Automatic Entity Recognition and Typing for Massive Text Corpora
—A Phrase and Network Mining Approach

Xiang Ren (UIUC), Ahmed El-Kishky (UIUC), Chi Wang (MSR), Jiawei Han (UIUC)
Outline

1. Introduction to entity recognition and typing
2. Entity recognition – overview and phrase mining approach
3. Entity typing – overview and network mining approach
4. Trends and research problems
Motivation of Entity Recognition and Typing

- Making sense of large text corpus

<table>
<thead>
<tr>
<th>words</th>
<th>organizations</th>
<th>persons</th>
<th>locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>cities 0.75</td>
<td>United States 0.4</td>
<td>Ray Nagin 0.2</td>
<td>New Orleans 0.1</td>
</tr>
<tr>
<td>storm 0.63</td>
<td>Red Cross 0.3</td>
<td>Mayor 0.1</td>
<td>Louisiana 0.05</td>
</tr>
<tr>
<td>residents 0.58</td>
<td>US government 0.1</td>
<td>President Bush 0.02</td>
<td>Washington DC 0.02</td>
</tr>
<tr>
<td>government 0.51</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>donate 0.44</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>red 0.31</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>death 0.3</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- topics

<table>
<thead>
<tr>
<th>Topic 1</th>
<th>government 0.3</th>
<th>response 0.2</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topic 2</td>
<td>city 0.2</td>
<td>new 0.1</td>
<td>orleans 0.05</td>
</tr>
<tr>
<td>Topic 3</td>
<td>donate 0.1</td>
<td>relief 0.05</td>
<td>help 0.02</td>
</tr>
</tbody>
</table>

- Criticism of government response to the hurricane ...

- corpus
Example: Linking Entities to Knowledge Base

The criticism consisted primarily of condemnations of mismanagement in response to Hurricane Katrina. Specifically, there was a delayed response to the flooding of New Orleans, Louisiana. New Orleans Mayor Ray Nagin was also criticized for failing to implement his evacuation plan.

Bush was criticized for not returning to Washington, D.C. from his vacation in Texas until after Wednesday afternoon. On the morning of August 28, the president telephoned Mayor Nagin to "plead" for a mandatory evacuation of New Orleans, and Nagin and Gov. Blanco decided to evacuate the city in response to that request.
The criticism consisted primarily of condemnations of mismanagement in response to Hurricane Katrina. Specifically, there was a delayed response to the flooding of New Orleans, Louisiana. New Orleans Mayor Ray Nagin was also criticized for failing to implement his evacuation plan.

Bush was criticized for not returning to Washington, D.C. from his vacation in Texas until after Wednesday afternoon. On the morning of August 28, the president telephoned Mayor Nagin to "plead" for a mandatory evacuation of New Orleans, and Nagin and Gov. Blanco decided to evacuate the city in response to that request.

"Entities" are what a large part of our knowledge is about
Motivation of Entity Recognition and Typing

- Organizing and exploring text data

The prevalence of unstructured text data

Structures are useful for knowledge discovery

Too expensive to be structured by human:
Automated & scalable

Vast majority of the CEOs expressed frustration over their organization’s inability to glean insights from available data
-- IBM study with 1500+ CEOs
Example: News Articles

- Every day, >90,000 news articles are produced
- Unstructured data: news content
- Extracted entities: persons, locations, organizations, ...
Every year, hundreds of thousands papers are published
- Unstructured data: paper text
- Loosely structured entities: authors, venues
- Extracted entities: scientific concepts (techniques, applications, datasets, ...)
What Power Can We Gain if More Structures are available?

- Structured database queries
- Information network analysis, …
**Structures Facilitate Heterogeneous Information Network Analysis**

- **Example:** DBLP: A Computer Science bibliographic database

<table>
<thead>
<tr>
<th>Knowledge hidden in DBLP Network</th>
<th>Mining Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who are the <strong>leading</strong> researchers on Web search?</td>
<td>Ranking</td>
</tr>
<tr>
<td>Who are the <strong>peer</strong> researchers of Jure Leskovec?</td>
<td>Similarity Search</td>
</tr>
<tr>
<td>Whom <strong>will</strong> Christos Faloutsos <strong>collaborate with</strong>?</td>
<td>Relationship Prediction</td>
</tr>
<tr>
<td>Which types of <strong>relationships</strong> are most <strong>influential</strong> for an author to decide her topics?</td>
<td>Relation Strength Learning</td>
</tr>
<tr>
<td>How was the field of Data Mining <strong>emerged</strong> or <strong>evolving</strong>?</td>
<td>Network Evolution</td>
</tr>
<tr>
<td>Which authors are <strong>rather different</strong> from his/her peers in IR?</td>
<td>Outlier/anomaly detection</td>
</tr>
</tbody>
</table>
What Is Entity Recognition and Typing (ER)

- **Identify** token spans of entity mentions in text, and **classify** them into predefined set of types of interest


[TNF alpha] is produced chiefly by activated [macrophages]
What Is Entity Recognition and Typing (ER)

- **Identify** token spans of entity mentions in text, and **classify** them into predefined set of types of interest.


  [TNF alpha] is produced chiefly by activated [macrophages].
Why Entity Recognition and Typing is Challenging

- Many entities may share the same surface name
  - Name ambiguity!
- An entity may have multiple surface name(s)
  - Barack Obama, Obama, President Obama, president, ...
- An entity may associate with multiple types
  - Person, Politician, US president, US congressman, ...
  - Type ambiguity!
- Entity may have grammatically informal name
  - “in-and-out”
- ...

...
Scenario I: Sequential Text As Input

- Process each document or text fragment one by one

Entity Recognition and Typing Module

- ER Web API
- Query Search Intent
- Question Answering
Scenario II: Large Text Corpus As Input

- Process large document collection(s) in a batch

Entity Recognition and Typing Module

Input Corpus

2015 news articles

Data Source 1

New York Times

Data Source N

Washington Post
Example: Business Intelligence

- Top 10 active *politicians* regarding healthcare issues?
- Influential *high-tech companies* in Silicon Valley?

<table>
<thead>
<tr>
<th>Type</th>
<th>Entity</th>
<th>Mention</th>
</tr>
</thead>
<tbody>
<tr>
<td>politician</td>
<td>Obama</td>
<td><em>Obama</em> says more than 6M signed up for health care...</td>
</tr>
<tr>
<td>high-tech company</td>
<td>Apple</td>
<td><em>Apple</em> leads in list of Silicon Valley's most-valuable brands...</td>
</tr>
</tbody>
</table>
Example: Knowledge-Base Population

- As the primitive step in identifying newly emerging entities from dynamic text corpora (e.g., news, microblogs, tweets)
Focus of This Tutorial: Large Corpora
Characteristics of Text Corpus

- General vs. specific domain
- News vs. scientific publications
- Good amount of labeled data vs. few (no) open labeled data

Tagged datasets for named entity recognition tasks

1. 1999 Information Extraction – Entity Recognition Evaluation
   Notes: This dataset is apparently in public domain.
2. MUC-3 and MUC-4 datasets
   Notes: This dataset is apparently in public domain.
   Notes: This dataset is a manual annotation of a subset of RCV1 (Reuters CoNLL site. The raw text of RCV1 documents must be requested from NIST.
4. Message Understanding Conference (MUC) 6
   Notes: Consult the LDC Web site for current pricing and usage agreement.
5. Message Understanding Conference (MUC) 6 Additional News Text
   Notes: Consult the LDC Web site for current pricing and usage agreement.
6. Message Understanding Conference (MUC) 7
   Notes: Consult the LDC Web site for current pricing and usage agreement.
The prime minister’s reaction was risky and foolish: he asked the Greek people to reject a proposal which, at the moment they voted on it, did not exist. The referendum supplied the result Mr Tsipras wanted but in many ways his position has deteriorated. His opportunistic manoeuvre infuriated almost every other European leader. The prospect of Grexit suddenly became more real.
1. Introduction to entity recognition and typing
2. Entity recognition – overview and phrase mining approach
3. Entity typing – overview and network mining approach
4. Trends and research problems
Entity Mention Detection

Entity mention detection seeks to identify *spans of tokens* in text for analysis in whether they align to certain pre-defined categories such as:
- names of people, organizations, locations, dishes, concepts, etc

*Barack Obama arrived this afternoon in Washington, D.C.*
*President Obama’s wife Michelle accompanied him*

To effectively detect these candidate, intuitively requires the underlying *grammatical structure of sentences* and answer such questions as:
- which words go together as phrases, subject and object of verbs/verb phrases, etc
- Fortunately this is extensively studied in NLP!
Full Sentence Parsing

- Partitioning sentences into grammatical text segments

Raw text sentence (string) | Full parse tree (grammatical analysis)

Full-text Parsing

Parsing segments input text sentences into parse trees. Noun Phrases indicate entity mention candidates

- Full syntax understanding

- Low accuracy
- Adapts poorly to new domains (Twitter)
- Computationally slow (Intractable on web-scale)
Inefficiencies of Full Parsing

1. Parsing yields low accuracy in identifying entity mentions
2. Parsing requires non-trivial training data – manually curated
3. Parsing adapts poorly to new domains (e.g. twitter, biomedical, yelp)
4. Parsing is computationally slow. Cannot be applied on web-scale data

Motivates a family of “shallow” entity detection techniques.
Alternatives to Full Parsing: Direct Detection of Entity Mentions

A. Supervised/Semi-supervised Entity Mention Detection

B. Unsupervised Entity Mention Detection

C. Weakly and Distantly Supervised Mention Detection
## Entity Mention Pipeline

- **Making sense of large text corpora**

---

<table>
<thead>
<tr>
<th>raw corpus</th>
<th>tokenization</th>
<th>POS tagging</th>
<th>candidate detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>donate</td>
<td>verb</td>
<td>United States</td>
<td></td>
</tr>
<tr>
<td>united</td>
<td>adjective</td>
<td>Red Cross</td>
<td></td>
</tr>
<tr>
<td>residents</td>
<td>noun</td>
<td>US Government</td>
<td></td>
</tr>
<tr>
<td>red</td>
<td>adjective</td>
<td>President Bush</td>
<td></td>
</tr>
<tr>
<td>cities</td>
<td>noun</td>
<td>New Orleans</td>
<td></td>
</tr>
<tr>
<td>government</td>
<td>noun</td>
<td>Washington DC.</td>
<td></td>
</tr>
<tr>
<td>death</td>
<td>noun</td>
<td>Louisiana</td>
<td></td>
</tr>
</tbody>
</table>

Segmentation → Part-of-speech tagging → Entity Mention Detection
Noun Phrase Chunking

1. Apply tokenization and part-of-speech tagging to each sentence
2. Search for noun phrase chunks.

Things to think about

- Not all phrases are useful for entity mentions
- Can other signals in addition to POS tags be helpful?
- Noun chunks often smaller than noun phrases
A. Supervised/Semi-supervised Entity Mention Detection

B. Unsupervised Entity Mention Detection

C. Weakly and Distantly Supervised Mention Detection
Assumptions

1. Unsupervised methods cannot possibly take into consideration the innumerable features, signals, and cues for entity mentions.

