Leveraging Knowledge Graphs for Web Search

Part 2 - Named Entity Recognition and Linking to Knowledge Graphs

Gianluca Demartini

University of Sheffield

gianlucademartini.net

Entity Management and Search

- Entity Extraction: Recognize entity mentions in text

  "<u>Paris</u> has been changing over time"

- Entity Linking: Assign URIs to entities

  <http://dbpedia.org/resource/Paris>

- Indexing: Database vs Inverted Index

Uri	Label
http://dbpedia.org/resource/Paris	Paris, France
http://dbpedia.org/resource/Rome	Rome, Italy
http://dbpedia.org/resource/Berlin	Berlin, Germany

- Search: ranking entities given a query

  "Capital cities in Europe"

Outline

- A NLP Pipeline:
  - Named Entity Recognition
  - Entity Linking
  - Ranking entity types
- NER and disambiguation in scientific documents

- **Slides: gianlucademartini.net/kg**

Information extraction: entities

- Entity extraction / Named Entity Recognition
  - "Slovenia borders Italy"
- Entity resolution
  - "Apple released a new Mac".
  - From "Apple", "Mac"
  - To Apple_Inc., Macintosh_(computer)
- Entity classification
  - Into a set of predefined categories of interest
  - Person, location, organization, date/time, e-mail address, phone number, etc.
  - E.g. <"Slovenia", type, Country>

Steps

- Tokenization

- Sentence splitting

- Part-of-speech (POS) tagging

- Named Entity Recognition (NER) and linking

- Co-reference resolution

- Relation extraction

NN   NN   ADJ

"The cat is in London. **It** is nice."

Sentence 1   Sentence 2
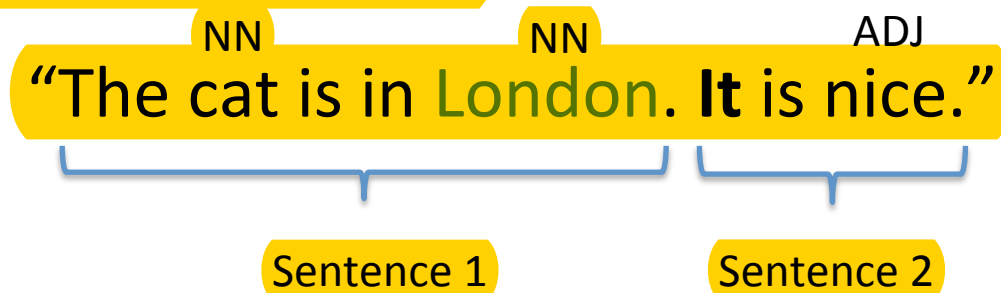
Entity Extraction

- Employed by most modern approaches
- Part-of-speech tagging
- Noun phrase chunking, used for entity extraction
- Abstraction of text
  - From: "Slovenia borders Italy"
  - To: "noun – verb – noun"

- Approaches to Entity Extraction:
  - Dictionaries
  - Patterns
 - Learning Models

NER methods

- Rule Based
  - Regular expressions, e.g. capitalized word + {street, boulevard, avenue} indicates location
  - Engineered vs. learned rules
- NER can be formulated as **classification** tasks
  - NE extraction: assign word mentions to tags (B beginning of an entity, I continues the entity, O word outside the entity)
  - NE classification: assign entity mentions to categories (Person, Organization, etc.)
  - Use ML methods for classification: Decision trees, SVM, AdaBoost
  - Standard classification assumes cases are disconnected (i.i.d)
- **Probabilistic sequence models**: HMM, CRF
  - Each token in a sequence is assigned a label
  - Labels of tokens are dependent on the labels of other tokens in the sequence particularly their neighbors (not i.i.d).

Classification

Generative p(y,x) **Naïve Bayes**
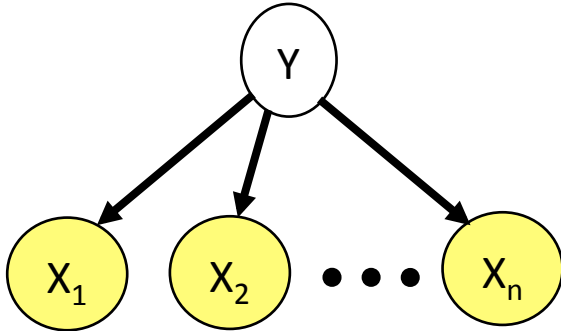
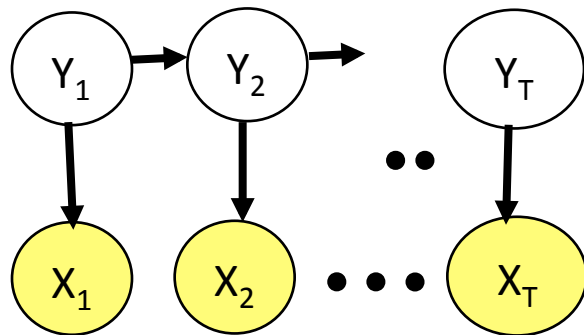**Conditional**
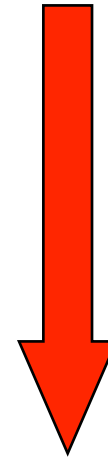
Discriminative p(y|x)

Logistic Regression

Sequence Labeling

Generative p(y,x) HMM

Conditional

Discriminative p(y|x)

Linear-chain CRF

Sunita Sarawagi and William W. Cohen. Semi-Markov Conditional Random Fields for Information Extraction. In NIPS, 2005.

NER features

- Gazetteers (background knowledge)
  - location names, first names, surnames, company names
- Word
  - Orthographic
    - initial-caps, all-caps, all-digits, contains-hyphen, contains-dots, roman-number, punctuation-mark, URL, acronym
  - Word type
    - Capitalized, quote, lowercased, capitalized
  - Part-of-speech tag
    - NP, noun, nominal, VP, verb, adjective
- Context
  - Text window: words, tags, predictions
  - Trigger words
    - Mr, Miss, Dr, PhD for person and city, street for location

Some NER tools

- Java
  - Stanford Named Entity Recognizer
    - http://nlp.stanford.edu/software/CRF-NER.shtml
  - GATE
    - http://gate.ac.uk/ http://services.gate.ac.uk/annie/
  - LingPipe http://alias-i.com/lingpipe/
- C
  - SuperSense Tagger
    - http://sourceforge.net/projects/supersensetag/
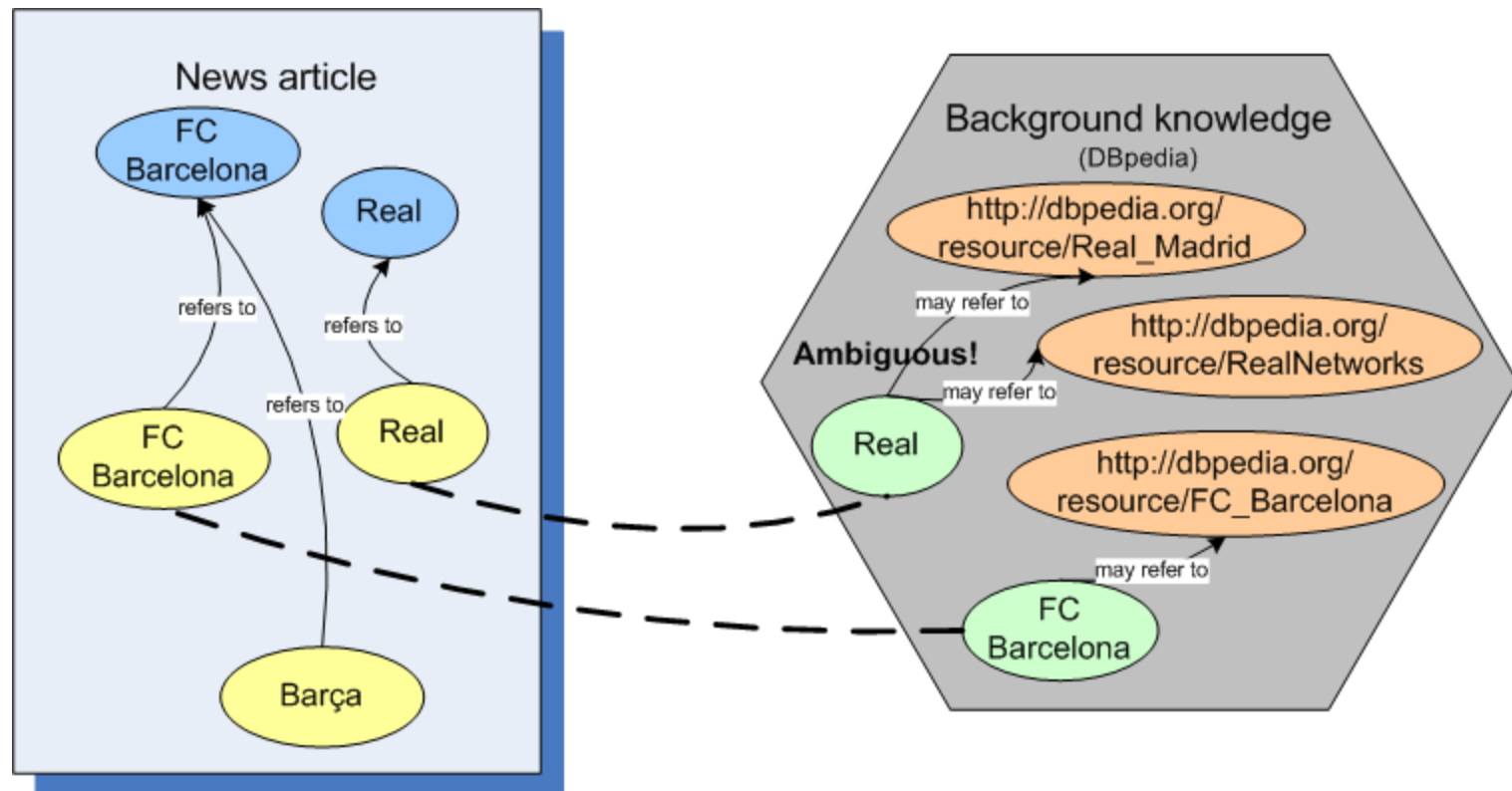- Python
  - NLTK: http://www.nltk.org
  - spaCy: http://spacy.io/

