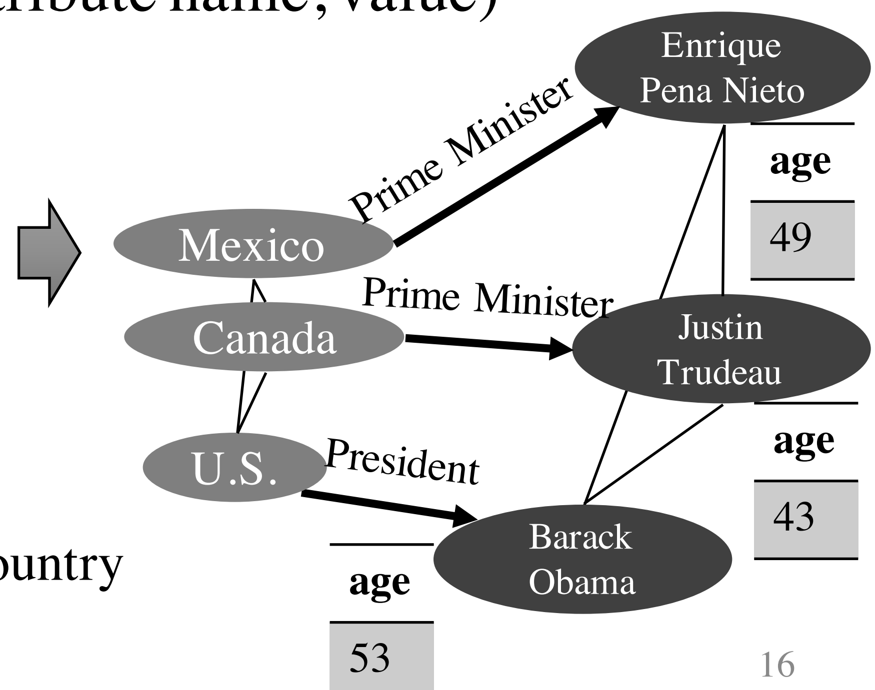
Rank	Phrase	In English
…	…	…
62	首席_执行官	CEO
63	中间_偏右	Middle-right
…	…	…
84	百度_百科	Baidu Pedia
85	热带_气旋	Tropical cyclone
86	中国科学院_院士	Fellow of Chinese Academy of Sciences
…	…	…
1001	十大_中文_金曲	Top-10 Chinese Songs
1002	全球_资讯网	Global Info Website
1003	天一阁_藏_明代_科举_录_选刊	A Chinese book name
…	…	…
9934	国家_戏剧_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

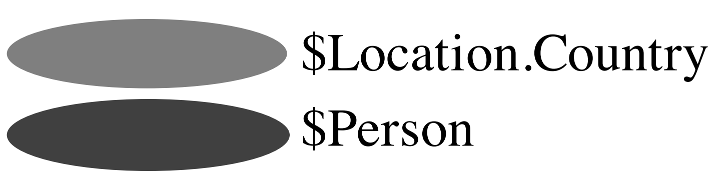
Construction of Heterogeneous Information Networks from Text

Philosophy: Not extensive "labeling" but exploring the power of massive text corpora!

- Mining phrases (the minimal semantic units)
- **Entity recognition and typing**
- Attribute discovery (entity, attribute name, value)

…here by **Canada Prime Minister Justin Trudeau, 43**, the so-called #APEChottie…of **Mexico's Enrique Pena Nieto, 49**, … **United States President Barack Obama, 53**, who…





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. …

Plain text

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. …

Text with typed entities

FOOD

LOCATION

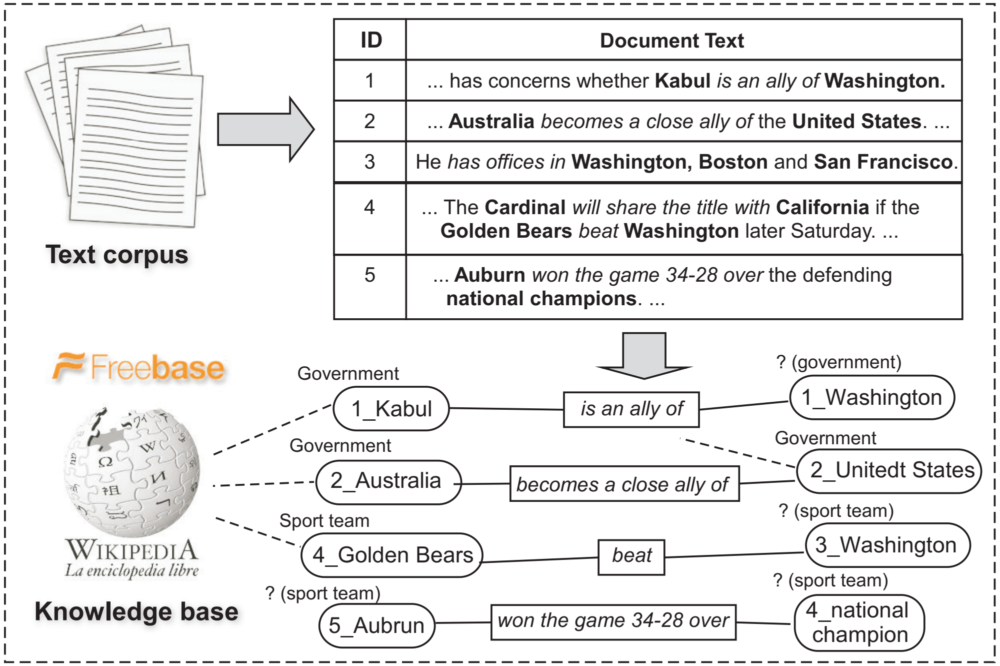
EVENT

Enabling structured analysis of unstructured text corpus

ClusType: A Distant Supervision Framework

Problem: *Distantly-supervised entity recognition in a domain-specific corpus*

- ❑ Given: (1) a domain-specific corpus *D*, (2) a knowledge base (e.g., Freebase), (3) a set of target types (*T*) from a KB
- ❑ Detect candidate entity mentions in D, and categorize each candidate mention by target types or Not-Of-Interest (NOI)



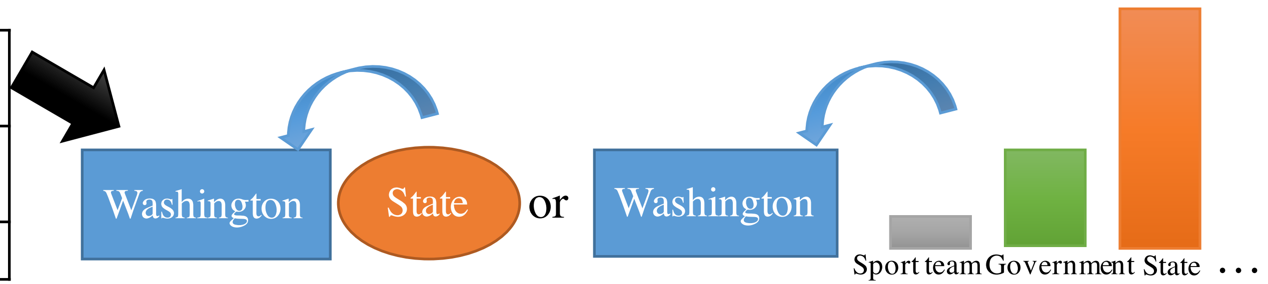
Solution:

- ❑ Detect entity mentions from text
- ❑ Map candidate mentions to KB entities of target types
- ❑ Use confidently mapped {mention, type} to infer types of remaining candidate mentions

Entity Recognition and Typing: Challenges and Solutions

- Challenge 1: Domain Restriction: Extensive training, use general-domain corpora, not work well on specific, dynamic or emerging domains (*e.g.*, tweets, Yelp reviews)
  - Solution: Domain-agnostic phrase mining: Extracts candidate entity mentions with minimal linguistic assumption (e.g., only use POS tagging)
- Challenge 2: Name ambiguity: Multiple entities may share the same surface name
  - Solution: Model each mention based on its surface name and context

While Griffin is not the part of Washington's plan on Sunday's game, …
… news from Washington indicates that the congress is going to…
It is one of the best state parks in Washington.