2. Training data for entity mentions is more expensive than POS tagging, but less so than full parsing.

- Training data consisting of chunked data can be used for supervised training of entity mention chunkers.

The I-O-B Representation

- (Inside – Outside – Beginning)
  - I – Denotes token inside of a chunk
  - O – Denotes tokens outside of a chunk
  - B – Denotes token at the beginning of a chunk

```
We  | saw  | the  | yellow | dog
PRP | VBD  | DT   | JJ     | NN
B-NP| O    | B-NP | I-NP   | I-NP
```
NP Chunkers as Classifiers

- Insights
  - Under the I-O-B Representation, each word should be tagged with its I-O-B label
  - Like POS Taggers, I-O-B taggers can be solved through standard classification methods such as Naïve Bayes or more sophisticated methods
Unigram Chunking

Given each word’s POS tag, one can directly classify each word to its IOB chunk.

- We: PRP B-NP
- saw: VBD O
- the: DT B-NP
- yellow: JJ I-NP
- dog: NN I-NP

Each word gets its “most likely” IOB tag.

IOB Accuracy: 93%  Precision: 80%  Recall: 87%  F1: 83%

Pretty good! Can we do better?
To improve beyond using only the current unigram in isolation of any context, we can look at higher order contexts.

This IOB chunk of a word chosen in consideration of tags of previous two

Results for Bigram Chunker

<table>
<thead>
<tr>
<th>Measure</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOB Accuracy</td>
<td>93%</td>
</tr>
<tr>
<td>Precision</td>
<td>82%</td>
</tr>
<tr>
<td>Recall</td>
<td>87%</td>
</tr>
<tr>
<td>F1</td>
<td>85%</td>
</tr>
</tbody>
</table>

Improvement over unigram
Classical/Non-sequential Classifiers

- These methods consider higher-order features to classify each word into its appropriate I-O-B tag. Any classifier can be used for this task including:
  - Support Vector Machines
  - Ensemble Methods
  - Naïve Bayes
  - Logistic Regression
  - etc
Support Vector Machine Chunking

- Weighted vote of 8 Support Vector Machines trained on 8 distinct chunk representations

Word: \( w_{i-2} \rightarrow w_{i-1} \rightarrow w_i \rightarrow w_{i+1} \)

POS: \( t_{i-2} \rightarrow t_{i-1} \rightarrow t_i \rightarrow t_{i+1} \)

Chunk: \( c_{i-2} \rightarrow c_{i-1} \rightarrow c_i \)

I-O-B Representation (4 variants)

Two Directions:
- Forward Parsing
- Backward Parsing

4 * 2 = 8

Cross validation to set weights
Joint Tagging & Chunking with Bigrams

Three separate models are learned.

1. **Contextual Language Model**
   - A smoothed bigram model learnt from the sequences of part-of-speech tags and chunk descriptors in a training corpus.

2. **Chunking Model**
   - Smoothed bigram model learnt from the sequences of part-of-speech tags corresponding to chunks in the training corpus.

3. **Lexical Probabilities**
   - Estimated using word frequencies, tag frequencies, word-per-tag frequencies (smoothing is performed for unseen categories)
Joint Tagging & Chunking with Bigrams

- Combines different knowledge sources to obtain corresponding POS Tags and Chunks
- Once all the LM’s have been learnt, they are combined into an Integrated LM
- Shows possible concatenations of lexical tags, syntactical units, and their transition probabilities / lexical probabilities
- Tagging/shallow parsing performed by using dynamic programming (Viterbi) to find the maximum probability sequence of states.
Maximum Entropy Classifier

- Maximum Entropy classifiers are based on the assumption that the probability distribution which best represents the current state of knowledge is the one with largest entropy.

- External Features
  - Current Word
  - POS tag of current word
  - Surrounding words
  - POS tags of surrounding words

- Model Generated Features
  - Chunk tags of previous words

\[
P(w|h) = \frac{1}{Z(h)} \cdot e^{\sum_i \lambda_i f_i(h,w)}
\]
Ranking Algorithms for Entity Mentions

Insight

- Reranking the top N hypotheses from a maximum-entropy tagger may improve recovery of entity boundaries from text corpora

Methodology

1. Use a state-of-the-art max-ent tagger to generate top N segmentations
2. Re-rank these segmentations using global features and proposed methods (boosting and voted perceptron)

Global Features

- May be tied to each candidate segmentation’s boundaries, Quotation marks, Number of uppercase words, etc.

Results for Precision/Recall/F-Measure

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max-Ent</td>
<td>84.4</td>
<td>86.3</td>
<td>85.3</td>
</tr>
<tr>
<td>Boosting</td>
<td>87.3(18.6)</td>
<td>87.9 (11.6)</td>
<td>87.6 (15.6)</td>
</tr>
<tr>
<td>Voted Perceptron</td>
<td>87.3(18.6)</td>
<td>88.6 (16.8)</td>
<td>87.9 (17.7)</td>
</tr>
</tbody>
</table>

Parenthesis indicate relative improvement in error rate.
Classifiers for Sequential Data Models

Moving forward from classical classifiers that use only features for classification, many state-of-the-art methods apply sequential data models to detect these temporal patterns.

Successfully applied to part-of-speech tagging, sequential data models posit that sequential observations are related to each other such as through a Markov process, in contrast to traditional models that assume independence.

In a Markov Model, hidden states and their transitions explain observations.
Hidden Markov Models for Mention Detection

- A HMM is a finite state automaton with stochastic transitions defined on states and observations
  - For state $s$, $p(s|s')$
  - For observation $o$, $p(o|s)$
- Markov Assumption, Stationary Assumption, and Output Independence Assumption
- The task resorts to inferring most likely latent states given observations (words)
Hidden Markov Models for Mention Detection

Figure 1: Comparison of our system with others on MUC-6 and MUC-7 NE tasks

<table>
<thead>
<tr>
<th>Composition</th>
<th>F</th>
<th>P</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f = f^1$</td>
<td>77.6</td>
<td>81.0</td>
<td>74.1</td>
</tr>
<tr>
<td>$f = f^1 f^2$</td>
<td>87.4</td>
<td>88.6</td>
<td>86.1</td>
</tr>
<tr>
<td>$f = f^1 f^2 f^3$</td>
<td>89.3</td>
<td>90.5</td>
<td>88.2</td>
</tr>
<tr>
<td>$f = f^1 f^2 f^4$</td>
<td>92.9</td>
<td>92.6</td>
<td>93.1</td>
</tr>
<tr>
<td>$f = f^1 f^2 f^3 f^4$</td>
<td>94.1</td>
<td>93.7</td>
<td>94.5</td>
</tr>
</tbody>
</table>

Effect of adding additional features
Shortcomings of HMMs

Shortcomings of HMM
1. HMM’s maximize likelihood of observation sequence (metric divergence problem)
2. Don’t consider non-independent observational variables or difficult to enumerate observational variables

Addressing HMM Shortcomings
1. Instead of modeling the joint probability of state and observation \( p(O_T, S_T) \), model the discriminative probability, \( p(S_T | O_T) \).
2. This allows for a plethora of features that can be used
   - words
   - line length
   - grammatical
   - contextual

\[ p(O_T, S_T) \text{ vs } p(S_T | O_T) \]

Generative

Discriminative
Maximum Entropy Markov Models

Max-Ent Markov Models

- conditional model that represents the probability of reaching a state given an observation and the previous state
- conditional probabilities are specified by exponential models based on arbitrary observation features

Learning

- Given \( O \) and \( S \), find \( M \) such that \( p(S|O,M) \) is maximized (maximum likelihood)
Maximum Entropy Markov Models

1. **ME-Stateless**: 24 Features, no context
2. **TokenHMM**: Traditional, fully-connected HMM (model switches states at line boundaries)
3. **FeatureHMM**: Similar to TokenHMM but lines are converted into features
4. **Maximum Entropy Markov Model**:

<table>
<thead>
<tr>
<th>Learner</th>
<th>COAP</th>
<th>SegPrec</th>
<th>SegRecall</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME-Stateless</td>
<td>0.520</td>
<td>0.038</td>
<td>0.362</td>
</tr>
<tr>
<td>TokenHMM</td>
<td>0.865</td>
<td>0.276</td>
<td>0.140</td>
</tr>
<tr>
<td>FeatureHMM</td>
<td>0.941</td>
<td>0.413</td>
<td>0.529</td>
</tr>
<tr>
<td>MEMM</td>
<td>0.965</td>
<td>0.867</td>
<td>0.681</td>
</tr>
</tbody>
</table>

**COAP**: COo-occurrence agreement probability  
**SegPrec**: Segmentation Precision  
**SegRecall**: Segmentation Probability
Conditional Random Fields for Entity Mentions

Insights

- Discriminative models often achieve better results than fully generative models (HMM)
- As such training Conditional Random Fields is natural method for effective noun-phrase chunking

Best of both words:

- Like classification models, they can accommodate many statistically correlated features of the inputs, and they are trained discriminatively
- Like generative models, they can trade off decisions at different sequence positions to obtain a globally optimal labeling
Conditional Random Fields for Entity Mentions

CRF's outperform other state-of-the-art methodologies including MEMM and SVM.

<table>
<thead>
<tr>
<th>Model</th>
<th>F score</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM combination (Kudo and Matsumoto, 2001)</td>
<td>94.39%</td>
</tr>
<tr>
<td>CRF</td>
<td>94.38%</td>
</tr>
<tr>
<td>Generalized winnow (Zhang et al., 2002)</td>
<td>93.89%</td>
</tr>
<tr>
<td>Voted perceptron</td>
<td>94.09%</td>
</tr>
<tr>
<td>MEMM</td>
<td>93.70%</td>
</tr>
</tbody>
</table>
Application: Anatomical Entity Mention Detection

Anatomical entities such as *kidney, muscle, blood* are prevalent in the life-science and biomedical literature.

- Detection of these entities is therefore quite invaluable in the automatic analysis of the structure of these *domain texts*.

- CRF for Entity Mention
- Meta-Map for Entity mention
- Combination Method
Semi-Markov CRF

Relaxing the Markov Assumption

- Semi-markov models extend traditional HMMs by relaxing the Markov assumption and allowing a state $S_i$ to persist for a non-unit length of time.
- These are also conditionally trained and therefore are discriminative and not generative.