Entity Resolution / Linking

Basic situation

Pipeline

1. Identify named entity mentions in source text using a named entity recognizer

2. Given the mentions, gather candidate KB entities that have that mention as a label

3. Rank the KB entities

4. Select the best KB entity for each mention

Relatedness

- Intuition: **entities that co-occur in the same context tend to be more related**

- How can we express relatedness of two entities in a numerical way?

  – Statistical co-occurrence

  – Similarity of entities' descriptions
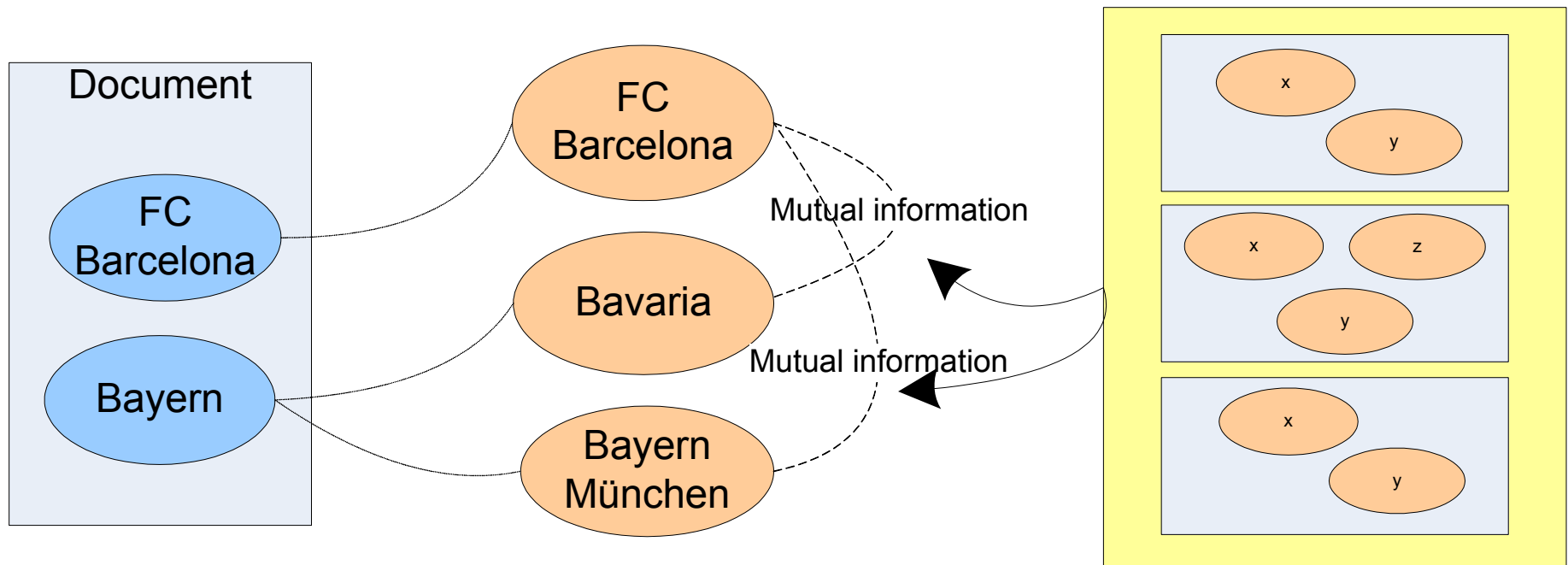
  – Relationships in the ontology

Semantic relatedness

- *If entities have an explicit assertion connecting them (or have common neighbours), they tend to be related*
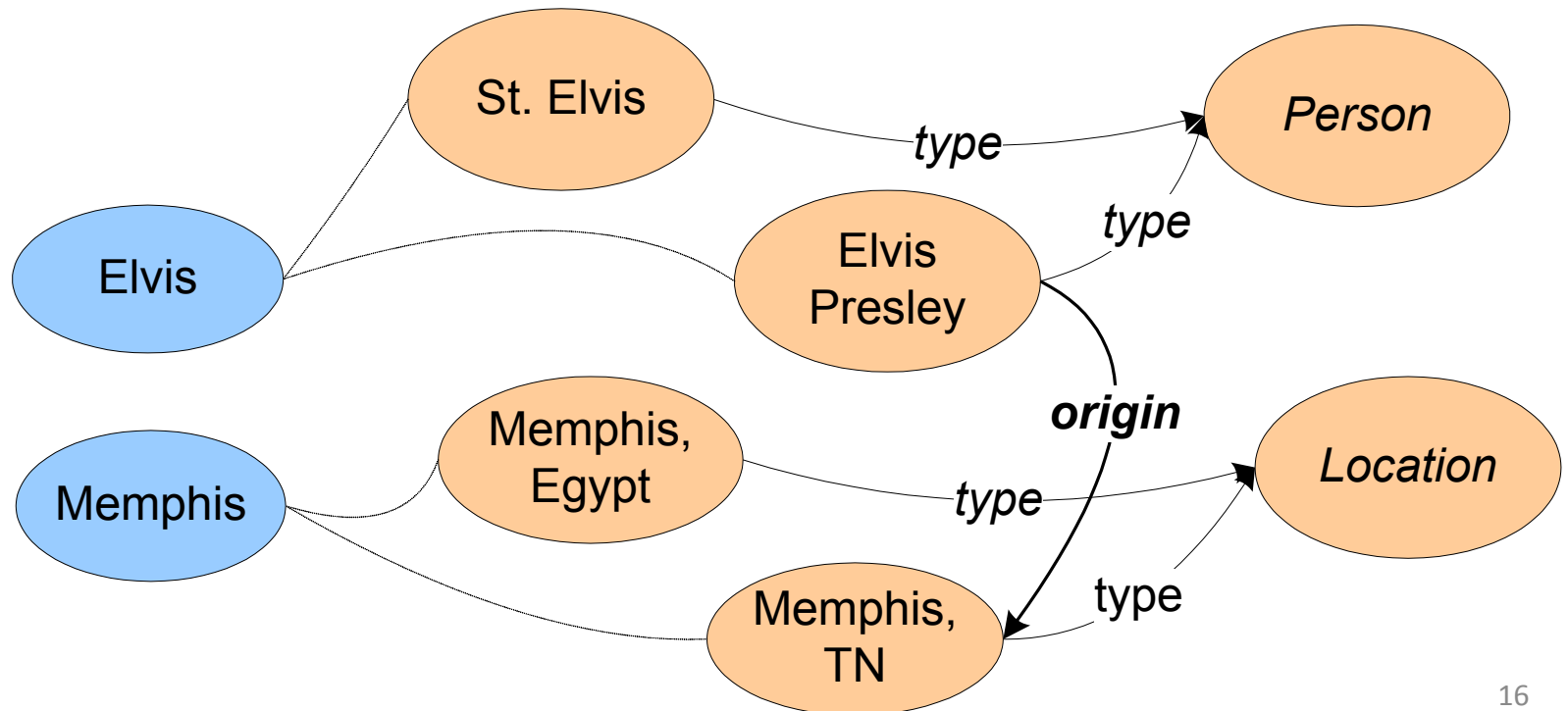
Co-occurrence as relatedness

- *If distinct entities occur together more often than by chance, they tend to be related*

Where do entities appear?