- Challenge 3: Context Sparsity: There are many ways to describe the same relation
  - Solution: cluster *relation phrase*, infer synonymous relation phrases

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

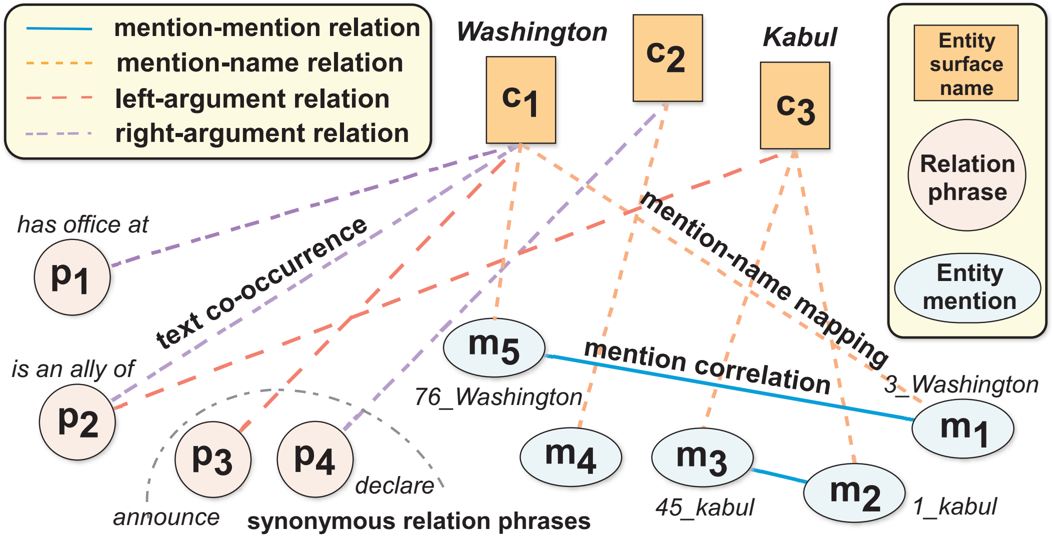
The ClusType Framework: Phrase Segmentation and Heterogeneous Graph Construction

- **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

Multi-view relation phrase clustering

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

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

Candidate Generation

- Phrase mining 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
  - Extend ToPMine to partition corpus into segments which meet both significance threshold and POS patterns → candidate entity mentions & relation phrases

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.

Experiment: Entity detection: Performance comparison between our method and 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)

Type Inference: A Joint Optimization Problem

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

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

Type propagation between entity surface names and relation phrases

Mention modeling & mention correlation

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

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

Multi-view relation phrases clustering)

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 pattern-based learning; **SemTagger**:bootstrapping method which trains contextual classifier based on seed mentions; **FIGER**: distantly-supervised sequence labeling method trained on Wiki corpus; **NNPLB**: label propagation using ReVerb assertion and seed mention; **APOLLO**: mention-level label propagation using Wiki concepts and KB entities;
  - **ClusType-NoWm**: ignore mention correlation; **ClusType-NoClus**: conducts only type propagation; **ClusType-TwpStep**: first performs hard clustering then type propagation

Comparing ClusType with Other Methods and Its Variants

Performance comparison on three datasets in terms of Precision, Recall and F1 score

Table 5: Performance comparisons on three datasets in terms of Precision, Recall and F1 score.

Data sets	**NYT**			**Yelp**			**Tweet**		
Method	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Pattern [9]	0.4576	0.2247	0.3014	0.3790	0.1354	0.1996	0.2107	0.2368	0.2230
FIGER [16]	0.8668	0.8964	0.8814	0.5010	0.1237	0.1983	**0.7354**	0.1951	0.3084
SemTagger [12]	0.8667	0.2658	0.4069	0.3769	0.2440	0.2963	0.4225	0.1632	0.2355
APOLLO [29]	0.9257	0.6972	0.7954	0.3534	0.2366	0.2834	0.1471	0.2635	0.1883
NNPLB [15]	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**

- **vs. FIGER**: Effectiveness of our candidate generation and type propagation
- vs. **NNPLB** and **APOLLO**: ClusType utilizes not only semantic-rich relation phrase as type cues, but also cluster synonymous relation phrases to tackle context sparsity
- vs. our **variants**: (i) models mention correlation for name disambiguation; and (ii) integrates clustering in a mutually enhancing way

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

Table 6: F1 score comparison with trained NER.

Method	NYT	Yelp	Tweet
Stanford NER [6]	0.6819	0.2403	0.4383
ClusType-NoClus	0.9031	0.4522	0.4167
ClusType	**0.9419**	**0.5943**	**0.4717**

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

- 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

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**

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

Fine-grained Entity Typing

- **Fine-grained Entity Typing**: Type labels for a mention forms a "*type-path*" (not necessarily ending in a leaf node) in a given (tree-structured) type hierarchy

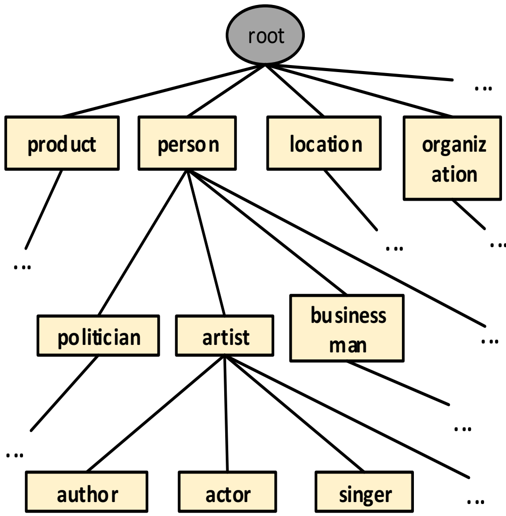
ID	Sentence
S1	Republican presidential candidate ***Donald Trump*** spoke during a campaign event in Rock Hill.
S2	***Donald Trump***'s company has threatened to withhold up to $1 billion of investment if the U.K. government decides to ban his entry into the country.
S3	In ***Trump***'s TV reality show, "The Apprentice", 16 people competed for a job.
...	...