Features Used

- Indicators for key words within 3-word window
- Capitalization/letter patterns (digits, etc) within 3-word window
- External dictionary for dictionary-derived features
Semi-Markov CRF

F1 values for different order CRFs

<table>
<thead>
<tr>
<th></th>
<th>CRF/1</th>
<th>CRF/4</th>
<th>semi-CRF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L = 1$</td>
<td>$L = 2$</td>
<td>$L = 3$</td>
</tr>
<tr>
<td>Address_State</td>
<td>20.8</td>
<td>20.1</td>
<td>19.2</td>
</tr>
<tr>
<td>Address_City</td>
<td>70.3</td>
<td>71.0</td>
<td>71.2</td>
</tr>
<tr>
<td>Email_persons</td>
<td>67.6</td>
<td>63.7</td>
<td>66.7</td>
</tr>
</tbody>
</table>
**Incremental Joint Entity and Relation Detection**

**Insight**
- Jointly extract both entities and relations to improve both subtasks

**Joint Extraction**
- Segment-based decoder based on a semi-Markov chain is adopted (instead of token-based taggers)
- Incrementally detects mention & relation boundaries (detects mentions on the segment level)
- Global features used as soft constraints
Incremental Joint Entity and Extraction

Comparison of pipeline vs joint extraction (global and local features)
Three Families of Methods

A. Supervised/Semi-supervised Entity Mention Detection

B. Unsupervised Entity Mention Detection

C. Weakly and Distantly Supervised Mention Detection
Unsupervised Entity Mention Detection

Assumptions

1. Part-of-speech tags are relatively inexpensive to obtain training data for
2. Part-of-speech tags generalize much better to new domains than parsing does
3. Training data is not available.

As such, we consider the use of POS tags as an input to these methods.

There are a variety of methods in use for POS tagging
NP-Chunking with Chunking Grammars

Observations

- Noun phrase chunks are smaller than full noun phrases (NP Chunks should not contain other NP Chunks)

Grammar: `<DT>?<JJ>*<NN>`

We saw the big yellow dog.
More Chunking Patterns

- After observing the data, one can define many relevant chunking patterns for entity mentions

  - `<DT>?<JJ>*<NN>`
  - `<PP>?<JJ>*<NN>`
  - `<JJ>*<NN>+`
  - `<JJ>*<NNP>+`
Improving Chunking

- Sometimes the Chunking Patterns may be less aggressive in identifying entity mentions
- One approach is to specify items (stopwords or POS tags) that can be used to split large noun chunks into smaller elements
- It may be easier to specify what *shouldn’t belong in a chunk*
Leveraging Corpus Level Information

Corpus-level entity mention detection has the benefit of leveraging corpus-level statistics to aid in determining mention boundaries.

Insights

1. Redundancy: Core entity mentions likely appear multiple times in the corpus
2. Longer candidate entity mentions should not be favored over shorter, more common, sub-mentions without evidence
A Noun Collocation Mining Approach

Good entity mentions are noun phrases that appear more frequently in a corpus than expected.

- Humans can define high-precision chunking grammars
- Corpus level statistics through *redundancy* can aid entity mention detection

Detecting high-quality entity mention candidates requires *both*:
  - *accurate POS-based pattern matching*
  - *Identification of significant patterns*
A Noun Collocation Mining Approach

- A framework for identifying entity mentions within domain-specific corpora

  raw corpus → sentence segmentation → tokenization

  entity mention identification → frequent segment mining → stopword removal

We identify these entity mentions using a Significant Mention Chunking Algorithm
Corpus Level Statistics

- $V(\text{segment})$ denotes the count of a segment
- Given two segments, we can obtain a **significance** of merging two such segments

$$\rho_X(S_1, S_2) = \frac{\nu(S_1 \oplus S_2) - N \frac{\nu(S_1)}{N} \frac{\nu(S_2)}{N}}{\sqrt{\nu(S_1 \oplus S_2)}} \cdot I_X(S_1 \oplus S_2)$$

dropped nearly inches of snow in Western Oklahoma and at Dallas Fort Worth International Airport sleet and ice caused

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>dallas</td>
<td>30</td>
</tr>
<tr>
<td>fort</td>
<td>60</td>
</tr>
<tr>
<td>international</td>
<td>80</td>
</tr>
<tr>
<td>airport</td>
<td>40</td>
</tr>
<tr>
<td>worth</td>
<td>80</td>
</tr>
<tr>
<td>dallas fort</td>
<td>26</td>
</tr>
<tr>
<td>International airport</td>
<td>35</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Differences from KeyPhrase Extraction

- Other methods may use significance score to rank methods that are significant highly.
  - This may allow for low quality entity phrases that appear significant to rank highly.

- This Noun Collocation mining differs from key phrase extraction in one major way.
  - Noun Collocation Mining goes to the exact location where a candidate phrase occurs and *segments the sentence* which simultaneously filters out bad entity candidates.
Significant Mention Chunking Algorithm

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

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

<table>
<thead>
<tr>
<th>Phrase</th>
<th>Raw freq.</th>
<th>True freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>[support vector machine]</td>
<td>90</td>
<td>80</td>
</tr>
<tr>
<td>[vector machine]</td>
<td>95</td>
<td>0</td>
</tr>
<tr>
<td>[support vector]</td>
<td>100</td>
<td>20</td>
</tr>
</tbody>
</table>

Markov Blanket Feature Selection for Support Vector Machines.
1. With all stopwords removed from consideration, we search for chunks that meet the following grammar

2. Among grammar matches, only merge “significant” noun phrases

Entity Grammar

\(<\text{JJ}>*<\text{NN}>*\)

Over the weekend the system dropped nearly inches of snow in western [Oklahoma] and at [Dallas Fort Worth International Airport] sleet and ice caused hundreds of [flight cancellations] ... It is forecast to reach by [Tuesday afternoon] [Washington] and [New York] by [Wednesday afternoon]
Application: Significant Keyphrase Extraction

1. First take input text corpus and apply POS-Constrained Collocation Mining

Over the weekend the system dropped nearly inches of snow in Western Oklahoma and at Dallas Fort Worth International Airport sleet and ice caused hundreds of flight cancellations ...

Over the weekend the system dropped nearly inches of snow in Western Oklahoma and at [Dallas Fort Worth International Airport] sleet and ice caused hundreds of [flight cancellations] ...

The POS constrain the collocation mining. This finds corpus-relevant key phrases.

These significant multi-word phrases can be used for a variety of applications.
2. One application is applying phrase-based 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

Over the weekend the system dropped nearly inches of snow in Western Oklahoma and at [Dallas Fort Worth International Airport] sleet and ice caused hundreds of [flight cancellations] ...

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

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

Running time of different algorithms

Phrase quality measured by z-score
POS-Constraining ToPMine

ToPMine divides the topical phrase extraction process into two steps

1. Segmenting the raw corpus into single and multi-word phrases
2. Performing phrase-constrained topic modeling

Since POS-Constrained noun collocation mining also segments the corpus, we can integrate the noun-collocation mining as a first step into ToPMine

This leads to POS-Constrained ToPMine:
Each phrase is a higher-quality phrase because of the part-of-speech constraints!
Observing ToPMine on Yelp Reviews, we can see some bad topical phrases can be filtered by enforcing our POS constraints.

### Topic 1

<table>
<thead>
<tr>
<th>ToPMine</th>
<th>POS-Constrained TopMine</th>
</tr>
</thead>
<tbody>
<tr>
<td>spring rolls</td>
<td>spring rolls</td>
</tr>
<tr>
<td><strong>food was good</strong></td>
<td>fried rice</td>
</tr>
<tr>
<td>fried rice</td>
<td>egg rolls</td>
</tr>
<tr>
<td>egg rolls</td>
<td>dim sum</td>
</tr>
<tr>
<td><strong>pretty good</strong></td>
<td>Thai food</td>
</tr>
<tr>
<td>dim sum</td>
<td>Chinese food</td>
</tr>
<tr>
<td>Thai food</td>
<td>pad thai</td>
</tr>
</tbody>
</table>

### Topic 2

<table>
<thead>
<tr>
<th>ToPMine</th>
<th>POS-Constrained TopMine</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>great selection</strong></td>
<td>grocery store</td>
</tr>
<tr>
<td>farmer’s market</td>
<td>farmer’s market</td>
</tr>
<tr>
<td><strong>great prices</strong></td>
<td>parking lot</td>
</tr>
<tr>
<td>wal mart</td>
<td>shopping center</td>
</tr>
<tr>
<td><strong>prices are reasonable</strong></td>
<td>county market</td>
</tr>
<tr>
<td>great place</td>
<td>fresh produce</td>
</tr>
<tr>
<td>love this place</td>
<td>wal mart supercenter</td>
</tr>
</tbody>
</table>
A. Supervised/Semi-supervised Entity Mention Detection

B. Unsupervised Entity Mention Detection

C. Weakly and Distantly Supervised Mention Detection
Weakly Supervised Methods

Assumptions

1. Unsupervised methods cannot possibly take into consideration the innumerable features, signals, and cues for entity mentions.
2. Full supervision can be too expensive (time-wise) to manage.

- Use methods that require small numbers of labeled instances (small number of seed entities).
- Rely on entity information from knowledge bases as seed entities.
Semi-Supervised Chunker with Structure learning

**Insight:** Use unlabeled to identify underlying structure of what makes a “good classifier”

1. Learns the concept of a “good classifier” by learning from thousands of automatically generated auxiliary classification on unlabeled data
2. Predictive structure shared by multiple classifiers can be discovered and used to improve performance on target problem

<table>
<thead>
<tr>
<th></th>
<th>English, all (204K) training examples</th>
<th>dev.</th>
<th>93.15</th>
<th>+2.25</th>
<th>+3.00</th>
<th>+2.62</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASO-semi co/self oracle</td>
<td>90.64</td>
<td>+0.04</td>
<td>+0.20</td>
<td>+0.11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASO-semi co/self oracle</td>
<td>test</td>
<td>89.31</td>
<td>+3.20</td>
<td>+4.51</td>
<td>+3.86</td>
<td></td>
</tr>
<tr>
<td>ASO-semi co/self oracle</td>
<td>85.40</td>
<td>-0.04</td>
<td>-0.05</td>
<td>-0.05</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Exploiting Dictionaries in Mention Detection

Challenges

- Most mention detections sequentially classify words in whether they participate in a candidate mention
- Similarity measures are applied to *full entity mention candidates*

**Proposed Method**

- Semi-markov extraction, sequentially classifies segments instead of tokens
- Allows for integration of entity mention detection methods and similarity methods with external data
Exploiting Dictionaries in Mention Detection

Observations

- Semi-Markov Model & HMM implementations with & without dictionary features on NER tasks
- Distance-based incorporation of dictionary values outperforms binary features