- Documents
  - Text in general
    - For example, news articles
    - Exploiting natural language structure and semantic coherence
  - Specific to the Web
    - Exploiting structure of web pages, e.g. annotation of web tables
      - Hazem Elmeleegy, Jayant Madhavan, Alon Y. Halevy: Harvesting relational tables from lists on the web. VLDB J. 20(2): 209-226 (2011)
- Queries
  - Short text and no structure

Entities in web search queries

- – ~70% of queries contain a named entity (*entity mention queries*)
  - brad pitt height
- – ~50% of queries have an entity focus (*entity seeking queries*)
  - brad pitt attacked by fans
- – ~10% of queries are looking for a class of entities
  - brad pitt movies
- [Pound et al, WWW 2010], [Lin et al WWW 2012]

Entities in web search queries

- Entity mention query = <entity> {+ <intent>}
  - Intent is typically an additional word or phrase to
    - Disambiguate, most often by type e.g. *brad pitt actor*
    - Specify action or aspect e.g. *brad pitt net worth, toy story trailer*
- Approaches for NER in queries
  - Matching keywords. [Blanco et al. ISWC 2013]
    - https://github.com/yahoo/Glimmer/
  - Matching aliases, i.e., look up entity names in the KG. Roi Blanco, Giuseppe Ottaviano and Edgar Meij. *Fast and space-efficient entity linking in queries.* WSDM 2015

Exercises

- 1) Write a piece of code that
  - Calls NER APIs to run over some text (e.g., http://nerd.eurecom.fr/documentation or https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki)
  - Create an inverted index of entities appearing in documents

- 2) Use the ERD dataset and try your own disambiguation idea (using step 1 results)
  - http://web-ngram.research.microsoft.com/ERD2014/

Entity Recognition and Disambiguation Challenge (at SIGIR 2014)

- Sample of Freebase KG
- Short text: web search queries from past TREC competitions
  - Winning approach: extract entities from search results for the query
- Long text: ClueWeb pages
  - Winning approach: supervised machine learning, training on Wikipedia
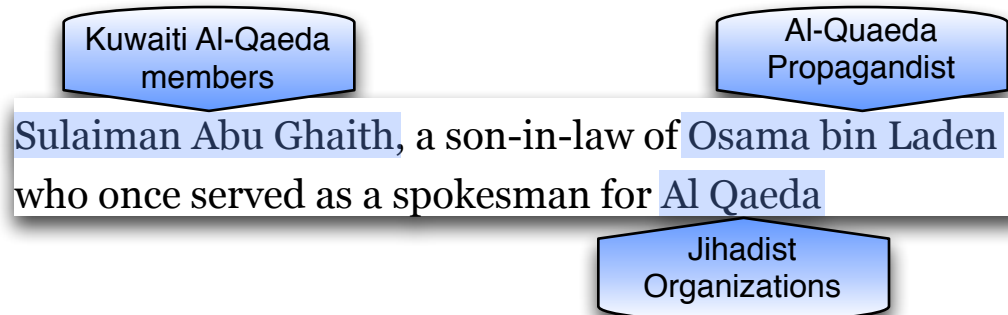
Entity Types

Alberto Tonon, Michele Catasta, Gianluca Demartini, Philippe Cudré-Mauroux, and Karl Aberer. **TRank: Ranking Entity Types Using the Web of Data**. In: The 12th International Semantic Web Conference (ISWC 2013)
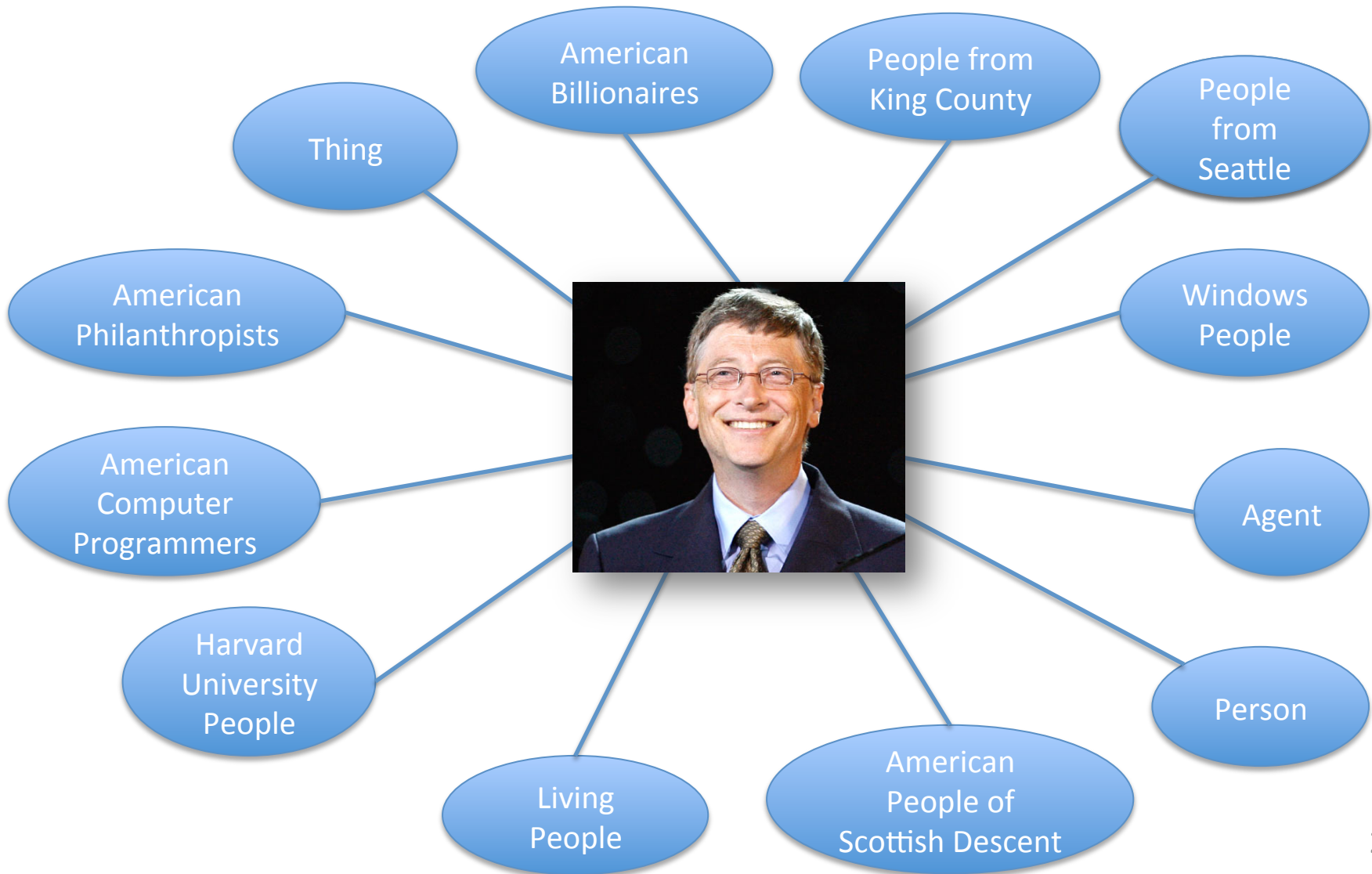
…and Why Types?