Type-path

Person → politician

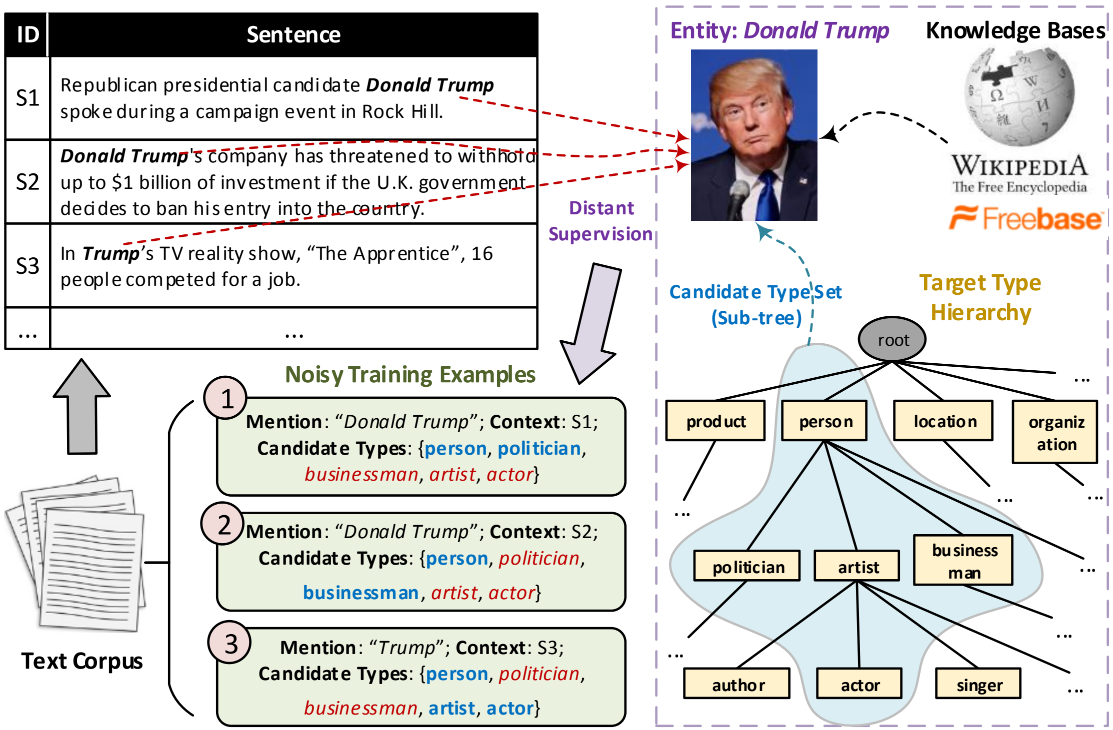
Person → businessman

Person → artist → actor



- Manually annotating training corpora with **100+** entity types
  - Expensive & Error-prone
- **Current practice**: use distant supervision to ***automatically labeled training corpora***

Label Noise in Entity Typing



Donald Trump is mentioned is sentences S1-S3.

- Distant supervision
  - Assign *same* types (blue region) to *all* the mentions
  - Does not consider ***local contexts*** when assigning type labels
  - Introduce ***label noise*** to the mentions

The types assigned to entity Trump include person, artist, actor, politician, businessman, while only {person, politician} are correct types for the **mention "*Trump*" in S1**

Label Noise in Entity Typing (cont.)

- Current typing systems either **ignore this issue**
  - assume all candidate labels obtained by supervision are "true" labels

Dataset	Wiki	OntoNotes	BBN	NYT
# of target types	113	89	47	446
(1) noisy mentions (%)	27.99	25.94	22.32	51.81
(2a) sibling pruning (%)	23.92	16.09	22.32	39.26
(2b) min. pruning (%)	28.22	8.09	3.27	32.75
(2c) all pruning (%)	45.99	23.45	25.33	61.12

- Or use **simple pruning heuristics** to ***delete*** mentions with conflicting types
  - aggressive deletion of mentions → significant loss of training data

The larger the target type set, the more severe the loss!

Label Noise Reduction: Task Description

- Define a *new* task, called **Label Noise Reduction in Entity Typing,** to identify the correct type-path for *each mention in training set*, from its *noisy candidate type set*
  - **VS. typical typing systems**: they focus on designing models for typing *unlabeled mentions*
  - The first systematic study of type label noise in distant supervision
  - A fundamental task for entity typing systems (the bottleneck of their performance)
- **Problem Definition**
  - **Input**:
    - (1) Automatically labeled training corpus: set of (mention, context, candidate type labels) triples
    - (2) Knowledge base, along with its entity-type facts (i.e., set of (entity, type) tuples)
    - (3) Target type hierarchy **T**
  - **Output**: Estimate ***a single type-path*** (not required to end in a leaf node) in the hierarchy **T**, based on the mention itself as well as its context in the sentence
- **Non-goals**: Entity mention detection; Entity linking; Type hierarchy creation

Label Noise Reduction: Challenges

Presence of incorrect type labels in a mention's candidate type set

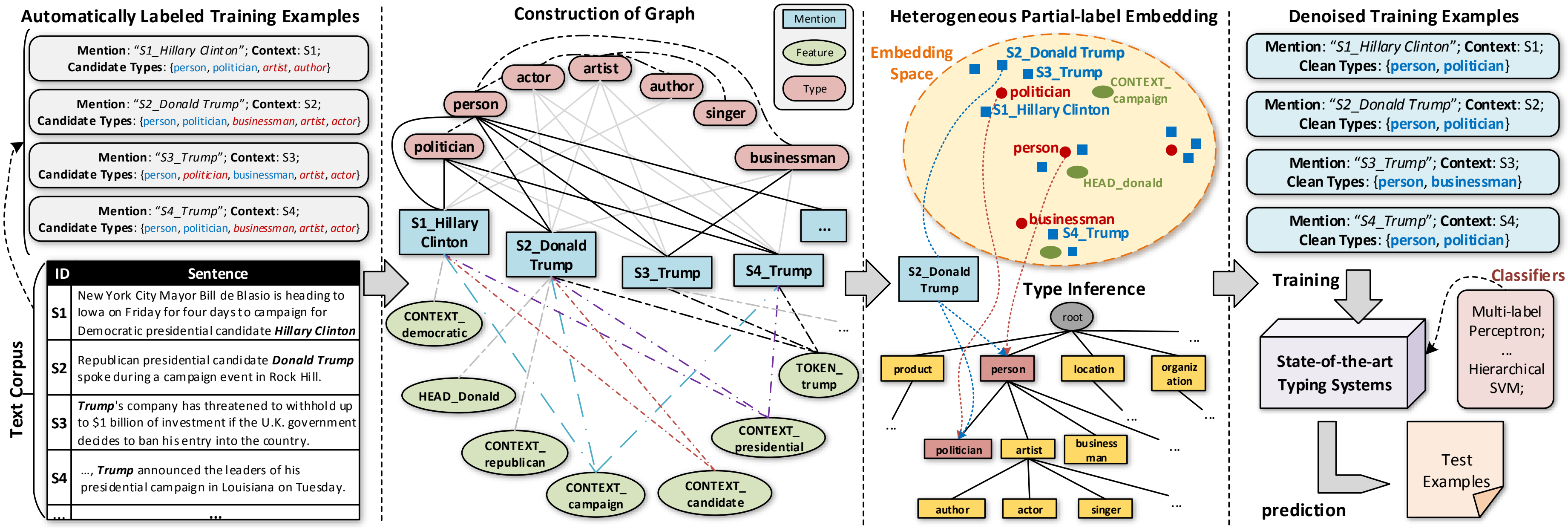
- Supervised/semi-supervised techniques both assume "*all labels are correct/reliable labels*"
- How to accurately estimate the relatedness between mentions and types?
  - **Aspect I**: How to model the *noisy associations between mention and its candidate labels*, to indicate the "truth status" of the candidate labels
  - **Aspect II**: How to incorporate the *semantic similarity between types*, as we are estimating the type-path holistically for a mention
    - vs. estimating individual labels independently