<table>
<thead>
<tr>
<th></th>
<th>Without dictionary</th>
<th></th>
<th>With dictionary</th>
<th></th>
<th>Distance features</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall</td>
<td>Prec.</td>
<td>F1</td>
<td>Recall</td>
<td>Prec.</td>
<td>F1</td>
</tr>
<tr>
<td>Address-state</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lookup</td>
<td>5.2</td>
<td>56.8</td>
<td>9.5</td>
<td>32.2</td>
<td>100.0</td>
<td>48.7</td>
</tr>
<tr>
<td>HMM-VP(3)</td>
<td>8.9</td>
<td>90.7</td>
<td><strong>16.2</strong></td>
<td>19.3</td>
<td>82.6</td>
<td><strong>31.3</strong></td>
</tr>
<tr>
<td>HMM-VP(4)</td>
<td>8.2</td>
<td>62.2</td>
<td>14.6</td>
<td>13.0</td>
<td>97.3</td>
<td>23.0</td>
</tr>
<tr>
<td>SMM-VP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address-city</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lookup</td>
<td>60.1</td>
<td>79.3</td>
<td>68.3</td>
<td>14.8</td>
<td>68.8</td>
<td>24.3</td>
</tr>
<tr>
<td>HMM-VP(3)</td>
<td>59.1</td>
<td>87.3</td>
<td>70.5</td>
<td>68.0</td>
<td>84.2</td>
<td>75.2</td>
</tr>
<tr>
<td>HMM-VP(4)</td>
<td>62.8</td>
<td>87.5</td>
<td><strong>73.1</strong></td>
<td>64.1</td>
<td>91.2</td>
<td>75.2</td>
</tr>
<tr>
<td>SMM-VP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email-person</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lookup</td>
<td>60.4</td>
<td>74.9</td>
<td>66.8</td>
<td>38.7</td>
<td>82.6</td>
<td>57.3</td>
</tr>
<tr>
<td>HMM-VP(3)</td>
<td>60.9</td>
<td>80.2</td>
<td>69.3</td>
<td>73.4</td>
<td>83.7</td>
<td>78.2</td>
</tr>
<tr>
<td>HMM-VP(4)</td>
<td>64.1</td>
<td>80.3</td>
<td><strong>71.3</strong></td>
<td>71.1</td>
<td>87.6</td>
<td>78.5</td>
</tr>
<tr>
<td>SMM-VP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job-company</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lookup</td>
<td>1.3</td>
<td>34.7</td>
<td>2.5</td>
<td>14.1</td>
<td>54.8</td>
<td>22.3</td>
</tr>
<tr>
<td>HMM-VP(3)</td>
<td>3.6</td>
<td>59.8</td>
<td>6.8</td>
<td>2.0</td>
<td>28.1</td>
<td>3.8</td>
</tr>
<tr>
<td>HMM-VP(4)</td>
<td>5.2</td>
<td>55.3</td>
<td><strong>9.6</strong></td>
<td>11.5</td>
<td>80.6</td>
<td>20.2</td>
</tr>
<tr>
<td>SMM-VP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job-title</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lookup</td>
<td>18.4</td>
<td>43.7</td>
<td>25.9</td>
<td>29.4</td>
<td>29.5</td>
<td>29.4</td>
</tr>
<tr>
<td>HMM-VP(3)</td>
<td>17.3</td>
<td>51.5</td>
<td>25.9</td>
<td>23.9</td>
<td>43.2</td>
<td>30.8</td>
</tr>
<tr>
<td>HMM-VP(4)</td>
<td>20.9</td>
<td>52.0</td>
<td><strong>29.8</strong></td>
<td>27.9</td>
<td>48.4</td>
<td>35.4</td>
</tr>
</tbody>
</table>

Table 3: Performance of NER methods on five IE tasks under three conditions: with no external dictionary; with an external dictionary and binary features; with an external dictionary and distance features.
Citation recommendation is an interesting but challenging research problem in data mining area.

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

Principal Component Analysis is a linear dimensionality reduction technique commonly used in machine learning applications.
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: this paper demonstrates

- Pointwise mutual information:

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

- Pointwise KL divergence:

  \[ PKL(v \| (u_l, u_r)) = p(v) \log \frac{p(v)}{p(u_l)p(u_r)} \]

- 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:

<table>
<thead>
<tr>
<th>sequence</th>
<th>frequency</th>
<th>phrase?</th>
<th>rectified</th>
</tr>
</thead>
<tbody>
<tr>
<td>support vector machine</td>
<td>100</td>
<td>yes</td>
<td>80</td>
</tr>
<tr>
<td>support vector</td>
<td>160</td>
<td>yes</td>
<td>50</td>
</tr>
<tr>
<td>vector machine</td>
<td>150</td>
<td>no</td>
<td>6</td>
</tr>
<tr>
<td>support vector</td>
<td>500</td>
<td>N/A</td>
<td>150</td>
</tr>
<tr>
<td>vector</td>
<td>1000</td>
<td>N/A</td>
<td>200</td>
</tr>
<tr>
<td>machine</td>
<td>1000</td>
<td>N/A</td>
<td>150</td>
</tr>
</tbody>
</table>
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 $\alpha$: 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**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#docs</th>
<th>#words</th>
<th>#labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBLP</td>
<td>2.77M</td>
<td>91.6M</td>
<td>300</td>
</tr>
<tr>
<td>Yelp</td>
<td>4.75M</td>
<td>145.1M</td>
<td>300</td>
</tr>
</tbody>
</table>

- **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+
Performance Study: Processing Efficiency

- SegPhrase+ is linear to the size of corpus!
Performance Study: Processing Efficiency

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

<table>
<thead>
<tr>
<th>Rank</th>
<th>Phrase</th>
<th>In English</th>
</tr>
</thead>
<tbody>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>62</td>
<td>首席_执行官</td>
<td>CEO</td>
</tr>
<tr>
<td>63</td>
<td>中间_偏右</td>
<td>Middle-right</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>84</td>
<td>百度_百科</td>
<td>Baidu Pedia</td>
</tr>
<tr>
<td>85</td>
<td>热带_气旋</td>
<td>Tropical cyclone</td>
</tr>
<tr>
<td>86</td>
<td>中国科学院_院士</td>
<td>Fellow of Chinese Academy of Sciences</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>1001</td>
<td>十大_中文_金曲</td>
<td>Top-10 Chinese Songs</td>
</tr>
<tr>
<td>1002</td>
<td>全球_资讯网</td>
<td>Global Info Website</td>
</tr>
<tr>
<td>1003</td>
<td>天一阁_藏_明代_科举_录_选刊</td>
<td>A Chinese book name</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>9934</td>
<td>国家_戏剧_院</td>
<td>National Theater</td>
</tr>
<tr>
<td>9935</td>
<td>谢谢_你</td>
<td>Thank you</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
## Experimental Results: Interesting Phrases Generated (From the Titles and Abstracts of SIGMOD)

<table>
<thead>
<tr>
<th>Query</th>
<th>SIGMOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>SegPhrase+</td>
</tr>
<tr>
<td>1</td>
<td>data base</td>
</tr>
<tr>
<td>2</td>
<td>database system</td>
</tr>
<tr>
<td>3</td>
<td>relational database</td>
</tr>
<tr>
<td>4</td>
<td>query optimization</td>
</tr>
<tr>
<td>5</td>
<td>query processing</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>51</td>
<td>sql server</td>
</tr>
<tr>
<td>52</td>
<td>relational data</td>
</tr>
<tr>
<td>53</td>
<td>data structure</td>
</tr>
<tr>
<td>54</td>
<td>join query</td>
</tr>
<tr>
<td>55</td>
<td>web service</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>201</td>
<td>high dimensional data</td>
</tr>
<tr>
<td>202</td>
<td>location based service</td>
</tr>
<tr>
<td>203</td>
<td>xml schema</td>
</tr>
<tr>
<td>204</td>
<td>two phase locking</td>
</tr>
<tr>
<td>205</td>
<td>deep web</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Only in SegPhrase+:
- high dimensional data
- two phase locking
- deep web

Only in Chunking:
- important issue
- efficient implementation
- location based service
- xml schema

**Note:** SegPhrase+ uses TF-IDF & C-Value for chunking.
Experimental Results: Interesting Phrases Generated (From the Titles and Abstracts of SIGKDD)

<table>
<thead>
<tr>
<th>Query</th>
<th>SIGKDD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>SegPhrase+</td>
</tr>
<tr>
<td>1</td>
<td>data mining</td>
</tr>
<tr>
<td>2</td>
<td>data set</td>
</tr>
<tr>
<td>3</td>
<td>association rule</td>
</tr>
<tr>
<td>4</td>
<td>knowledge discovery</td>
</tr>
<tr>
<td>5</td>
<td>time series</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>51</td>
<td>association rule mining</td>
</tr>
<tr>
<td>52</td>
<td>rule set</td>
</tr>
<tr>
<td>53</td>
<td>concept drift</td>
</tr>
<tr>
<td>54</td>
<td>knowledge acquisition</td>
</tr>
<tr>
<td>55</td>
<td>gene expression data</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>201</td>
<td>web content</td>
</tr>
<tr>
<td>202</td>
<td>frequent subgraph</td>
</tr>
<tr>
<td>203</td>
<td>intrusion detection</td>
</tr>
<tr>
<td>204</td>
<td>categorical attribute</td>
</tr>
<tr>
<td>205</td>
<td>user preference</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Only in SegPhrase+</th>
<th>...</th>
<th>Only in Chunking</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimal solution</td>
<td>...</td>
<td>effective way</td>
</tr>
<tr>
<td>semantic relationship</td>
<td></td>
<td>space complexity</td>
</tr>
<tr>
<td>small set</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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”

<table>
<thead>
<tr>
<th>Query</th>
<th>data mining</th>
<th>Chunking</th>
<th>olap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>SegPhrase+</td>
<td>driven methodologies</td>
<td>data warehouse</td>
</tr>
<tr>
<td>1</td>
<td>knowledge discovery</td>
<td>text mining</td>
<td>online analytical processing</td>
</tr>
<tr>
<td>2</td>
<td>text mining</td>
<td>financial investment</td>
<td>data cube</td>
</tr>
<tr>
<td>3</td>
<td>web mining</td>
<td>knowledge discovery</td>
<td>olap queries</td>
</tr>
<tr>
<td>4</td>
<td>machine learning</td>
<td>building knowledge</td>
<td>multidimensional databases</td>
</tr>
<tr>
<td>5</td>
<td>data mining techniques</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query</th>
<th>blu-ray</th>
<th>Chunking</th>
<th>noodle</th>
<th>Chunking</th>
<th>valet parking</th>
<th>Chunking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>SegPhrase+</td>
<td>new microwave</td>
<td>Ramen</td>
<td>noodle soup</td>
<td>valet</td>
<td>huge lot</td>
</tr>
<tr>
<td>1</td>
<td>Dvd</td>
<td>new microwave</td>
<td>Ramen</td>
<td>noodle soup</td>
<td>valet</td>
<td>huge lot</td>
</tr>
<tr>
<td>2</td>
<td>vhs</td>
<td>Lifetime warranty</td>
<td>Noodle soup</td>
<td>Asian noodle</td>
<td>self-parking</td>
<td>private lot</td>
</tr>
<tr>
<td>3</td>
<td>cd</td>
<td>Recliner</td>
<td>Rice noodle</td>
<td>Beef noodle</td>
<td>valet service</td>
<td>self-parking</td>
</tr>
<tr>
<td>4</td>
<td>new release</td>
<td>Battery</td>
<td>Egg noodle</td>
<td>Stir fry</td>
<td>free valet parking</td>
<td>valet</td>
</tr>
<tr>
<td>5</td>
<td>sony</td>
<td>New battery</td>
<td>Pasta</td>
<td>Fish ball</td>
<td>covered parking</td>
<td>front lot</td>
</tr>
</tbody>
</table>
8. Li, Qi, and Heng Ji. "Incremental joint extraction of entity mentions and relations." ACL. 2014.