- "Summarization" of texts

Article Title	Entities	Types
Bin Laden Relative Pleads Not Guilty in Terrorism Case	Osama Bin Laden Abu Ghaith Lewis Kaplan Manhattan	Al-QaedaPropagandists Kuwaiti Al-Qaeda members Judge Borough (New York City)

- Contextual entities summaries in Web-pages

Kuwaiti Al-Qaeda members

Al-Quaeda Propagandist

Sulaiman Abu Ghaith, a son-in-law of Osama bin Laden who once served as a spokesman for Al Qaeda

Jihadist Organizations

- Disambiguation of other entities

- Diversification of search results

Entities May Have Many Types

Thing
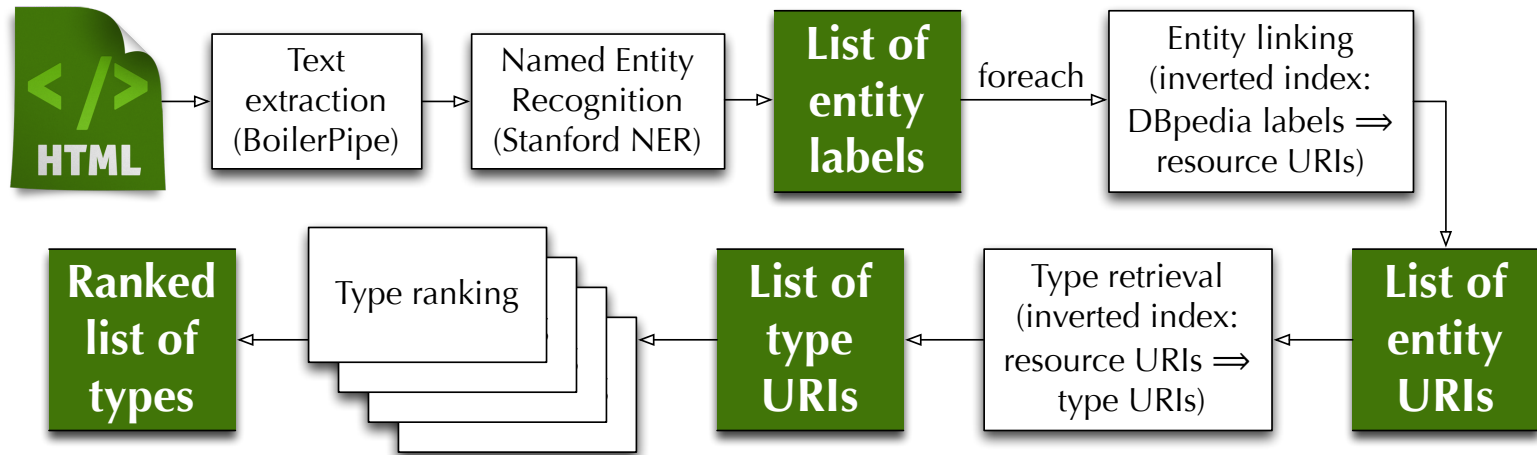
American Billionaires

People from King County

People from Seattle

American Philanthropists

Windows People

American Computer Programmers

Agent

Harvard University People

Person

Living People

American People of Scottish Descent

TRank Pipeline


```
HTML → Text extraction (BoilerPipe) → Named Entity Recognition (Stanford NER) → List of entity labels → foreach → Entity linking (inverted index: DBpedia labels ⟹ resource URIs) → List of entity URIs → Type retrieval (inverted index: resource URIs ⟹ type URIs) → List of type URIs → Type ranking → Ranked list of types
```
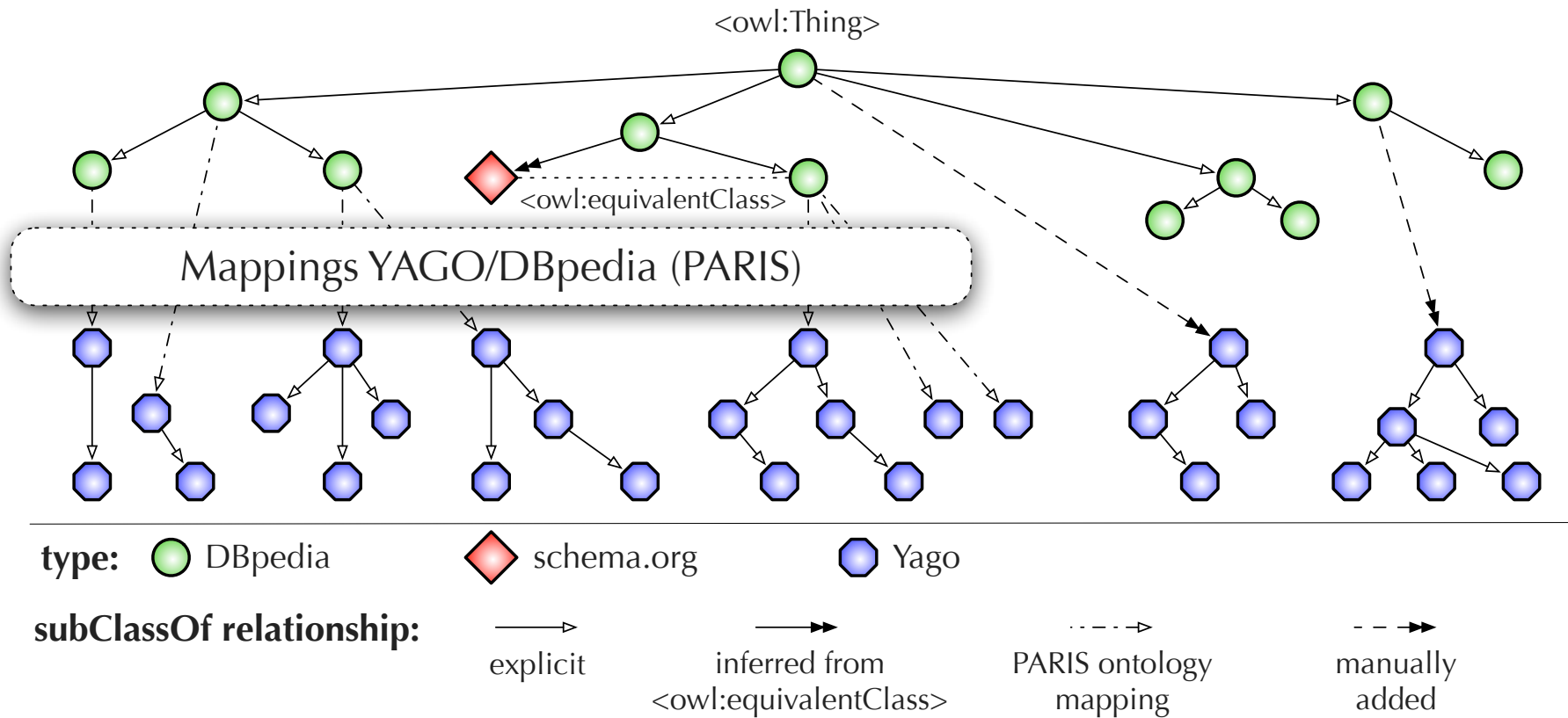
Alberto Tonon, Michele Catasta, Gianluca Demartini, Philippe Cudré-Mauroux, and Karl Aberer. TRank: Ranking Entity Types Using the Web of Data. In: The 12th International Semantic Web Conference (**ISWC 2013**). Sydney, Australia, October 2013.