Label Noise Reduction: Solution Ideas

- Propose a weakly-supervised (unsupervised) approach, where the end goal is to estimate the *relatedness between mentions and types*
  1. sim(mention, **true** candidate label) > sim(mention, **false** candidate label)
  2. sim(mention, **fine-grained** true label) > sim(mention, **coarse-grained** true label)

1. Model the "truth status" of candidate labels as "latent values" using a novel ***partial-label loss*** → progressively estimate them by incorporating multiple signals:
   - *Co-occurrences between text features and mentions* in the large corpus
   - *Collective associations between type labels and mentions* in the large corpus
2. Model *semantic similarity between types* (*i.e.*, type correlation) derived from KB, to ensure holistic type-path estimation

Label Noise Reduction: Framework Overview



1. Generate text features and construct a heterogeneous graph
2. Perform joint embedding of the constructed graph G into the same low-dimensional space
3. For each mention, search its candidate type sub-tree in a top-down manner and estimate the true type-path from learned embedding

Text Features for Fine-grained Typing

- Features are extracted from:
  - (1) mention's name string: *e.g., head token, POS tags, Brown Cluster of head token*
  - (2) mention's context in the sentence: *e.g., n-grams, dependency roles*

Feature	Description	Example
Head	Syntactic head token of the mention	"HEAD_Turing"
Token	Tokens in the mention	"Turing", "Machine"
POS	Part-of-Speech tag of tokens in the mention	"NN"
Character	All character trigrams in the head of the mention	":tu", "tur", ..., "ng:"
Word Shape	Word shape of the tokens in the mention	"Aa" for "Turing"
Length	Number of tokens in the mention	"2"
Context	Unigrams/bigrams before and after the mention	"CXT_B:Maserati ,", "CXT_A:and the"
Brown Cluster	Brown cluster ID for the head token (learned using $\mathcal{D}$)	"4_1100", "8_1101111", "12_111011111111"
Dependency	Stanford syntactic dependency [16] associated with the head token	"GOV:nn", "GOV:turing"

- "*Turing Machine*" is used as an example mention from the sentence:
  - "The band's former drummer Jerry Fuchs—who was also a member of Maserati, Turing Machine and The Juan MacLean—died after falling down an elevator shaft.".

Construction of Heterogeneous Graphs

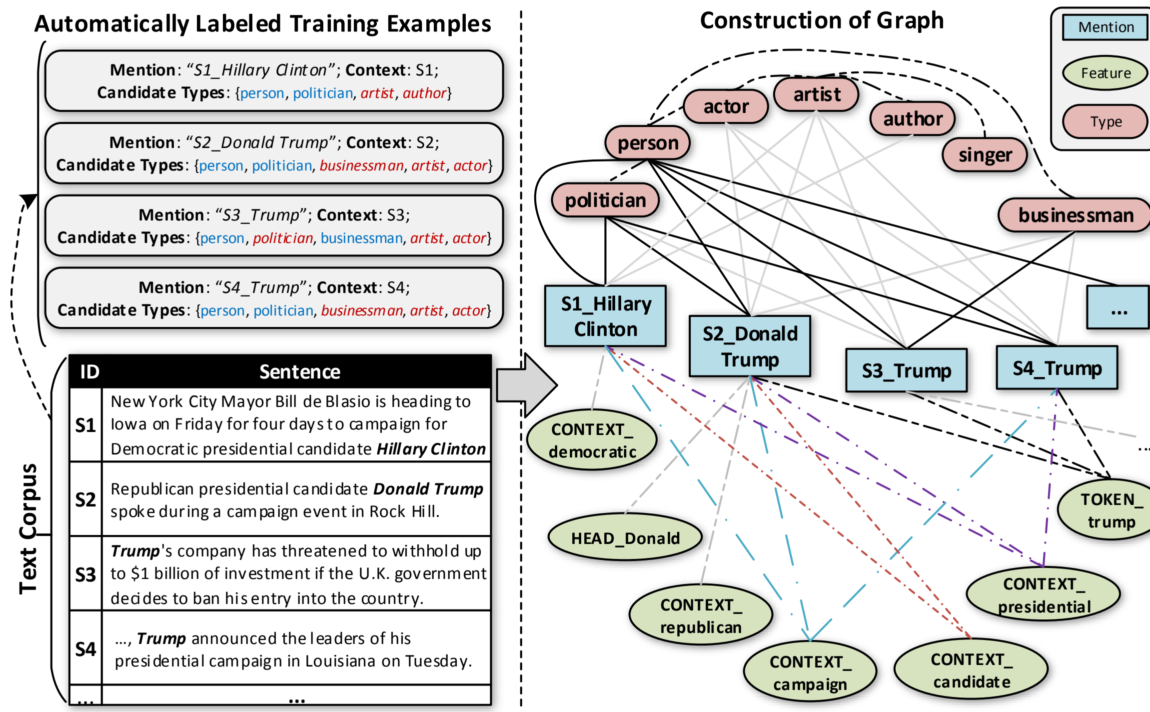
- With three types of objects extracted from corpus: entity mentions, target types, and text features

Three types of links:

1. **Mention-type link**: represents each mention's candidate type assignment

2. **Mention-feature link**: captures *corpus-level co-occurrences* between mentions and text features

3. **Type correlation link**: encodes the type correlation derived from KB or target type hierarchy



Mention-Type Association Subgraph

- Forms a bipartite graph between entity mentions and target types
  - Each mention is linked to its candidate types with binary weight
  - Some links are "false" links in the constructed mention-type subgraph
  - The likelihood of a mention-type link is measured by the relevance between the corresponding mention and type

Example: In sentence S1, context words *democratic* and *presidential* infer that type **politician** is more relevant than type **actor** for mention "Hillary Clinton"

Hypothesis 1 (Partial Label Association):
A mention should be embedded closer to its most relevant candidate type than to any other non-candidate type, yielding higher similarity between the corresponding embedding vectors.

ID	Sentence
S1	New York City Mayor Bill de Blasio is heading to Iowa on Friday for four days to campaign for Democratic presidential candidate ***Hillary Clinton***
S2	Republican presidential candidate ***Donald Trump*** spoke during a campaign event in Rock Hill.
S3	***Trump***'s company has threatened to withhold up to $1 billion of investment if the U.K. government decides to ban his entry into the country.
S4	…, ***Trump*** announced the leaders of his presidential campaign in Louisiana on Tuesday.

Mention-Feature Co-occurrence Subgraph

- Intuition
  - Mentions sharing many text features tend to have close type semantics
  - Text features which co-occur with many entity mentions in the corpus likely represent similar entity types.

Example: mentions "Donald Trump" in S2 and "Trump" in S4 share multiple features (e.g., *Trump*, *presidential* and *campaign*), and thus are likely of the same type **politician**. Conversely, features *campaign* and *presidential* likely represent the same type politician since they co-occur with similar sets of mentions in the corpus.

