Occurrence statistics of entities on the web

Seminar Presentation

Aman Madaan

Indian Institute of Technology Bombay, Mumbai

May 2nd, 2014

Motivation

- Web Search is a quest for Structure
- The open web is huge, **1.8 billion** indexed web pages[1], but unstructured
- The knowledge is scattered around in pieces
- A user needs to carve the structure out of the web

[1] As of 31st March 2014

"Words that you use when you are doing the search, well they aren't *just words*, they refer to **real** things in the world"

— Jack Menzel, Product Management Director, Google Knowledge Graph

Structuring the web

- Structuring the web : A web page is more than just a bundle of strings
- Learn what the text is all about
- Go beyond web of strings to the web of entities

"Albert Einstein"

- Beyond the tricks like finding pages having the **String** "Albert Einstein", pages which link to pages having the **String** "Albert Einstein"

Words are not just words

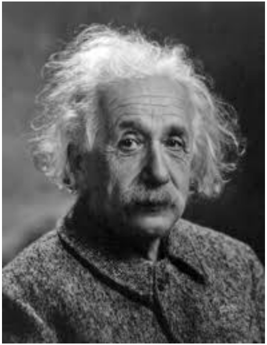


Figure: Albert Einstein

Words are not just words

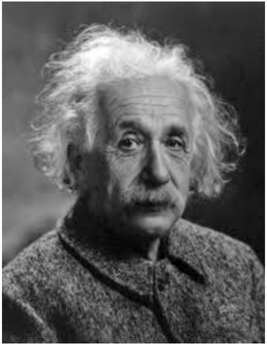


Figure: Albert Einstein

Words are not just words

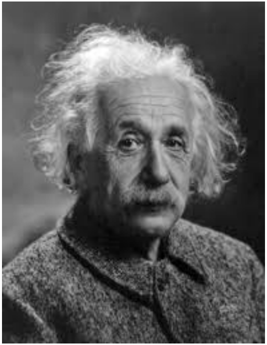


Figure: Albert Einstein

Words are not just words

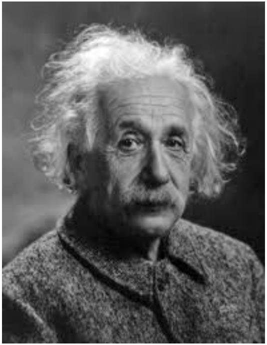


Figure: Albert Einstein

- **Born :** 1879, Germany
- **Died :** 1955, US

Words are not just words

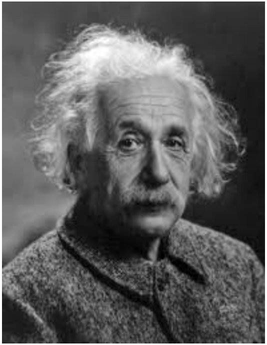


Figure: Albert Einstein

- **Born** : 1879, Germany
- **Died** : 1955, US

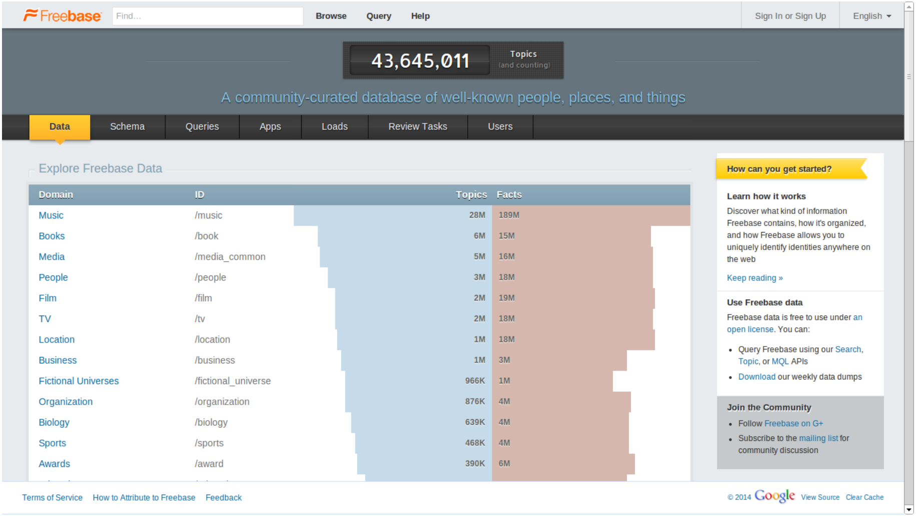
"Entities" *like* Albert Einstein

- **Born** : Issac Newton
- **Died** : Stephen Hawking

Knowledge Bases

- Want to make the web smarter by *understanding* the content
- *What* Entities? *Which* Relations?
- *Encyclopedia that a Computer can understand*
- A standard reference set of entities, relations, type hierarchies
- **Wordnet** The maiden knowledge base, has clean type system but limited entity base
- **Wikipedia** Huge, crowd sourced, but extremely loose and vague type systems
- Several knowledge bases have emerged as middle ground