Outline

1. Introduction to entity recognition and typing
2. Entity recognition – overview and phrase mining approach
3. Entity typing – overview and network mining approach
4. Trends and research problems
Entity Typing on General-Domain, Formal Corpora

Assumptions

1. A good amount of label data is available
2. Primitive NLP methods (e.g., POS tagging, NP chunking, dependency parsing) can provide decent & robust features
3. Most entities mentioned in text are available in knowledge bases
Entity Typing on General-Domain Corpora

A. Supervised Entity Typing
B. Semi-Supervised Entity Typing
C. Entity linking for typing
D. Weakly-Supervised Entity Typing
E. Distantly-Supervised Entity Typing
Entity Typing on General-Domain Corpora

A. Supervised Entity Typing
- Decision tree
- Support Vector Machine
- Sequence labeling models

B. Semi-Supervised Entity Typing

C. Entity linking for Entity Typing
Supervised Learning for Entity Typing

- **Diagram A**: I-O-B encoding for classification

  Steve  Jobs  was  a  co-founder  of  Apple  Inc.

  - Steve  B-PER
  - Jobs  I-PER
  - was  O
  - a  O
  - co-founder  O
  - of  O
  - Apple  B-ORG
  - Inc.  I-ORG

- **Problem setting**: classify each token into corresponding I-O-B label

- **Diagram B**: detected entity mentions for classification

  Steve  Jobs  was  a  co-founder  of  Apple  Inc.

  - Steve  PER
  - Jobs  PER
  - was  O
  - a  O
  - co-founder  O
  - of  O
  - Apple  B-ORG
  - Inc.  I-ORG

- **Problem setting**: classify each mention into corresponding type
Workflow of Supervised Entity Typing

- **Training**
  1. Collect a set of training documents
  2. Label each token (entity mention) for its entity class or other (O)
  3. Design feature extractors appropriate to the text and classes
  4. Train a classifier to predict the labels from the data

- **Testing**
  1. Receive testing document (a single document or a batch)
  2. Run trained classifier to label each token
  3. Appropriately output the recognized entities
## Features for Classification (word-level)

<table>
<thead>
<tr>
<th>Features</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case</td>
<td>- Starts with a capital letter</td>
</tr>
<tr>
<td></td>
<td>- Word is all uppercased</td>
</tr>
<tr>
<td></td>
<td>- The word is mixed case (e.g., ProSys, eBay)</td>
</tr>
<tr>
<td>Punctuation</td>
<td>- Ends with period, has internal period (e.g., St., I.B.M.)</td>
</tr>
<tr>
<td></td>
<td>- Internal apostrophe, hyphen or ampersand (e.g., O’Connor)</td>
</tr>
<tr>
<td>Digit</td>
<td>- Digit pattern (see section 3.1.1)</td>
</tr>
<tr>
<td></td>
<td>- Cardinal and Ordinal</td>
</tr>
<tr>
<td></td>
<td>- Roman number</td>
</tr>
<tr>
<td></td>
<td>- Word with digits (e.g., W3C, 3M)</td>
</tr>
<tr>
<td>Character</td>
<td>- Possessive mark, first person pronoun</td>
</tr>
<tr>
<td></td>
<td>- Greek letters</td>
</tr>
<tr>
<td>Morphology</td>
<td>- Prefix, suffix, singular version, stem</td>
</tr>
<tr>
<td></td>
<td>- Common ending (see section 3.1.2)</td>
</tr>
<tr>
<td>Part-of-speech</td>
<td>- proper name, verb, noun, foreign word</td>
</tr>
<tr>
<td>Function</td>
<td>- Alpha, non-alpha, n-gram (see section 3.1.3)</td>
</tr>
<tr>
<td></td>
<td>- lowercase, uppercase version</td>
</tr>
<tr>
<td></td>
<td>- pattern, summarized pattern (see section 3.1.4)</td>
</tr>
<tr>
<td></td>
<td>- token length, phrase length</td>
</tr>
</tbody>
</table>

Feature is more important than classification models!
Features for Classification (doc/corpus-level)

<table>
<thead>
<tr>
<th>Features</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple occurrences</td>
<td>- Other entities in the context</td>
</tr>
<tr>
<td></td>
<td>- Uppercased and lowercased occurrences (see 3.3.1)</td>
</tr>
<tr>
<td></td>
<td>- Anaphora, coreference (see 3.3.2)</td>
</tr>
<tr>
<td>Local syntax</td>
<td>- Enumeration, apposition</td>
</tr>
<tr>
<td></td>
<td>- Position in sentence, in paragraph, and in document</td>
</tr>
<tr>
<td>Meta information</td>
<td>- Uri, Email header, XML section, (see section 3.3.3)</td>
</tr>
<tr>
<td></td>
<td>- Bulleted/numbered lists, tables, figures</td>
</tr>
<tr>
<td>Corpus frequency</td>
<td>- Word and phrase frequency</td>
</tr>
<tr>
<td></td>
<td>- Co-occurrences</td>
</tr>
<tr>
<td></td>
<td>- Multiword unit permanency (see 3.3.4)</td>
</tr>
</tbody>
</table>

- Distributional features
  - Each word will appear in contexts - induce a distribution over contexts
  - Cluster words based on how similar their distributions are
  - Use cluster IDs as features → great way to combat sparsity

[Nadeau & Sekine 07]
Standard Classification: Decision Trees

- Binary classification following diagram B

Workflow:
- Select feature to test at each node in the tree.
- Top-down, greedy search through the space of possible decision trees. It picks the best attribute and never looks back to reconsider earlier choices.

Pros
- + generate understandable rules
- + provide a clear indication of which features are most important for classification

Cons
- - error prone in multi-class classification and small number of training examples
- - expensive to train due to pruning
Standard Classification: Support Vector Machine

- Binary classification following diagram B
- Insights:
  - Negative examples are sampled from co-occurring entities which are not of the target types
  - Quadratic kernel gives the best performance

\[ f(x) = \text{sign}(g(x)) \]
\[ g(x) = \sum_{i=1}^{\ell} w_i K(x, z_i) + b. \]

[Isozaki et al. 02]
Sequence Labeling Models

- Insights
  - vs. standard classification:
    - label depends not only on its corresponding observation but also possibly on other observations and other labels in the sequence

Steve B-PER
Jobs ?
was O
a O
co-founder O
of O
Apple B-ORG
Inc. I-ORG

Sequence labeling model I-PER
# Model Trade-offs

<table>
<thead>
<tr>
<th></th>
<th>Speed</th>
<th>Discrim vs. Generative</th>
<th>Normalization</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>very fast</td>
<td>generative</td>
<td>local</td>
</tr>
<tr>
<td>MEMM</td>
<td>mid-range</td>
<td>discriminative</td>
<td>local</td>
</tr>
<tr>
<td>CRF</td>
<td>kinda slow</td>
<td>discriminative</td>
<td>global</td>
</tr>
</tbody>
</table>

[Chris Manning 07]
Greedy inference:
- The classifier can depend on previous labeling decisions as well as observed data
- Fast; greedy (make commit errors)

Viterbi Inference
- Dynamic programming or memorization
- Exact; harder to implement long-distance state-state interactions

Beam inference:
- Keep top-k complete sequences at each position; extend them in each local way
- Fast; inexact (fall off global optimal sequence)
Entity Typing on General-Domain Corpora

A. Supervised Entity Typing

B. Semi-Supervised Entity Typing
- Feature-level semi-supervised learning
- Semi-supervised sequence models

C. Entity linking for typing
Semi-Supervised Entity Typing

- **Goal:** leveraging large amount of unannotated corpus in addition to annotated corpus to augment model learning
  - More accurate results using similar amount of labeled data
  - Comparable performance with less amount of labeled data
- **Assumption:**
  - Data (feature) statistics from unannotated corpus can enhance model learning
- **Insights**
  - Features derived from unannotated corpus can be feed into supervised sequence models
  - Standard sequence models can be extended to model unlabeled data jointly
Feature-level semi-supervised learning

- **Insights**
  - Unsupervised word feature derived from a large corpus (both annotated and unannotated) can improve performance of existing supervised models

- **Feature representations**
  - Distributional word representation
    - Words from context windows
  - Clustering-based word representation
    - Brown clusters
  - Distributed word representations (word embedding)

    | System                        | Dev   | Test  |
    |-------------------------------|-------|-------|
    | Baseline                      | 94.16 | 93.79 |
    | HLBL, 50-dim                  | 94.63 | 94.00 |
    | C&W, 50-dim                   | 94.66 | 94.10 |
    | Brown, 3200 clusters          | 94.67 | 94.11 |
    | Brown+HLBL, 37M               | 94.62 | 94.13 |
    | C&W+HLBL, 37M                 | 94.68 | 94.25 |
    | Brown+C&W+HLBL, 37M           | 94.72 | 94.15 |
    | Brown+C&W, 37M                | 94.76 | 94.35 |
    | Ando and Zhang (2005), 15M    | -     | 94.39 |
    | Suzuki and Isozaki (2008), 15M| -     | 94.67 |
    | Suzuki and Isozaki (2008), 1B | -     | 95.15 |

Turian et al. ACL 2010.
Semi-supervised sequence models

- **Goal**: incorporate unlabeled data into discriminative sequence model training in an effective way
  - Unlabeled data can be naturally incorporated into generative models following an expectation-maximization

- **Insights**
  1. Consider dependencies between labels of nearby instances during model training \( \rightarrow \) encourage nearby data points to share label distribution
  2. Encourage the optimization procedure to find parameters that predict a similar label distribution on the unlabeled examples, given human-provided label prior distribution
Semi-supervised sequence models