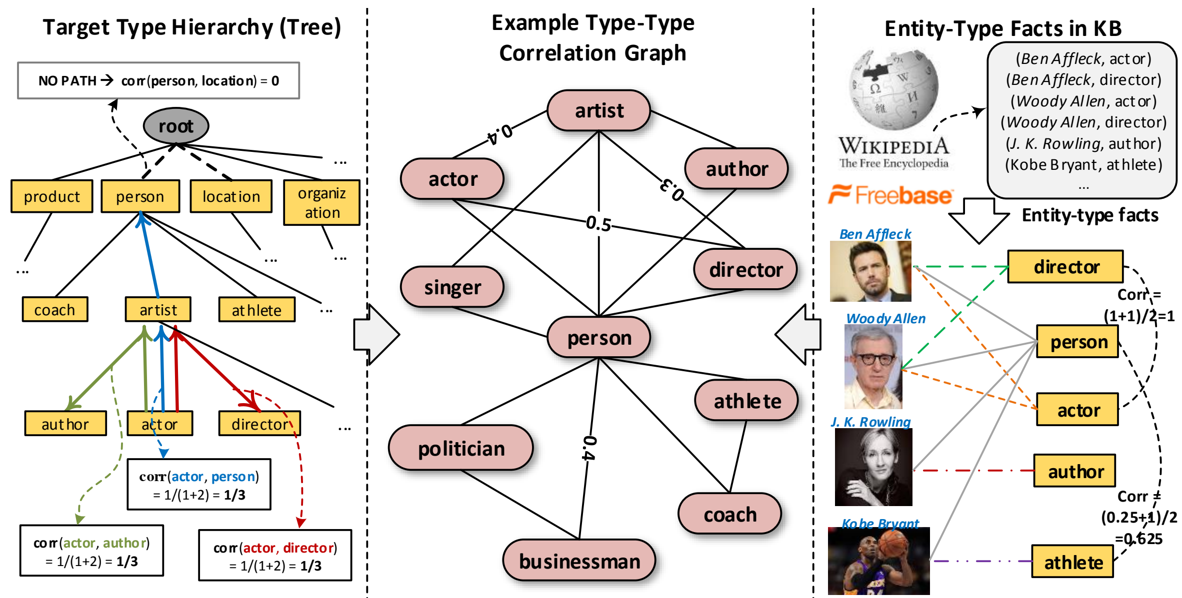
ID	Sentence
S1	New York City Mayor Bill de Blasio is heading to Iowa on Friday for four days to campaign for Democratic presidential candidate ***Hillary Clinton***
S2	Republican presidential candidate ***Donald Trump*** spoke during a campaign event in Rock Hill.
S3	***Trump***'s company has threatened to withhold up to $1 billion of investment if the U.K. government decides to ban his entry into the country.
S4	..., ***Trump*** announced the leaders of his presidential campaign in Louisiana on Tuesday.

Hypothesis 2 (Mention-Feature Co-occurrences):
If two entity mentions share similar features, they should be close to each other in the embedding space (i.e., high similarity score). If two features co-occur with a similar set of mentions, their embedding vectors tend to be similar.

Type Correlation Subgraph

- Build a homogeneous graph to represent the semantic similarity between types
  - **Simple way**: Use distance in the target type hierarchy
    - In target type hierarchy, types closer to each other tend to be more related
    - **Example:** actor is more related to artist than to person in the left column
  - **Advanced way**: Exploit entity-type facts in KB
    - Given two target types, the correlation between them is proportional to the number of entities they share in the KB

Hypothesis 3 (Type Correlation):
If high correlation exists between two target types based on either type hierarchy or KB, they should be embedded close to each other.



Heterogeneous Partial-Label Embedding (PLE): The Joint Optimization Problem

$$\min_{\{\mathbf{u}_i\}_{i=1}^N,\{\mathbf{c}_j\}_{j=1}^M,\{\mathbf{v}_k,\mathbf{v}'_k\}_{k=1}^K} \mathcal{O} = \mathcal{O}_{MY} + \mathcal{O}_{MF} + \mathcal{O}_{YY}$$

$$\mathcal{O}_{MY} = \sum_{i=1}^N \ell_i + \frac{\lambda}{2}\sum_{i=1}^N \|\mathbf{u}_i\|_2^2 + \frac{\lambda}{2}\sum_{k=1}^K \|\mathbf{v}_k\|_2^2.$$

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

Partial label loss between mentions and types (Hypo 1)

$$\mathcal{O}_{MF} = -\sum_{(m_i,f_j)\in G_{MF}} w_{ij}\cdot\log p(f_j|m_i)$$

Model mention-feature links using second-order skip-gram objective (Hypo 2)

$$\mathcal{O}_{YY} = -\sum_{(y_k,y_{k'})\in G_{YY}} w_{kk'}\Big[\log p(y_{k'}|y_k) + \log p(y_k|y_{k'})\Big]$$

Type correlation based on KB (Hypo 3)

PLE: Partial-Label Loss

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

- **Intuition**
  - For mention m_i, the maximum score associated with its candidate types $\mathcal{Y}_i$ is greater than the maximum score associated with any other non-candidate types $\overline{\mathcal{Y}_i}$, where the scores are measured using current embedding vectors.

- **vs. multi-label learning**
  - A large margin is enforced between *all* candidate types and non-candidate types without considering noisy types.

PLE: Second-Order Proximity Model

- Intuition
  - Nodes with similar distributions over neighbors are similar to each other
- Define the probability of feature f_j generated by mention m_i for each link (m_i, f_j) in the mention-feature subgraph as follows

$$p(f_j|m_i) = \exp(\mathbf{c}_j^T \mathbf{u}_i) / \sum_{f_{j'} \in \mathcal{F}} \exp(\mathbf{c}_{j'}^T \mathbf{u}_i)$$

- Enforce the conditional distribution specified by embeddings, i.e., $p(\cdot|m_i)$, to be close to the empirical distribution (i.e., link distribution of m_i over all features in the mention-feature subgraph)

Learning Algorithm for PLE

$$\min_{\{\mathbf{u}_i\}_{i=1}^N,\{\mathbf{c}_j\}_{j=1}^M,\{\mathbf{v}_k,\mathbf{v}'_k\}_{k=1}^K} \mathcal{O} = \mathcal{O}_{MY} + \mathcal{O}_{MF} + \mathcal{O}_{YY}$$

- Can be efficiently solved by alternative minimization algorithm based on block coordinate descent schema
- Algorithm complexity is linear to #links in the heterogeneous graph
- Mini-batch stochastic sub-gradient descent can also be applied for our problem

Algorithm 1: Model Learning of PLE

Input: $G = \{G_{MY}, G_{MF}, G_{YY}\}$, regularization parameter λ, learning rate α, number of negative samples Z

Output: entity mention embeddings $\{\mathbf{u}_i\}_{i=1}^N$, feature embeddings $\{\mathbf{c}_j\}_{j=1}^M$, type embeddings $\{\mathbf{v}_k\}_{k=1}^K$

```
Initialize: {u_i}, {c_j}, and {v_k} as random vectors
while O in Eq. (7) not converge do
    for each link in G_MF and G_YY do
        Draw Z negative links from noise distribution P_n(·)
    end
    for m_i ∈ M do
        u_i ← u_i − α · ∂O/∂u_i with ∂O/∂u_i defined in Eq. (9)
    end
    for f_j ∈ F do
        c_j ← c_j − α · ∂O/∂c_j using ∂O/∂c_j defined in Eq. (10)
    end
    for y_k ∈ Y do
        v_k ← v_k − α · ∂O/∂v_k based on ∂O/∂v_k in Eq. (11)
        v'_k ← v'_k − α · ∂O/∂v'_k using ∂O/∂v'_k in Eq. (12)
    end
end
```