Type Hierarchy

type: ● DBpedia ◆ schema.org ⬡ Yago

subClassOf relationship:

explicit | inferred from <owl:equivalentClass> | PARIS ontology mapping | manually added

Ranking Algorithms

- **Entity centric**
- **Hierarchy-based**
- **Context-aware (featuring type-hierarchy)**
- **Learning to Rank**

Hierarchy-Based Approaches
(An Example)

- **ANCESTORS**

Score(e, t) = number of *t*'s ancestors in the type hierarchy contained in T_e.
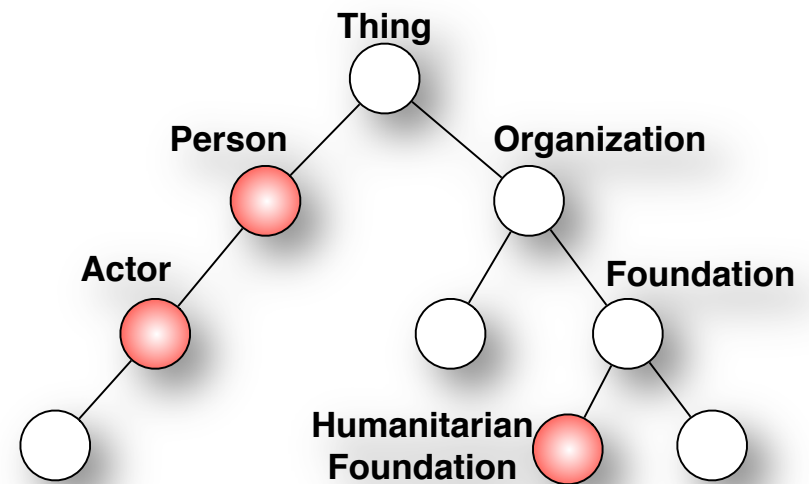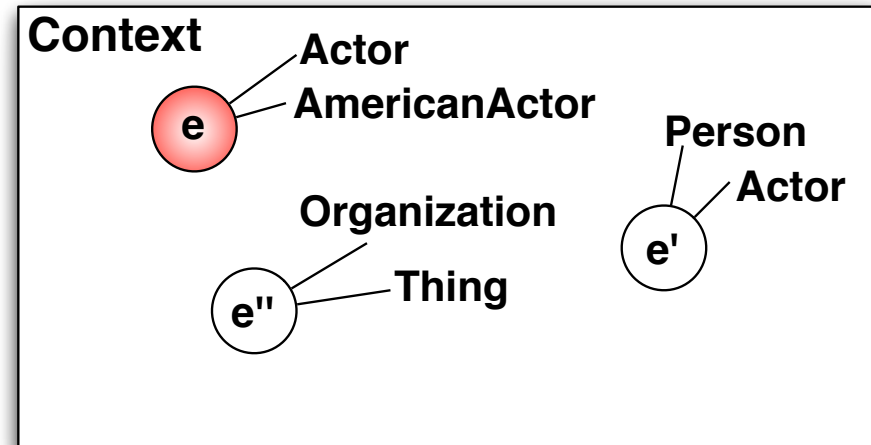

T_e often doesn't contain all super types of a specific type

Context-Aware Ranking Approaches (An Example)

- **SAMETYPE**

Score(e, t, c_T) = number of times *t* appears among the types of every other entity in c_T.

Learning to Rank Entity Types

Determine an optimal combination of all our approaches:

- Decision trees

- Linear regression models

- 10-fold cross validation

Datasets

- 128 recent NYTimes articles split to create:
  - *Entity Collection*
  - *Sentence Collection*
  - *Paragraph Collection*
  - *3-Paragraphs Collection*
- Ground-truth obtained by using crowdsourcing
  - *3 workers per entity/context*
 - *4 levels of relevance for each type*
  - *Overall cost: 190$*

Effectiveness Evaluation

Approach	Entity-only		Sentence		Paragraph		3-Paragraphs	
	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP
FREQ	0.6284	0.4659	0.5409	0.3758	0.5315	0.3739	0.5250	0.3577
WIKILINK-OUT	0.6874	0.5406	0.6050	0.4521	0.6063	0.4550	0.6059	0.4444
WIKILINK-IN	0.6832	0.5342	0.5907	0.4213	0.5879	0.4254	0.5853	0.4143
SAMEAS	0.6848	0.5328	0.6049	0.4310	0.5990	0.4221	0.6172	0.4417
LABEL	0.6672	0.5067	0.6075	0.4265	0.5883	0.4104	0.5821	0.4034
SAMETYPE	-	-	0.6024	0.4452	0.5917	0.4327	0.5813	0.4256
PATH	-	-	0.6507	0.4956	0.6538	0.4974	0.6315	0.4742
DEPTH	0.7432	0.6128	0.6754	0.5385	0.6797	0.5475	0.6741	0.5354
ANCESTORS	0.7424	0.6154	$0.6967^{\dagger}$	$0.5637^{\dagger}$	$0.6949^{\dagger}$	$0.5662^{\dagger}$	$0.6879^{\dagger}$	$0.5562^{\dagger}$
ANC_DEPTH	$0.7469^{\dagger}$	$0.6236^{\dagger}$	0.6832	0.5488	0.6885	0.5546	0.6796	0.5423
DEC-TREE	0.7614	0.6361	0.7373^{*}	0.6079^{*}	0.7979^{*}	0.7019^{*}	0.7943^{*}	0.6914^{*}
LIN-REG	0.7373	0.6079	0.6906	0.5579	0.6987	0.5702	0.6899	0.5529

Check our paper for a complete description of all the approaches we evaluated
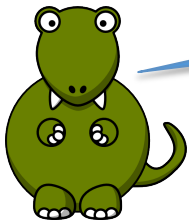
Avoiding SPARQL Queries with Inverted Indices and Map/Reduce
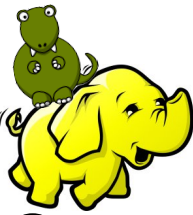
- TRank is implemented with *Hadoop* and *Map/Reduce*.
- All computations are done by using inverted indices:
  - Entity linking, Path index, Depth index
- CommonCrawl sample of 1TB, 1.3M web pages
  - Map/Reduce on a cluster of 8 machines with 12 cores, 32GB of RAM and 3 SATA disks
  - 25 min. processing time (> 100 docs/node x sec)

Text Extraction	NER	Entity Linking	Type Retrieval	Type Ranking
18.9%	35.6%	29.5%	9.8%	6.2%

- The inverted indices are publicly available at exascale.info/TRank

Using TRank

- Open Source (Scala)
  - https://github.com/MEM0R1ES/TRank
- Web Service (JSON)
  - http://trank.exascale.info

Extracting Scientific Concepts in Publications

Roman Prokofyev, Gianluca Demartini, and Philippe Cudré-Mauroux. **Effective Named Entity Recognition for Idiosyncratic Web Collections**. In: 23rd International Conference on World Wide Web (WWW 2014)

Problem Definition

1. INTRODUCTION

Nowadays, accessing information on the Internet through search engines has become a fundamental life activity. Current web search engines usually provide a ranked list of URLs to answer a query. This type of information access does a good job for dealing with simple navigational queries by leading users to specific websites. However, it is becoming increasingly insufficient for queries with vague or complex information need. Many queries serve just as the start of an exploration of related information space. Users may want to know about a topic from multiple aspects. Organizing the web content relevant to a query according to user intents would benefit user exploration. In addition, a list of URLs couldn't directly satisfy user information need. Users have
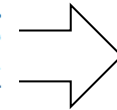
- search engine
- web search engine
- navigational query
- user intent
- information need
- web content
- …

Entity type: scientific concept

37

Traditional NER

Types:

- Maximum Entropy (Mallet, NLTK)
- Conditional Random Fields (Stanford NER, Mallet)

Properties:

- Require extensive training
- Usually domain-specific, different collections require training on their domain
- Very good at detecting such types as Location, Person, Organization
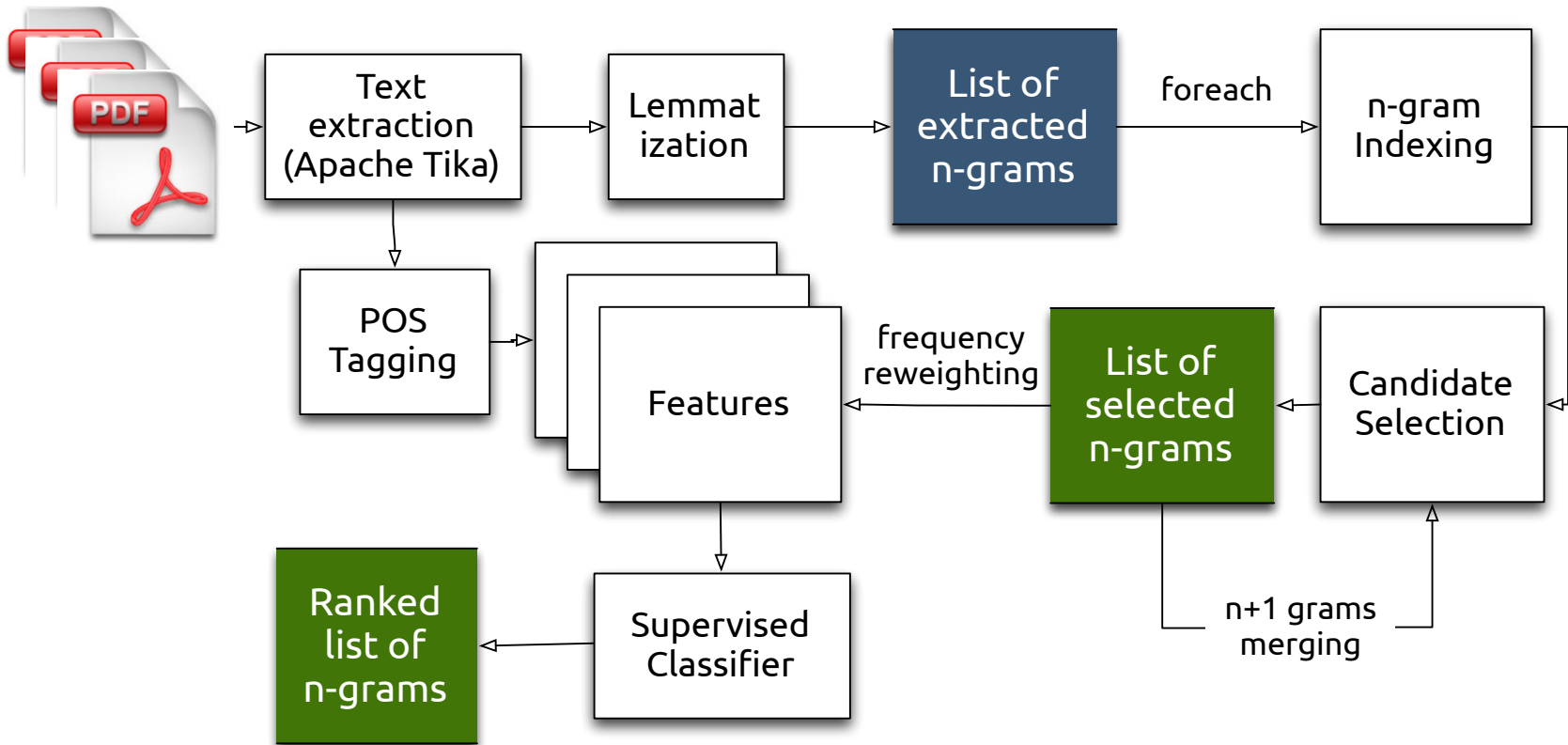
Proposed Approach

Our problem is defined as a classification task.

Two-step classification:

- Extract candidate named entities using frequency filtration algorithm.
- Classify candidate named entities using supervised classifier.

Candidate selection should allow us to greatly reduce the number of n-grams to classify, possibly without significant loss in Recall.

Pipeline

Text extraction (Apache Tika) → Lemmatization → List of extracted n-grams → foreach → n-gram Indexing

Text extraction → POS Tagging

POS Tagging → Features

Features → frequency reweighting → List of selected n-grams → Candidate Selection

n-gram Indexing → Candidate Selection

List of selected n-grams → n+1 grams merging → Candidate Selection

Features → Supervised Classifier → Ranked list of n-grams

Candidate Selection: Part I

Consider all bigrams with frequency > k (k=2):

```
candidate named:      5
entity are:           4
entity candidate:     3
entity in:            18
entity recognition:   12
named entity:         101
of named:             10
that named:           3
the named:            4
```
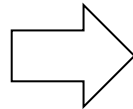
⇒
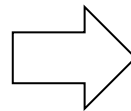
```
candidate named:      5
entity candidate:     3
entity recognition:   12
named entity:         101
```
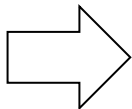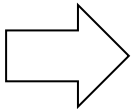
NLTK stop word filter

Candidate Selection: Part II

Trigram frequency is looked up from the n-gram index.

```
candidate named:      5
entity candidate:     3
entity recognition:  12
named entity:        101
```

```
candidate named entity: 5
named entity candidate: 3
named entity recognition: 12
named entity: 101
candidate named: 5
entity candidate: 3
entity recognition: 12
```

```
candidate named entity: 5
named entity candidate: 3
named entity recognition: 12
named entity: 81
candidate named: 0
entity candidate: 0
entity recognition: 0
```

Candidate Selection: Discussion

Possible to extract n-grams (n>2) with frequency ≤k
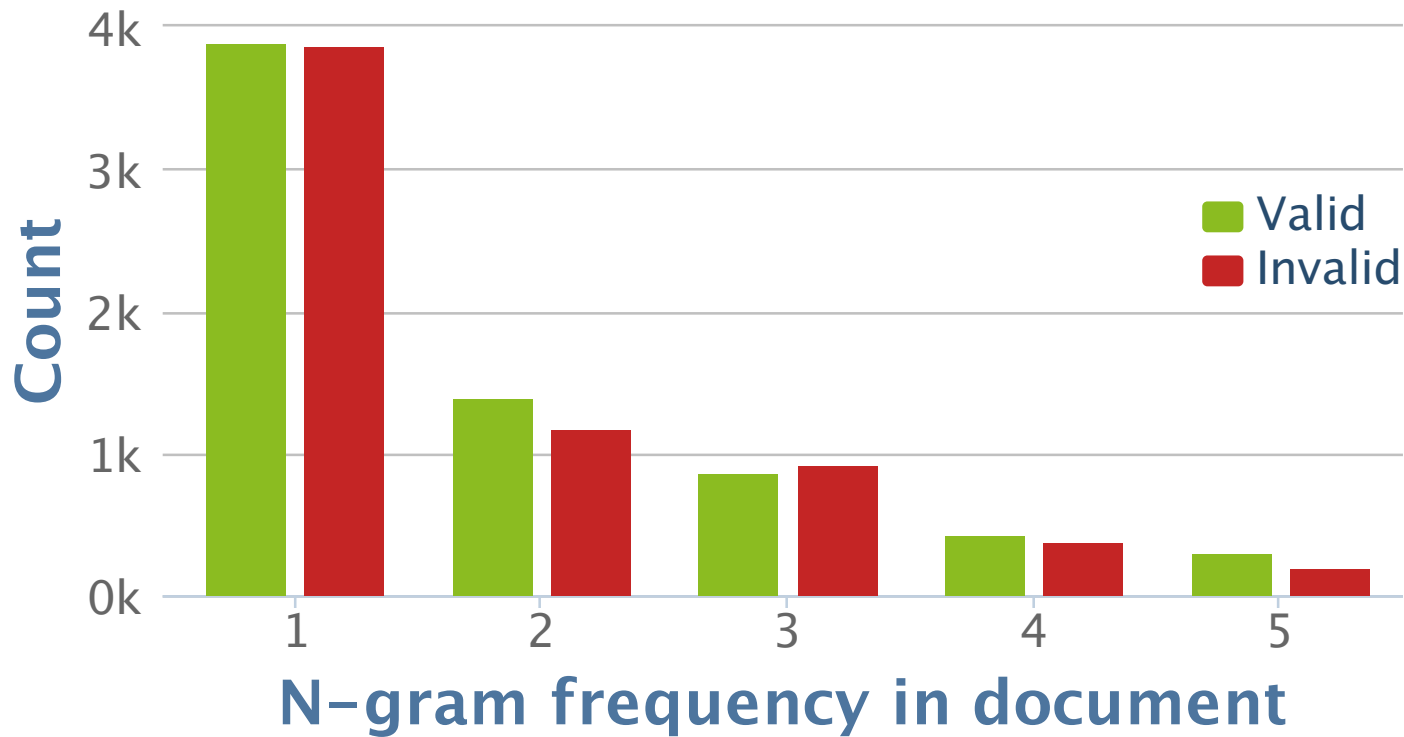
After Candidate Selection

ABSTRACT

Many private and/or public organizations have been reported to create and monitor targeted Twitter streams to collect and understand users' opinions about the organizations. Targeted Twitter stream is usually constructed by filtering tweets with user-defined selection criteria (*e.g.*, tweets published by users from a selected region, or tweets that match one or more predefined keywords). Targeted Twitter stream is then monitored to collect and understand users' opinions about the organizations. There is an emerging need for early crisis detection and response with such target stream. Such applications require a good named entity recognition (NER) system for Twitter, which is able to automatically discover emerging named entities that is potentially linked to the crisis. In this paper, we present a novel 2-step unsupervised NER system for targeted Twitter stream, called TwiNER. In the first step, it leverages on the global context obtained from Wikipedia and Web N-Gram corpus to partition tweets into valid segments (phrases) using a dynamic programming algorithm. Each such tweet segment is a candidate named entity. It is observed that the named entities in the targeted stream usually exhibit a gregarious property, due to the way the targeted stream is constructed. In the second step, TwiNER constructs a random walk model to exploit the gregarious property in the local context derived from the Twitter stream. The highly-ranked segments have a higher chance of being true named entities. We evaluated TwiNER on two sets of real-life tweets simulating two targeted streams. Evaluated using labeled ground truth, TwiNER achieves comparable performance as with conventional approaches in both streams. Various settings of TwiNER have also been examined to verify our global context + local context combo idea.