- Models
  - Insight 1: semi-supervised CRF with entropy regularization on the unlabeled data
    \[
    RL(\theta) = \sum_{i=1}^{N} \log p_{\theta}(y^{(i)}|x^{(i)}) - U(\theta) \tag{2}
    \]
    \[
    \gamma \sum_{i=N+1}^{M} \sum_{y} p_{\theta}(y|x^{(i)}) \log p_{\theta}(y|x^{(i)})
    \]
  - Insight 2: use generalized expectation criteria to optimize CRF model

\[
p(y|x; \theta) = \frac{1}{Z(x)} \exp \left( \sum_{k} \theta_{k} \Psi_{k}(x, y) \right)
\]

\[
O(\theta; \mathcal{D}) = \sum_{d} \log p(y_{d}|x_{d}; \theta) - \frac{\sum_{k} \theta_{k}^{2}}{2\sigma^{2}}
\]
Entity Typing on General-Domain Corpora

A. Supervised Entity Typing
B. Semi-Supervised Entity Typing
C. Entity linking for typing
# Large Scale Taxonomies

<table>
<thead>
<tr>
<th>Name</th>
<th>Source</th>
<th># types</th>
<th># entities</th>
<th>Hierarchy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dbpedia (v3.9)</td>
<td>Wikipedia infoboxes</td>
<td>529</td>
<td>3M</td>
<td>Tree</td>
</tr>
<tr>
<td>YAGO2s</td>
<td>Wiki, WordNet, GeoNames</td>
<td>350K</td>
<td>10M</td>
<td>Tree</td>
</tr>
<tr>
<td>Freebase</td>
<td>Miscellaneous</td>
<td>23K</td>
<td>23M</td>
<td>Flat</td>
</tr>
<tr>
<td>Probase (MS.KB)</td>
<td>Web text</td>
<td>2M</td>
<td>5M</td>
<td>DAG</td>
</tr>
</tbody>
</table>

---

**Arnold Schwarzenegger**

Discuss "Arnold Schwarzenegger" | Show Empty Fields

- **Types:** Person (Person), US Politician (Governor), Film actor (Film), Film producer (Film), Pro Athlete (Sports), Sports Award Winner (Sports)
- **Also known as:** Arnold Alois Schwarzenegger, The Governor
- **Gender:** Male
- **Date of Birth:** Jul 30, 1947
- **Place of Birth:** Thal, Austria
- **Country Of Nationality:** United States
- **Profession:** Politician, Bodybuilder, Entrepreneur, Actor
Type Entities in Text

- Assumptions
  - Entities exist in the knowledge bases
  - Linked entities do not have type ambiguity (in the target type set)
    - “Washington D.C.” → Type: GOVERNMENT or LOCATION?

- Insights
  - [Context Similarity]: Contexts of the entity mention provide cues for linking it to the knowledge bases --- [Bunescu & Pascal 06] etc.
  - [Topic Coherence]: Entity mentions in a document/paragraph may share the same topics --- [Cucerzan 07] etc.
  - [Entity Popularity]: popular entity candidate is preferred to be linked to
  - Linking of multiple entity mentions in could be modeled jointly --- [Hoffart et al. 11] etc.
Limitation of Entity Linking

- Low recall of knowledge bases
- Sparse concept descriptors

Can we disambiguate entities without relying on knowledge bases?
- Yes! Exploit the redundancy in the corpus
  - Not relying on knowledge bases: targeted disambiguation of ad-hoc, homogeneous entities [Wang et al. 12]
  - Partially relying on knowledge bases: mining additional evidence in the corpus for disambiguation [Li et al. 13]

82 of 900 shoe brands exist in Wiki

Michael Jordan won the best paper award
Entity Typing on Domain-Specific, Informal Corpora

Assumptions

1. Very limited amount of (or no) labeled entity mentions are available for the corpus

2. Primitive NLP methods (e.g., NP chunking, dependency parsing) do not work well on the corpus

3. Only a small portion of entities in the corpus exist in knowledge bases
Entity Typing on Domain-Specific, Informal Corpora

A. Weakly-Supervised Entity Typing
- Pattern-based bootstrapping methods
- Contextual classifier induction method
- Graph-based semi-supervised learning

B. Distantly-Supervised Entity Typing
Weakly-Supervised Entity Typing

- **Problem setting**
  - A large unannotated corpus is available
  - A small set of labeled entity names (seeds) from the corpus are available

- **Assumptions on labeled data (seeds)**
  - Labeled entity names have *sufficient occurrences* in the corpus
  - Labeled entity names do not have *type ambiguity*
  - Labeled entity names *evenly cover all entity types*
Pattern-Based Bootstrapping Methods

To capture the redundancy in large corpus

Seed
Goldman-Sachs
Microsoft

Pattern-based bootstrapping module

Unlabeled Corpus

Context patterns
analyst at <X>
companies such as <X>
joint venture between <X>

Goldman-Sachs
Microsoft
Google
Morgan Stanley
Facebook

analyst at <X>
companies such as <X>
joint venture between <X>
Pattern-Based Bootstrapping Methods

Seed entities and unlabeled corpus

Label data using the current set of entities

Score candidate entities and select top-N

Create candidate patterns

Select Top-K patterns and apply them

Score candidate patterns

T iterations

[Thelen & Riloff 02]
Pattern-Based Bootstrapping Methods

- Assumption:
  - **Mutual exclusion**: positive examples (i.e., entity names) for one type are negative examples for other types

- Key questions:
  - How to induce effective patterns given entities $\rightarrow$ pattern induction
  - How to evaluate the extracted patterns? $\rightarrow$ pattern scoring
  - How to evaluate the extracted entities? $\rightarrow$ entity promotion
Pattern-Based Bootstrapping Methods

- **Pattern induction**
  - **Context word pattern** [Gupta & Manning 14]
    - Window of words (or lemmas) before/after a labeled entity mention (with POS restriction)
    - E.g., pay DT visit to X, X be located in, ...
  - **Literal pattern** [Lin et al. 03]
    - Context word window around a tag
    - E.g., “<PER>Steve Jobs </PER> is the co-founder of Apple Inc.”
      - right tag literal pattern: [* </PER> be DT co-founder of]
Pattern-Based Bootstrapping Methods

- Improve pattern induction
  - Lexico-syntactic pattern [Thelen & Riloff. 02]
  - Abstract context pattern using POS tags and dependency structures
  - E.g., `<subj>` passive-verb: `<victim>` was murdered
  - Trigger word-based automata [Talukdar et al. 06]
    - Extract dominating (trigger) word from the context and induce a automata for each trigger word

expression of -ENT- in
expression of -ENT- mrna
expression of the -ENT- gene

expression
Pattern-Based Bootstrapping Methods

- Pattern scoring
  - Seed for type “ANIMAL”: \{dog\}
  - Text:
    I own a dog named Tommy. I run with my pet dog. I also nap with my pet cat. I own a car.
  - Pattern 1: “my pet X”
    -- positive:\{dog\}, unlabeled:\{cat\}
  - Pattern 2: “own a X”
    -- positive:\{dog\}, unlabeled:\{car\}
  → Score: “my pet X” == “own a X”

- Improve pattern scoring
  - Positive entities, negative entities & unlabeled entities

- Past work
  - Ignore unlabeled entities [Carlson et al. 20, Blum & Mitchell 99]
  - Assumed negative [Downey et al., 04]
  - Both [Yangarber et al. 02, Lin et al. 03]

- Can we exploit unsupervised sources [Gupta & Manning 14]:
  - E.g, Sim(cat, dog) > Sim (car, dog)
  → “my pet X” > “own a X”
Pattern-Based Bootstrapping Methods

- **Entity promotion**
  - **Pattern quality**: prefer entities that match more reliable patterns
  - “Entities that are extracted more high weighted patterns get higher score”
  - **Entity domain popularity**: prefer more prevalent entities in the corpus
  - “Entity that appear more frequently in the corpus compared to the general domain”

- **Limitations**
  - Each entity name is assigned with only one type
  - Cannot handle ambiguous names---”Washington D.C.”
  - Error aggregation:
    - False entities/patterns cannot be corrected in the further iterations
Contextual classifier induction method

- **Insight**: learning contextual classifiers for entity promotion
- **Framework** [Huang et al. 10]
  - Phase I: Inducing contextual classifiers:
    - Binary classifier on context word windows of seeds of *each type*

*Fluffy was diagnosed with FELV after a blood test showed that he tested positive.*

```
was_3 diagnosed_2 with_1 after_1 test_2 showed_3 \Rightarrow DISEASE
```

```
with_3 FELV_2 after_1 blood_M showed_1 that_2 he_3 \Rightarrow TEST
```
Phase II: Cross-category bootstrapping:

- Re-train a suite of contextual classifiers
  - Positive examples of one type serve as negative examples for the other types
Contextual classifier induction method

- Assumptions
  - Seeds are semantically unambiguous
  - Noun phrase chunker can provide quality entity mention candidates

- Limitations
  - Seeded contexts may not be sufficient to trained an effective classifier
  - Error aggregation similar to that faced by pattern-based bootstrapping methods
Graph-Based Semi-Supervised Learning

- **Insights**
  - Many text corpus can be naturally and uniformly represented by a graph
  - Entity typing can then be modeled as graph-based semi-supervised learning problem

- **Assumptions**
  - Quality entity candidates are already extracted
  - **[Smoothness Assumption]**
    - “If two instances are similar according to the graph, then their labels should be similar.”
Graph-Based Semi-Supervised Learning

- **Graph construction**
  - Nodes
    - Entity (name) candidates
    - Context patterns
    - Group membership (e.g., web tables, web lists, ...)
    - ...
  - Link formation
    - Dense graph
    - K-nearest neighbor (KNN) graph
    - e-Neighborhood graph
Graph-Based Semi-Supervised Learning

- Graph construction
  - Link weighting
    - Binary (0/1)
    - Co-occurrence measures: e.g., PMI
    - Feature-based similarity: \( \text{sim}(x_i, x_j) \)
    - ...
  - Other construction approaches
    - Supervised metric learning
      - ITML [Kulis et al., ICML 2007]
    - Semi-supervised metric learning
      - IDML [Dhillon et al., UPenn TR 2010]
    - Local reconstruction [Daitch et al., ICML 2009]
Graph-Based Semi-Supervised Learning

- Learning Algorithms
  - Label propagation
  - Random walk
  - Graph Laplacian
  - LP-ZGL [Zhu et al. 03]

\[ f^T L f = \sum_{i,j} W_{ij} (f_i - f_j)^2 \]

Starting node

What next?