Top-Down Type Inference

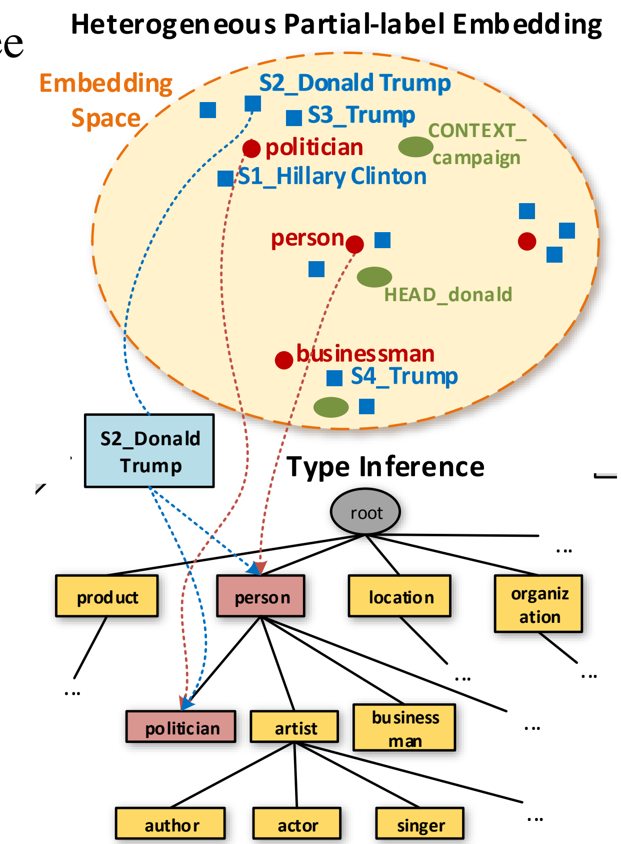
- Perform top-down search in the candidate type sub-tree to estimate the correct type-path

Algorithm 2: Type Inference

Input: candidate type sub-tree $\{\mathcal{Y}_i\}$, mention embeddings $\{\mathbf{u}_i\}$, type embeddings $\{\mathbf{v}_k\}$, threshold η

Output: estimated type-path $\{\mathcal{Y}_i^*\}$ for $m_i \in \mathcal{M}$

for $m_i \in \mathcal{M}$ **do**
  Initialize: $\mathcal{Y}_i^*$ as $\emptyset$, r as the root of $\mathcal{Y}$
  while $\mathcal{C}_i(r) \neq \emptyset$ **do**
    $r \leftarrow \mathrm{argmax}_{y_k \in \mathcal{C}_i(r)}\, s(\mathbf{u}_i, \mathbf{v}_k)$
    if $s(\mathbf{u}_i, \mathbf{v}_r) > \eta$ **then**
      Update the type-path: $\mathcal{Y}_i^* \leftarrow \mathcal{Y}_i^* \bigcup \{r\}$
    else
      return $\mathcal{Y}_i^*$ as the estimated type-path for m_i
    end
  end
end



Experiment Setting

- Datasets:
  - (1) **Wiki**: 1.5M sentences sampled from ~780k Wikipedia articles
  - (2) **OntoNotes**: 13,109 news
  - (3) **BBN**: 2,311 Wall Street Journal articles

Data sets	**Wiki**	**OntoNotes**	**BBN**
#Types	113	89	47
#Documents	780,549	13,109	2,311
#Sentences	1.51M	143,709	48,899
#Training mentions	2.69M	223,342	109,090
#Ground-truth mentions	563	9,604	121,001
#Features	644,860	215,642	125,637
#Edges in graph	87M	5.9M	2.9M

Experiment Setting

- Compared Methods
  - **(1) Sib**: removes siblings types; **(2) Min**: removes types that appear only once in the document; **(3) All**: first performs Sib pruning then Min pruning; (4)**DeepWalk**: embedding a homogeneous graph with binary edges**; (5) LINE**: second-order LINE; **(5) WSABIE**: adopts WARP loss with kernel extension; **(6) PTE**: applied PTE joint training algorithm on subgraphs G_{MF} and G_{MY}. **(7) PL-SVM**: uses a margin-based loss to handle label noise. **(8) CLPL**: uses a linear model to encourage large average scores for candidate types.
  - For PLE, we compare **(1)PLE**: adopts KB-based type correlation subgraph; **(2)PLE-CoH**: adopts type hierarchy-based correlation subgraph; **(3)PLE-NoCo**: does not consider type correlation.

Intrinsic Experiments: Effectiveness of Label Noise Reduction

- Goal: compare how accurately PLE and the other methods can estimate the true types of mentions from its noisy candidate type set

	Wiki							OntoNotes						
Method	Acc	Ma-P	Ma-R	Ma-F1	Mi-P	Mi-R	Mi-F1	Acc	Ma-P	Ma-R	Ma-F1	Mi-P	Mi-R	Mi-F1
Raw	0.373	0.558	**0.681**	0.614	0.521	**0.719**	0.605	0.480	0.671	**0.793**	0.727	0.576	**0.786**	0.665
Sib [7]	0.373	0.583	0.636	0.608	0.578	0.653	0.613	0.487	0.710	0.732	0.721	0.675	0.702	0.688
Min [7]	0.373	0.561	0.679	0.615	0.524	0.717	0.606	0.481	0.680	0.777	0.725	0.592	0.763	0.667
All [7]	0.373	0.585	0.634	0.608	0.581	0.651	0.614	0.487	0.716	0.724	0.720	0.686	0.691	0.689
DeepWalk-Raw [21]	0.328	0.598	0.459	0.519	0.595	0.367	0.454	0.441	0.625	0.708	0.664	0.598	0.683	0.638
LINE-Raw [29]	0.349	0.600	0.596	0.598	0.590	0.610	0.600	0.549	0.699	0.770	0.733	0.677	0.754	0.714
WSABIE-Raw [34]	0.332	0.554	0.609	0.580	0.557	0.633	0.592	0.482	0.686	0.743	0.713	0.667	0.721	0.693
PTE-Raw [28]	0.419	0.678	0.597	0.635	0.686	0.607	0.644	0.529	0.687	0.754	0.719	0.657	0.733	0.693
PLE-NoCo	0.556	0.795	0.678	0.732	0.804	0.668	0.730	0.593	0.768	0.773	0.770	0.751	0.762	0.756
PLE-CoH	0.568	0.805	0.671	0.732	0.808	0.704	0.752	0.620	0.789	0.785	0.787	0.778	0.769	0.773
PLE	**0.589**	**0.840**	0.675	**0.749**	**0.833**	0.705	**0.763**	**0.639**	**0.814**	0.782	**0.798**	**0.791**	0.766	**0.778**

40.57% improvement in Accuracy and 23.89% improvement in Macro-Precision compared to the best baseline on Wiki dataset