TwiNER: named entity recognition in targeted twitter stream

'SIGIR 2012

44

Classifier: Overview

Machine Learning algorithm:

Decision Trees from scikit-learn package.

Feature types:

- POS Tags and their derivatives
- External Knowledge Bases (DBLP, DBPedia)
- DBPedia relation graphs
- Syntactic features

Datasets

Two collections:

- CS Collection (SIGIR 2012 Research Track): 100 papers
- Physics collection: 100 papers randomly selected from arXiv.org High Energy Physics category

	CS Collection	Physics Collection
N# Candidate N-grams	21 531	18 129
N# Judged N-grams	15 057	11 421
N# Valid Entities	8 145	5 747
N# Invalid N-grams	6 912	5 674

Available at: github.com/XI-lab/scientific_NER_dataset
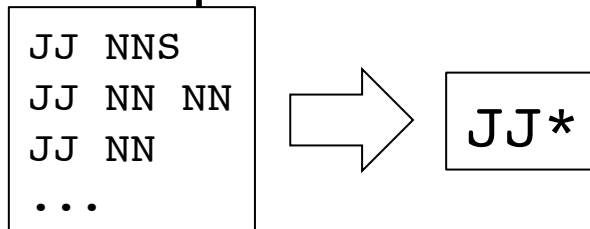
Features: POS Tags, part I
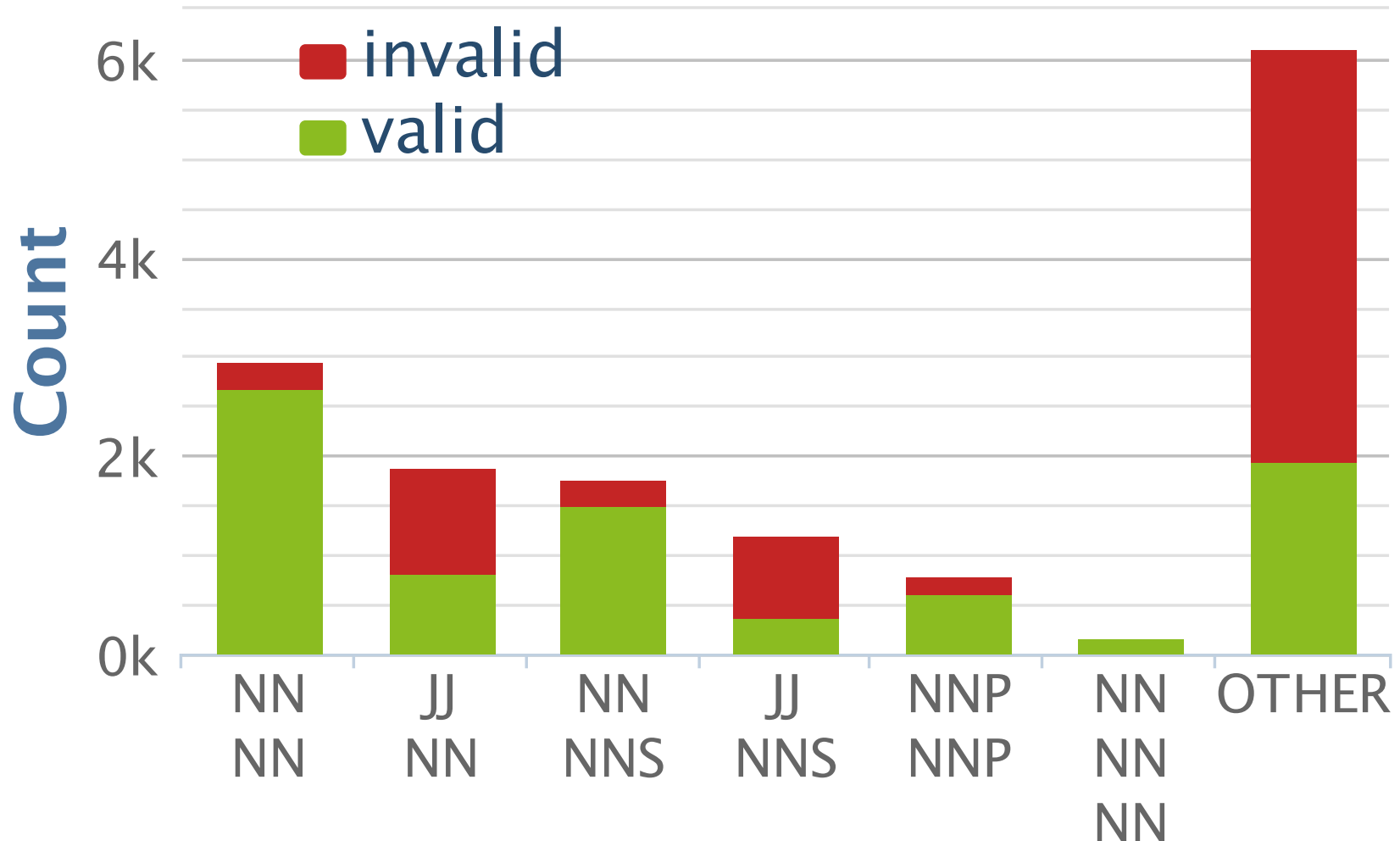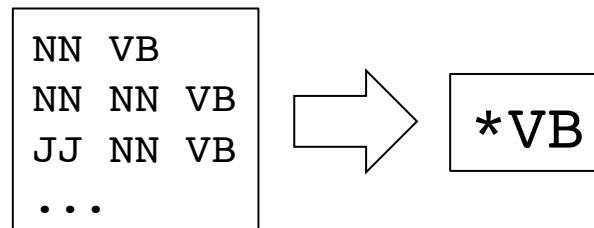

100+ different tag patterns

Features: POS Tags, part II

Two feature schemes:

- Raw POS tag patterns, each tag is a binary feature

- Regex POS tag patterns:
  - First tag match, for example:

```
JJ NNS
JJ NN NN        ⇒    JJ*
JJ NN
...
```

  - Last tag match:

```
NN VB
NN NN VB        ⇒    *VB
JJ NN VB
...
```

Features: External Knowledge Graphs

Domain-specific knowledge graphs:

- DBLP (Computer Science): contains author-assigned keywords to the papers
- ScienceWISE: high-quality scientific concepts (mostly for Physics domain) http://sciencewise.info

We perform exact string matching with these KGs.

Features: DBPedia, part I

DBPedia pages essentially represent valid entities

But there are a few problems when:
- N-gram is not an entity
- N-gram is not a scientific concept ("Tom Cruise" in IR paper)

	CS Collection		Physics Collection	
	Precision	Recall	Precision	Recall
Exact string matching	0.9045	0.2394	0.7063	0.0155
Matching with redirects	0.8457	0.4229	0.7768	0.5843

Features: Syntactic

Set of common syntactic features:

- N-gram length **in words**
- Whether n-gram is uppercased
- The number of other n-grams a given n-gram is part of

Experiments: Overview

1. Regex POS Patterns vs Normal POS tags
2. Redirects vs Non-redirects
3. Feature importance scores
4. MaxEntropy comparison

All results are obtained using average with 10-fold cross-validation.

Experiments: Comparison I

CS Collection	Precision	Recall	F1 score	Accuracy	N# features
Normal POS + Components	**0.8794**	0.8058*	0.8409*	0.8429*	54
Regex POS + Components	0.8475*	0.8524*	0.8499*	0.8448*	9
Normal POS + Components-Redirects	0.8678*	0.8305*	0.8487*	0.8473	50
Regex POS + Components-Redirects	0.8406*	**0.8769**	**0.8584**	**0.8509**	**7**

The symbol * indicates a statistically significant difference as compared to the approach in bold.