Measure of Non-Smoothness

Smooth

\[ \arg\min_{\hat{Y}} \sum_{l=1}^{m} W_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2 = \sum_{l=1}^{m} \hat{Y}_l^T L \hat{Y}_l \]

such that

\[ Y_{ul} = \hat{Y}_{ul}, \forall S_{uu} = 1 \]

Graph Laplacian

Match Seeds (hard)

Transductive learning
- Non-parametric
- Iterative update algorithms
Graph-Based Semi-Supervised Learning

- Learning Algorithms
  - Factor graph model [Kschischang et al. 01]
  - Bipartite graph
  - Variable nodes: e.g., type distribution on an entity candidate
  - Factor nodes: fitness function over variable assignment
Graph-Based Semi-Supervised Learning

- Learning Algorithms
  - Manifold regularization [Belkin et al., JMLR 2006]
  - Trains an inductive classifier which can generalize to unseen instances

\[
f^* = \arg \min_f \frac{1}{l} \sum_{i=1}^{l} V(y_i, f(x_i)) + \beta f^T L f + \gamma \|f\|_2^2
\]
Graph-Based Semi-Supervised Learning

- Graph construction can be key!
  - Similar to feature engineering for classifiers

- Advantages
  - Flexible to model various sources and signals uniformly
  - Easily parallelizable, scalable to large data

- Limitations
  - Cannot decide the exact type for each entity mention (name ambiguity)
  - Sensitive to seeds
A. Weakly-Supervised Entity Typing

- Multi-label Multi-class classification methods
- Label propagation methods
- Relation pattern repository-based method
- ClusType: A phrase and network mining approach

B. Distantly-Supervised Entity Typing

Entity Typing on Domain-Specific, Informal Corpora
Why Distantly-Supervised Entity Typing?

- Weakly-supervised methods still require human annotations

  - Assumptions on labels:
    - Labeled entity names have *sufficient occurrences* in the corpus
    - Labeled entity names are *semantically unambiguous*
    - Labeled entity names *evenly cover all entity types*

- Can we get rid of human supervision, and make it *fully automatic*?
  - Rich entity information in knowledge bases → “distant” supervision for entity typing
Typical Workflow of Distant Supervision

- 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

<table>
<thead>
<tr>
<th>ID</th>
<th>Document Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>... has concerns whether Kabul is an ally of Washington.</td>
</tr>
<tr>
<td>2</td>
<td>... Australia becomes a close ally of the United States. ...</td>
</tr>
<tr>
<td>3</td>
<td>He has offices in Washington, Boston and San Francisco.</td>
</tr>
<tr>
<td>4</td>
<td>... The Cardinal will share the title with California if the Golden Bears beat Washington later Saturday. ...</td>
</tr>
<tr>
<td>5</td>
<td>... Auburn won the game 34-28 over the defending national champions. ...</td>
</tr>
</tbody>
</table>
Multi-Class Multi-Label Classification

Assumptions:
- Entity mentions are already recognized from text
- Features for classifiers can be robustly computed from the corpus

Insights:
- Allow one entity mention to have multiple fine-grained types
- How to automatically label the data
- Use Wikipedia Anchor links as training data
- How to generate clean type set
- Filter by frequency & heuristics
Multi-Class Multi-Label Classification

- Pipeline of FIGER [Ling et al. 12]: segmentation + classification
  - Training:
    - Labeled Text from Wikipedia
      - He attends [Harvard University].
    - Organization, University
  - Test:
    - CRF for segmentation
    - Classifier for entity tagging
    - It was won by the Ottawa Senators, coached by Dave Gill.
    - Segmented Text:
      - It was won by the [Ottawa Senators], coached by [Dave Gill].
    - Person, Coach
    - Organization, Sports_Team

- Features
  - Tokens, bi-grams; Part-of-Speech tags; Brown word clusters; Head; Word shapes; Contextual tokens of the mention; Typed dependency; Light verb + noun patterns

- Limitations
  - Hard to find distant supervision like Wikipedia corpus for specific domains
Label propagation methods

- Assumptions
  - Entity mentions are pre-extracted for the corpus
  - There is no name ambiguity
  - Each entity surface name is assigned with one type

- Insights
  - Linked entities candidates serve as seeds
  - Apply entity linking on raw text
  - Contextual information (e.g., relation phrases) server as bridges to propagate type information between entity candidates
  - Construct graphs for type propagation
Label propagation methods

- Pipeline of NNPLB [Lin et al. 12]: detection + propagation

- Entity detection:
  - Noun phrases as candidates
  - Binary classifier: timestamps from Google Book n-grams as features

- Type propagation:
  - Leverage assertions (entity-relation-entity triple) extracted by OpenIE [Fader et al. 11]
  - Predict type of unlinkable candidate, by types of the linked entities which it share relations with

<table>
<thead>
<tr>
<th>system</th>
<th>correctly classified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority class baseline</td>
<td>50.4%</td>
</tr>
<tr>
<td>Named Entity Recognition</td>
<td>63.3%</td>
</tr>
<tr>
<td>Slope feature only</td>
<td>61.1%</td>
</tr>
<tr>
<td>PUF feature combination</td>
<td>69.1%</td>
</tr>
<tr>
<td>ALL features</td>
<td>78.4%</td>
</tr>
</tbody>
</table>

- Predicted textual relations and Freebase types:
  - "relied a great deal on"
  - "has released a minor update to"
  - "needs to let go of"
  - "has already announced"
  - "has a sharp focus on"
  - "also marketed"
  - "has released updates for"
  - "offers certifications for"

- Predicted Freebase types:
  - business_operation
  - employer
  - organization
  - software_developer
  - venture_funded_company
  - book_subject
  - venture_investor
  - website_owner

- Similar linked entities:
  - Microsoft (04sv4)
  - Apple Inc. (0k82)
  - Google (045c7b)
  - IBM (03sc8)
  - Adobe Systems (0vlf)
  - Cisco Systems (0dmtp)
  - Red Hat (02h5b_x)
  - AOL (0plw)
Relation pattern repository-based method

- Assumptions
  - Existence of a repository of relation patterns, organized in a type-signature taxonomy [PATTY, Nakashole et al. 12]
  - Entity candidates → noun phrases that are consistently capitalized

- Insights
  - Type signatures of the co-occurring (PATTY) relation patterns provide cues for the type of the entity candidates
  - Type disjointness constraints---indicates that a pair of types cannot apply to the same entity at the same time
Relation pattern repository-based method

- How to decide the types for an entity candidate
  - Aggregated likelihood
    \[
    \text{typeConf}(x, \text{domain}) = \sum_{\text{phrase}_i} \sum_{p_j} \left( \text{sim}(\text{phrase}_i, p_j) \times \Upsilon \right)
    \]
    Where \( \Upsilon = P[t_1, t_2 | p] \) or \( P[t_1 | t_2, p] \) or \( P[t_2 | t_1, p] \)

- Integer liner programming \( \rightarrow \) incorporate type disjointness constraints
  \[
  \max P_i T_i \times w_i \\
  \text{type disjointness constraint: } \forall (t_i, t_j)_{\text{disjoint}} T_i + T_j \leq 1
  \]
  Where \( W_i \) is the aggregated likelihood
Challenge I: Domain Restriction

- Most existing work assume entity mentions are already extracted by existing entity detection tools
- 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
Challenge 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

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.

Washington State or Washington Sport team -ment
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

<table>
<thead>
<tr>
<th>ID</th>
<th>Sentence</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The magnitude 9.0 quake <strong>caused widespread devastation in [Kesennuma city]</strong></td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>... tsunami that <strong>ravaged [northeastern Japan]</strong> last Friday</td>
<td>31</td>
</tr>
<tr>
<td>3</td>
<td>The resulting tsunami <strong>devastate [Japan]'s northeast</strong></td>
<td>244</td>
</tr>
</tbody>
</table>

Previous methods have difficulties in handling entity mention with sparse (infrequent) context
ClusType: Effective Entity Recognition and Typing by Relation Phrase-Based Clustering

XIANG REN, AHMED EL-KISHKY, CHI WANG, FANGBO TAO, CLARE R. VOSS, HENG JI, JIAWEI HAN, KDD 2015
Problem Definition

- Given:
  - A domain-specific corpus $D$
  - A knowledge base (e.g., Freebase)
  - A set of target types ($T$) from a KB

- Detect candidate entity mentions from corpus $D$

- Categorize each candidate mention by target types or Not-Of-Interest (NOI), with distant supervision
Our Solution

Domain-agnostic phrase mining algorithm

- Extracts candidate entity mentions with minimal linguistic/domain assumption ➞ address domain restriction

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 synonymous relation phrase co-occurring with entity mentions

- Helps form connecting bridges among entities that do not share identical context, but share synonymous relation phrases ➞ address context sparsity
A Relation Phrase-Based Entity Recognition Framework

- 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
A Relation Phrase-Based Entity Recognition Framework

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
**Framework Overview**

1. Perform phrase mining on a POS-tagged corpus to extract candidate entity mentions and relation phrases
2. Construct a heterogeneous graph to encode our insights on modeling the type for each entity mention
3. Collect seed entity mentions as labels by linking extracted mentions to the KB
4. Estimate type indicator for unlinkable candidate mentions with the proposed type propagation integrated with relation phrase clustering on the constructed graph
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

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>disperse; hit; struck; knock;</td>
</tr>
<tr>
<td>P</td>
<td>in; at; of; from; to;</td>
</tr>
<tr>
<td>V P</td>
<td>locate in; come from; talk to;</td>
</tr>
<tr>
<td>VW*(P)</td>
<td>caused major damage on; come lately</td>
</tr>
</tbody>
</table>

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

- Example output of candidate generation on NYT news articles


EP: entity mention candidate; RP: relation phrase

- Entity detection performance comparison with an NP chunker

<table>
<thead>
<tr>
<th>Method</th>
<th>NYT</th>
<th>Yelp</th>
<th>Tweet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prec</td>
<td>Recall</td>
<td>Prec</td>
</tr>
<tr>
<td>Our method</td>
<td>0.469</td>
<td>0.956</td>
<td>0.306</td>
</tr>
<tr>
<td>NP chunker</td>
<td>0.220</td>
<td>0.609</td>
<td>0.296</td>
</tr>
</tbody>
</table>

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

Let \( I \) different relation phrases, mapping between mentions and relation phrases: \( \Pi_L, \Pi_R \in \{0, 1\}^{M \times I} \)

Two bi-adjacency matrices for the subgraph

\[
W_L = \Pi_C^T \Pi_L \quad \text{and} \quad W_R = \Pi_C^T \Pi_R;
\]
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 $W_M \in \mathbb{R}^M \times M$

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

- 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.
Relation Phrase Clustering

- 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.