- **vs. pruning strategies**: LNR *identifies true types* from the candidate type sets instead of *aggressively deleting instances* with noisy type labels
- **vs. other embedding methods**: PLE obtains superior performance because it effectively *models the noisy type labels*
- **vs. PLE variants**: (i) PLE captures *type semantic similarity*; (ii) modeling type correlation with entity-type facts in KB yields more accurate and complete type correlation statistics than type hierarchy-based approach

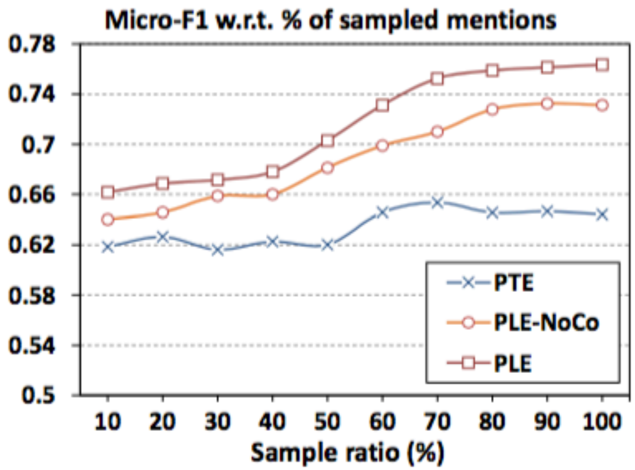
Intrinsic Experiments: Effectiveness of Label Noise Reduction

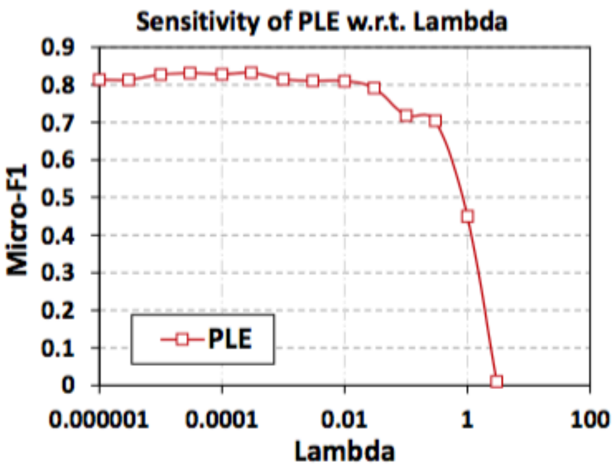
- Example output on news articles

Text	*NASA* says it may decide by tomorrow whether another space walk will be needed ...	... the board of *directors* which are composed of twelve members directly appointed by the *Queen*.
Wiki Page	https://en.wikipedia.org/wiki/NASA	https://en.wikipedia.org/wiki/Elizabeth_II
Cand. type set	person, artist, location, structure, organization, company, news_company	person, artist, actor, author, person_title, politician
WSABIE	person, artist	person, artist
PTE	organization, company, news_company	person, artist
PLE	organization, company	person, person_title

- PLE predicts fine-grained types with better accuracy (e.g., person_title)
- and avoids from overly-specific predictions (e.g., news_company)

Intrinsic Experiments: Effectiveness of Label Noise Reduction





- ❑ Testing the effect of training set size
  - ❑ Performance of all methods improves as the ratio increases, and becomes *insensitive* as the sampling ratio > 0.7
- ❑ Testing the effect of training set size
  - ❑ Performance of PLE becomes insensitive as becomes small enough (i.e., 0.01)

Extrinsic Experiments: Fine-Grained Entity Typing

- Compare performance gain of two state-of-the-art typing systems, when using denoised training data output by different compared methods

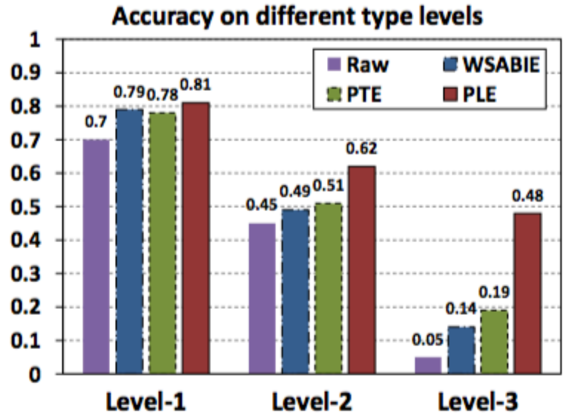
Typing System	Noise Reduction Method	Wiki Acc	Wiki Ma-F1	Wiki Mi-F1	OntoNotes Acc	OntoNotes Ma-F1	OntoNotes Mi-F1	BBN Acc	BBN Ma-F1	BBN Mi-F1
N/A	PL-SVM [20]	0.428	0.613	0.571	0.465	0.648	0.582	0.497	0.679	0.677
N/A	CLPL [2]	0.162	0.431	0.411	0.438	0.603	0.536	0.486	0.561	0.582
HYENA [35]	Raw	0.288	0.528	0.506	0.249	0.497	0.446	0.523	0.576	0.587
	Min [7]	0.325	0.566	0.536	0.295	0.523	0.470	0.524	0.582	0.595
	All [7]	0.417	0.591	0.545	0.305	0.552	0.495	0.495	0.563	0.568
	WSABIE-Min [34]	0.199	0.462	0.459	0.400	0.565	0.521	0.524	0.610	0.621
	PTE-Min [28]	0.238	0.542	0.522	0.452	0.626	0.572	0.545	0.639	0.650
	PLE-NoCo	0.517	0.672	0.634	0.496	0.658	0.603	0.650	0.709	0.703
	PLE	**0.543**	**0.695**	**0.681**	**0.546**	**0.692**	**0.625**	**0.692**	**0.731**	**0.732**
FIGER [14]	Raw	0.474	0.692	0.655	0.369	0.578	0.516	0.467	0.672	0.612
	Min	0.453	0.691	0.631	0.373	0.570	0.509	0.444	0.671	0.613
	All	0.453	0.648	0.582	0.400	0.618	0.548	0.461	0.636	0.583
	WSABIE-Min	0.455	0.646	0.601	0.425	0.603	0.546	0.481	0.671	0.618
	PTE-Min	0.476	0.670	0.635	0.494	0.675	0.618	0.513	0.674	0.657
	PLE-NoCo	0.543	0.726	0.705	0.547	0.699	0.639	0.643	0.753	0.721
	PLE	**0.599**	**0.763**	**0.749**	**0.572**	**0.715**	**0.661**	**0.685**	**0.777**	**0.750**

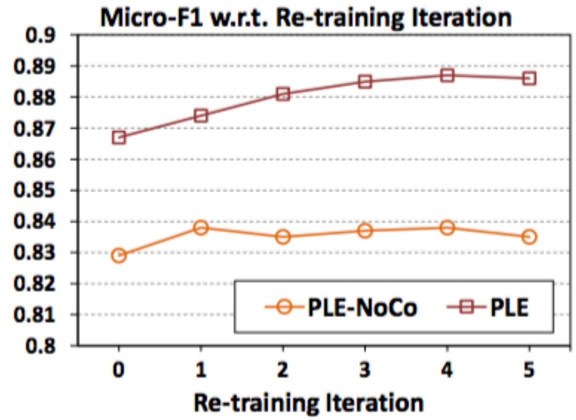
Table 9: Study of performance improvement on fine-grained typing systems **FIGER** [14] and **HYENA** [35] on the three datasets.