Experiments: Feature Importance

	Importance
NN STARTS	0.3091
DBLP	0.1442
Components + DBLP	0.1125
Components	0.0789
VB ENDS	0.0386
NN ENDS	0.0380
JJ STARTS	0.0364

	Importance
ScienceWISE	0.2870
Component + ScienceWISE	0.1948
Wikipedia redirect	0.1104
Components	0.1093
Wikilinks	0.0439
Participation count	0.0370

CS Collection, 7 features

Physics Collection, 6 features

Experiments: MaxEntropy

MaxEnt classifier receives full text as input.
(we used a classifier from NLTK package)

Comparison experiment: 80% of CS Collection as a training data, 20% as a test dataset.

	Precision	Recall	F1 score
Maximum Entropy	0.6566	0.7196	0.6867
Decision Trees	0.8121	0.8742	0.8420

Lessons Learned

Classic NER approaches are not good enough for Idiosyncratic Web Collections
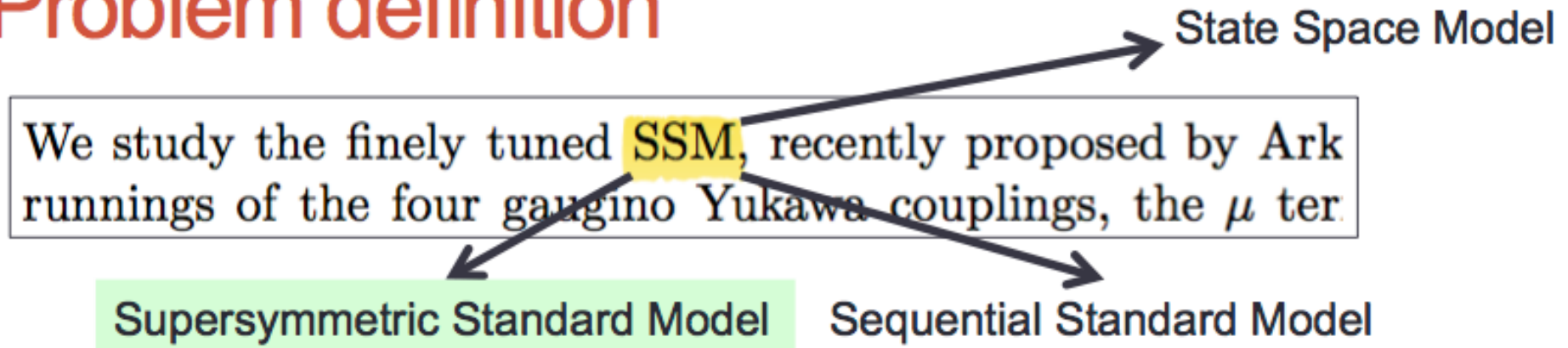
Leveraging the graph of scientific concepts is a key feature

Domain specific KBs and POS patterns work well

Experimental results show up to 85% accuracy over different scientific collections
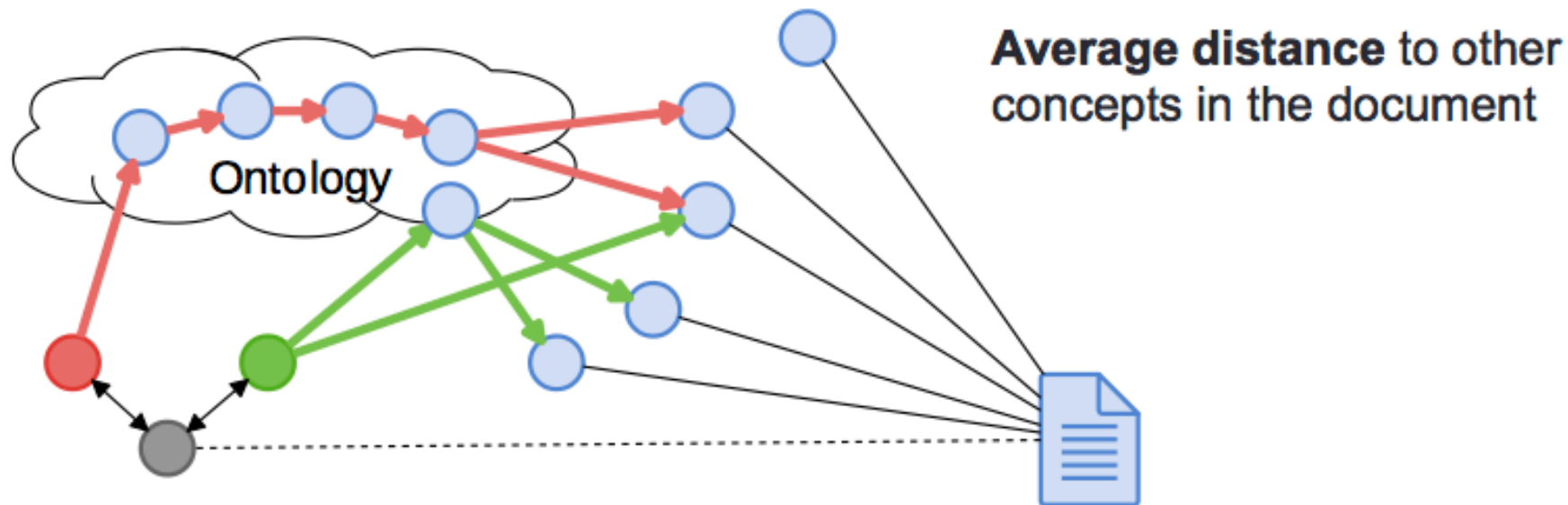
Entity Disambiguation
in Scientific Literature

Problem definition

State Space Model

We study the finely tuned **SSM**, recently proposed by Ark
runnings of the four gaugino Yukawa couplings, the μ ter

Supersymmetric Standard Model   Sequential Standard Model

- Using a background concept graph

Roman Prokofyev, Gianluca Demartini, Philippe Cudré-Mauroux, Alexey Boyarsky, and Oleg Ruchayskiy. Ontology-Based Word Sense Disambiguation in the Scientific Domain. In: 35th European Conference on Information Retrieval (ECIR 2013).

Ontology shortest path

Average distance to other concepts in the document

Ontology

Nearest neighbors

Co-occurring **1-hop neighbors** from the ontology
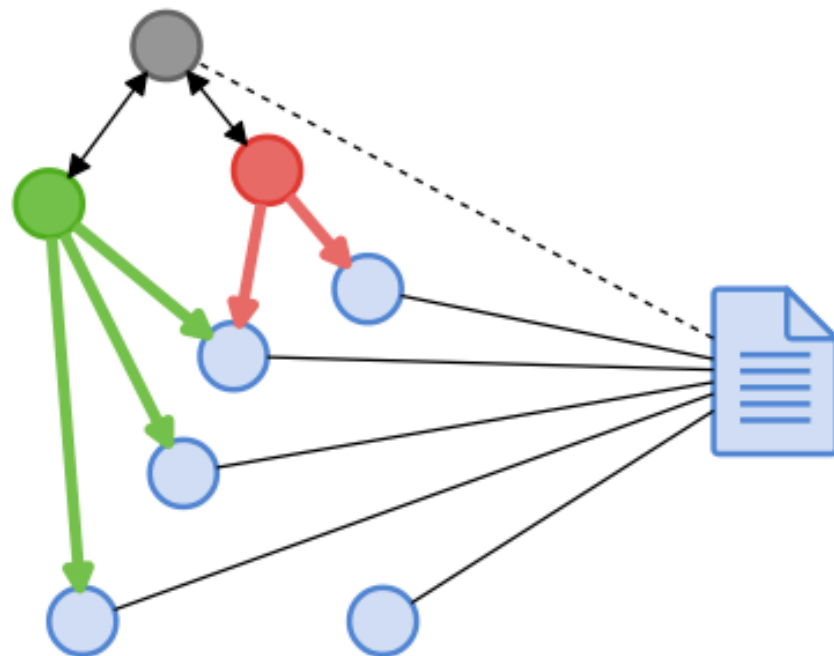
Summary

- NLP Pipeline:
  - Named Entity Recognition
  - Entity Linking
  - Ranking Entity Types
- NER and disambiguation in scientific documents
- Tomorrow
  - Searching for entities
  - Human Computation for better effectiveness