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Notation</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type signatures</td>
<td>( P_L, P_R \in \mathbb{R}^{l \times T} )</td>
<td></td>
</tr>
<tr>
<td>String features</td>
<td>( F_S \in \mathbb{R}^{l \times n_s} )</td>
<td></td>
</tr>
<tr>
<td>Context features</td>
<td>( F_C \in \mathbb{R}^{l \times n_c} )</td>
<td></td>
</tr>
</tbody>
</table>
Type Inference: A Joint Optimization Problem

\[ O_{\alpha, \gamma, \mu} = \mathcal{F}(C, P_L, P_R) + \mathcal{L}_\alpha(P_L, P_R, \{U^{(v)}, V^{(v)}\}, U^*) + \Omega_{\gamma, \mu}(Y, C, P_L, P_R). \] (2)

- **Type propagation between entity surface names and relation phrases** (Hypo 1)

\[ \mathcal{F}(C, P_L, P_R) = \sum_{i=1}^{n} \sum_{j=1}^{l} W_{L,ij} \left \| \frac{C_i}{\sqrt{D_{L,ii}}} - \frac{P_{L,j}}{\sqrt{D_{L,jj}}} \right \|^2 + \sum_{i=1}^{n} \sum_{j=1}^{l} W_{R,ij} \left \| \frac{C_i}{\sqrt{D_{R,ii}}} - \frac{P_{R,j}}{\sqrt{D_{R,jj}}} \right \|^2 \]

- **Mention modeling & mention correlation** (Hypo 2)

\[ \Omega_{\gamma, \mu}(Y, C, P_L, P_R) = \| Y - f(\Pi_C C, \Pi_L P_L, \Pi_R P_R) \|^2_F + \frac{\gamma}{2} \sum_{c \in C} \sum_{i,j=1}^{M_c} W_{ij}^{(c)} \left \| \frac{Y_i}{\sqrt{D_{ii}^{(c)}}} - \frac{Y_j}{\sqrt{D_{jj}^{(c)}}} \right \|^2 + \mu \| Y - Y_0 \|^2_F \]

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

\[ \mathcal{L}_\alpha(P_L, P_R, \{U^{(v)}, V^{(v)}\}, U^*) \]

\[ = \sum_{v=0}^{d} \beta^{(v)} (\| F^{(v)} - U^{(v)} V^{(v)T} \|^2_F + \alpha \| U^{(v)} Q^{(v)} - U^* \|^2_F). \]
The ClusType Algorithm

\[
\min_{\{Y, C, P, P_R, U^*, U^{(v)}, V^{(v)}, \beta^{(v)}\}} O_{\alpha, \gamma, \mu, \lambda_L, \lambda}\n\]

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

- 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
\[
Y^{(c)} = [(1 + \gamma + \mu)I_c - \gamma S^{(c)}]^{-1}(\Theta^{(c)} + \mu Y_0^{(c)}), \ \forall c \in C, (7)\]
\(C = \frac{1}{2} [S_L P_L + S_R P_R + \Pi_L^T (Y - \Pi_L P_L) - \Pi_R P_R]; \quad (8)\)
\(P_L = X_0^{-1} [S_L^T C + \Pi_L^T (Y - \Pi_L C - \Pi_L P_L) + \beta^{(0)} U^{(0)} V^{(0)T}];\)
\(P_R = X_1^{-1} [S_R^T C + \Pi_R^T (Y - \Pi_R C) - \Pi_R P_R + \beta^{(1)} U^{(1)} V^{(1)T}];\)

For each view, performs single-view NMF until converges
\[
V^{(v)}_{jk}^{(v)} = \frac{V^{(v)}_{jk} + \alpha \sum_{i=1}^t U^{(v)}_{ik} U^{(v)}_{ik}}{\Delta^{(v)}_{jk} + \alpha (\sum_{i=1}^t U^{(v)}_{ik})^2 (\sum_{i=1}^t V^{(v)}_{ik})}, \quad (9)\]
\[
U^{(v)}_{ik}^{(v)} = \frac{[F^{(v)} + V^{(v)} + \alpha U^*]_{jik}}{\sum_{v=0}^{d} \beta^{(v)} U^{(v)} Q^{(v)}}; \quad \beta^{(v)} = -\log \left( \frac{\delta^{(v)}}{\sum_{v=0}^{d} \delta^{(v)}} \right). \quad (12)\]

Until the objective converges
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

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

<table>
<thead>
<tr>
<th>Data sets</th>
<th>NYT</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>Method</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pattern [9]</td>
<td></td>
<td>0.4576</td>
<td>0.2247</td>
<td>0.3014</td>
<td>0.3790</td>
<td>0.1354</td>
<td>0.1996</td>
<td>0.2107</td>
<td>0.2368</td>
<td>0.2230</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIGER [16]</td>
<td></td>
<td>0.8668</td>
<td>0.8964</td>
<td>0.8814</td>
<td>0.5010</td>
<td>0.1237</td>
<td>0.1983</td>
<td>0.7354</td>
<td>0.1951</td>
<td>0.3084</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SemTagger [12]</td>
<td></td>
<td>0.8667</td>
<td>0.2658</td>
<td>0.4069</td>
<td>0.3769</td>
<td>0.2440</td>
<td>0.2963</td>
<td>0.4225</td>
<td>0.1632</td>
<td>0.2355</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>APOLLO [29]</td>
<td></td>
<td>0.9257</td>
<td>0.6972</td>
<td>0.7954</td>
<td>0.3534</td>
<td>0.2366</td>
<td>0.2834</td>
<td>0.1471</td>
<td>0.2635</td>
<td>0.1883</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NNPLB [15]</td>
<td></td>
<td>0.7487</td>
<td>0.5538</td>
<td>0.6367</td>
<td>0.4248</td>
<td>0.6397</td>
<td>0.5106</td>
<td>0.3327</td>
<td>0.1951</td>
<td>0.2459</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClusType-NoClus</td>
<td></td>
<td>0.9130</td>
<td>0.8685</td>
<td>0.8902</td>
<td>0.7629</td>
<td>0.7581</td>
<td>0.7605</td>
<td>0.3466</td>
<td>0.4920</td>
<td>0.4067</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClusType-NoWin</td>
<td></td>
<td>0.9244</td>
<td>0.9015</td>
<td>0.9128</td>
<td>0.7812</td>
<td>0.7634</td>
<td>0.7722</td>
<td>0.3539</td>
<td>0.5434</td>
<td>0.4286</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClusType-TwoStep</td>
<td></td>
<td>0.9257</td>
<td>0.9033</td>
<td>0.9143</td>
<td>0.8025</td>
<td>0.7629</td>
<td>0.7821</td>
<td>0.3748</td>
<td>0.5230</td>
<td>0.4367</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClusType</td>
<td></td>
<td>0.9550</td>
<td>0.9243</td>
<td>0.9394</td>
<td>0.8333</td>
<td>0.7849</td>
<td>0.8084</td>
<td>0.3956</td>
<td>0.5230</td>
<td>0.4505</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

46.08% and 48.94% improvement in F1 score compared to the best baseline on the Tweet and the Yelp datasets
Comparing on Different Entity Types

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

Table 6: F1 score comparison with trained NER.

<table>
<thead>
<tr>
<th>Method</th>
<th>NYT</th>
<th>Yelp</th>
<th>Tweet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanford NER [6]</td>
<td>0.6819</td>
<td>0.2403</td>
<td>0.4383</td>
</tr>
<tr>
<td>ClusType-NoClus</td>
<td>0.9031</td>
<td>0.4522</td>
<td>0.4167</td>
</tr>
<tr>
<td>ClusType</td>
<td>0.9419</td>
<td>0.5943</td>
<td>0.4717</td>
</tr>
</tbody>
</table>

- 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

Table 7: Example output of ClusType and the compared methods on the Yelp dataset.

<table>
<thead>
<tr>
<th>ClusType</th>
<th>SemTagger</th>
<th>NNPLBB</th>
</tr>
</thead>
<tbody>
<tr>
<td>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. ...</td>
<td>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. ...</td>
<td>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. ...</td>
</tr>
<tr>
<td>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.</td>
<td>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.</td>
<td>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.</td>
</tr>
</tbody>
</table>

- Extracts more mentions and predicts types with higher accuracy
- 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

Table 8: Example relation phrase clusters and their corpus frequency from the NYT dataset.

<table>
<thead>
<tr>
<th>ID</th>
<th>Relation phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>recruited by (5.1k); employed by (3.4k); want hire by (264)</td>
</tr>
<tr>
<td>2</td>
<td>go against (2.4k); struggling so much against (54); run for re-election against (112); campaigned against (1.3k)</td>
</tr>
<tr>
<td>3</td>
<td>looking at ways around (105); pitched around (1.9k); echo around (844); present at (5.5k);</td>
</tr>
</tbody>
</table>
Testing on Context Sparsity and Surface Name popularity

- **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

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

2. A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. EMNLP, 2011.


Reference III


27. Y. Li, C. Wang, F. Han, J. Han, D. Roth, and X. Yan. Mining Evidences for Named Entity Disambiguation, KDD, 2013.


34. S. Daitch, J. Kelner, and D. Spielman. Fitting a graph to vector data. ICML, 2009.


38. Xiang Ren, Ahmed El-Kishky, Chi Wang, Fangbo Tao, Clare R. Voss, Heng Ji, and Jiawei Han, ClusType: Effective Entity Recognition and Typing by Relation Phrase-Based Clustering, KDD 2015

39. Ahmed El-Kishky, Yanglei Song, Chi Wang, Clare R Voss, and Jiawei Han, Scalable Topical Phrase Mining from Text Corpora, VLDB 2015
Outline

1. Introduction to entity recognition and typing
2. Entity recognition – overview and phrase mining approach
3. Entity typing – overview and network mining approach
4. Trends and research problems
Conclusions

- Propose a fully automatic entity recognition and typing method for domain-specific corpora
  - Leverage minimal linguistic/domain assumption
  - Framework can be generalized to other languages
- Propose a novel relation phrase-based framework for distantly-supervised entity recognition and typing
  - A data-driven, domain-agnostic phrase mining algorithm
  - Candidate entity mentions and relation phrase generation
  - Integrate relation phrase clustering with type propagation on heterogeneous networks
  - Mutually enhance each other via solving a joint optimization problem
Looking Forward

- Extend to role discovery for scientific concepts
- Study of relation phrase clustering
  - joint entity/relation clustering
  - synonymous relation phrase canonicalization
- Lot of unanswered questions and research issues
  - What is the best framework for distantly-supervised entity typing?
  - Apply ClusType framework to other NLP applications
    - Relation/event detection and typing
    - Collective entity linking
  - Assist construction of quality heterogeneous information networks
Software

- Phrase Mining
  - SegPhrase: [https://github.com/shangjingbo1226/SegPhrase](https://github.com/shangjingbo1226/SegPhrase)
  - TopMine: [http://web.engr.illinois.edu/~elkishk2/code/ToPMine.zip](http://web.engr.illinois.edu/~elkishk2/code/ToPMine.zip)

- Entity Typing
  - ClusType: [http://shanzhenren.github.io/ClusType](http://shanzhenren.github.io/ClusType)

- Checking our research package dissemination portal