- **vs. other noise reduction methods**: the effectiveness of the proposed margin-based loss in modeling noisy candidate types
- **vs. partial-label learning methods**: PLE obtains superior performance because it jointly models type correlation derived from KB and feature-mention co-occurrences in the corpus

Case Analysis

- Testing at different type levels
  - It is more difficult to distinguish among deeper (more fine-grained) types.
  - PLE always outperforms the other two method, and achieves a 153% improvement in Accuracy.





- Iterative re-training of PLE
  - Analyze the effect of boostrapping PLE
  - The performance gain becomes marginal after 3 iterations of re-training

Acknowledgement

NSF
NSFC National Natural Science Foundation of China
TSINGHUA UNIVERSITY 1911
Carnegie Mellon University
UNIVERSITY OF MARYLAND 1856
Microsoft Research 微软亚洲研究院
Tencent 腾讯

References

D. Blei, A. Ng, and M. Jordan. "Latent dirichlet allocation." JMLR, 2003.

J. Herlocker, J. Konstan, L. Terveen, J. Riedl. "Evaluating collaborative filtering recommender systems." ACM TOIS, 2004.

Y. Koren, R. Bell, C. Volinsky. "Matrix factorization techniques for recommender systems." Computer, 2009.

Y. Koren. "Factorization meets the neighborhood: A multifaceted collaborative filtering model." KDD, 2008.

Y. Koren. "Collaborative filtering with temporal dynamics." CACM, 2010.

M. Balabanovic and Y. Shoham. "FAB: Content-based, collaborative recommendation." CACM, 1997.

N. Liu and Q. Yang. "Eigenrank: A ranking-oriented approach to collaborative filtering." SIGIR, 2008.

N. Liu, M. Zhao, and Q. Yang. "Probabilistic latent preference analysis for collaborative filtering." CIKM, 2009.

References

H. Ma, H. Yang, M. Lyu, and I. King. "Sorec: Social recommendation using probabilistic matrix factorization." CIKM, 2008.

H. Ma, T. Zhou, M. Lyu, and I. King. "Improving recommender systems by incorporating social contextual information." ACM TOIS, 2011.

H. Ma, D. Zhou, C. Liu, M. Lyu, and I. King. "Recommender systems with social regularization." WSDM, 2011.

J. Leskovec, A. Singh, and J. Kleinberg. "Patterns of influence in a recommendation network." PAKDD, 2006.

P. Massa and A. Paolo. "Trust-aware recommender systems." RecSys, 2007.

M. Jamali and E. Martin. "TrustWalker: A random walk model for combining trust-based and item-based recommendation." KDD, 2009.

H. Ma, I. King, and M. Lyu. "Learning to recommend with social trust ensemble." SIGIR, 2009.

H. Ma. I. King, and M. Lyu. "Learning to recommend with explicit and implicit social relations." ACM TIST, 2011.

References

M. Faloutsos, P. Faloutsos, and C. Faloutsos. "On power-law relationships of the internet topology." SIGCOMM, 1999.

A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Weiner. "Graph structure in the web." Computer Networks, 2000.

F. Chung and L. Lu. "The average distances in random graphs with given expected degrees." PNAS, 2002.

J. Kleinberg. "Authoritative sources in a hyperlinked environment." JACM, 1999.

H. Kwak, C. Lee, H. Park, and S. Moon. "What is Twitter, a social network or a news media?" WWW, 2010.

B. Hooi, H.A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos. "Fraudar: Bounding graph fraud in the face of camouflage." KDD, 2016.

C. Aggarwal and J. Han. "Frequent pattern mining." Springer, 2014.

J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M. Hsu. "FreeSpan: Frequent pattern-projected sequential pattern mining." KDD, 2000.

References

X. Yan and J. Han. "gspan: Graph-based substructure pattern mining." ICDM, 2003.

X. Yan and J. Han. "CloseGraph: Mining closed frequent graph patterns." KDD, 2003.

Y. Sun, J. Han, X. Yan, P.S. Yu, and T. Wu. "PathSim: Meta path-based top-k similarity search in heterogeneous information networks." VLDB, 2011.

Y. Sun, Y. Yu, and J. Han. "Ranking-based clustering of heterogeneous information networks with star network schema." KDD, 2009.

Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu. "RankClus: Integrating clustering with ranking for heterogeneous information network analysis." EDBT, 2009.

Y. Sun, R. Barber, M. Gupta, C. Aggarwar, and J. Han. "Co-author relationship prediction in heterogeneous bibliographic networks." ASONAM, 2011.

A. El-Kishky, Y. Song, C. Wang, C.R. Voss, and J. Han. "Scalable topical phrase mining from text corpora." VLDB, 2014.

J. Liu, J. Shang, C. Wang, X. Ren, and J. Han. "Mining quality phrases from massive text corpora." SIGMOD, 2015.

References

X. Ren, A. El-Kishky, C. Wang, F. Tao, C.R. Voss, and J. Han. “Effective entity recognition and typing by relation phrase-based clustering.” KDD, 2015.

X. Ren, W. He, M. Qu, C.R. Voss, H. Ji, and J. Han. “Label noise reduction in entity typing by heterogeneous partial-label embedding.” KDD, 2016.

C. Wang, M. Danilevsky, N. Desai, Y. Zhang, P. Nguyen, T. Taula, and J. Han. “A phrase mining framework for recursive construction of a topical hierarchy.” KDD, 2013.

E.E. Papalexakis, C. Faloutsos, N.D. Sidiropoulos. “ParCube: Sparse parallelizable tensor decompositions.” PKDD, 2012.

D. Koutra, U. Kang, J. Vreeken, and C. Faloutsos. “VOG: Summarizing and understanding large graphs.” SDM, 2014.

R. Gupta, A. Halevy, X. Wang, S.E. Whang, and F. Wu. “Biperpedia: An ontology for search applications.” VLDB, 2014.

M. Yahya, S. Whang, R. Gupta, and A. Halevy. “ReNoun: Fact extraction for nominal attributes.” EMNLP, 2014.

A. Halevy, N. Noy, S. Sarawagi, S.E. Whang, and X. Yu. “Discovering structure in the universe of attribute names.” WWW, 2016.

References

Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han. "Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation." SIGMOD, 2014.

Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han. "A confidence-aware approach for truth discovery on long-tail data." VLDB, 2014.

F. Ma, Y. Li, Q. Li, M. Qiu, J. Gao, S. Zhi, L. Su, B. Zhao, H. Ji, and J. Han. "Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation." KDD, 2015.

Y. Li, J. Gao, C. Meng, Q. Li, L. Su, B. Zhao, W. Fan, and J. Han. "A survey on truth discovery." KDD Explorations Newsletter, 2016.

S. Zhi, B. Zhao, W. Tong, J. Gao, D. Yu, H. Ji, and J. Han. "Modeling truth existence in truth discovery." KDD, 2015.

S. Kumar, R. West, and J. Leskovec. "Disinformation on the Web: Impact, characteristics, and detection of Wikipedia hoaxes." WWW, 2016.

S. Kumar, F. Spezzano, and V.S. Subrahmanian. "Identifying malicious actors on social media." ASONAM, 2016. (tutorial)

Thank you!

Data-Driven Behavioral Analytics:
Observations, Representations and Models