Freebase



Freebase
Find...
Browse
Query
Help
Sign In or Sign Up
English
43,645,011
Topics
(and counting)
A community-curated database of well-known people, places, and things
Data
Schema
Queries
Apps
Loads
Review Tasks
Users
Explore Freebase Data
Domain
ID
Topics
Facts
Music /music 28M 189M
Books /book 6M 15M
Media /media_common 5M 16M
People /people 3M 18M
Film /film 2M 19M
TV /tv 2M 18M
Location /location 1M 18M
Business /business 1M 3M
Fictional Universes /fictional_universe 966K 1M
Organization /organization 876K 4M
Biology /biology 639K 4M
Sports /sports 468K 4M
Awards /award 390K 6M
How can you get started?
Learn how it works
Discover what kind of information Freebase contains, how it's organized, and how Freebase allows you to uniquely identify identities anywhere on the web
Keep reading »
Use Freebase data
Freebase data is free to use under an open license. You can:
Query Freebase using our Search, Topic, or MQL APIs
Download our weekly data dumps
Join the Community
Follow Freebase on G+
Subscribe to the mailing list for community discussion
Terms of Service
How to Attribute to Freebase
Feedback
© 2014 Google
View Source
Clear Cache

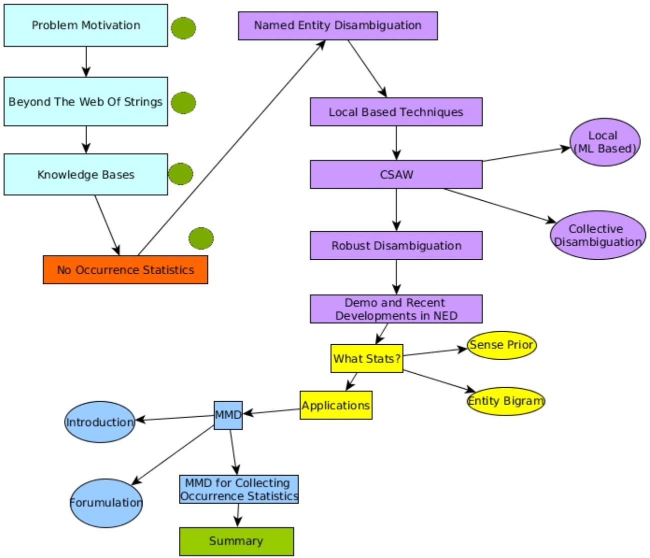
Freebase

- Freebase relies on crowd sourcing for creation of a rich but clean knowledge base
- The development of Freebase follows the same chain as Wikipedia, with users flagging issues, and cleaning and augmenting information
- Freebase also provides access to itself using web APIs.

Knowledge Bases : Not there yet

- None of the knowledge bases provides entity priors of any kind
- Co-occurrence statistics are also missing
- These pieces of information are really crucial for a number of tasks related like querying knowledge graphs.
- We motivate the need for such statistics after reviewing named entity disambiguation techniques.

Seminar Outline



Named Entity Disambiguation

Definition

NED aims to map mentions of ambiguous names in natural language onto a set of known entities (e.g. YAGO or DBpedia).[a]

[a] From (Efficient Entity Disambiguation via Similarity Hashing)

Running Example

Figure: The Problem Of Named Entity Disambiguation

Recognition and Tagging : Two step problem

Michael Jordan is a Professor at Berkeley

Recognition and Tagging : Two step problem

Michael Jordan is a Professor at Berkeley

- Step 1 : **Identify** entities

  Michael Jordan_PERSON is a professor at Berkeley_INSTITUTION

Recognition and tagging : Two step problem

Michael Jordan is a Professor at Berkeley

- Step 1 : **Identify** entities

  Michael Jordan_PERSON is a professor at Berkeley_INSTITUTION

- Step 2 : **Link** entities to knowledge bases :

  Michael Jordan_ENTITY (`http://en.wikipedia.org/wiki/Michael_I._Jordan`) is a professor at Berkeley_ENTITY (`http://en.wikipedia.org/wiki/University_of_California,_Berkeley`)

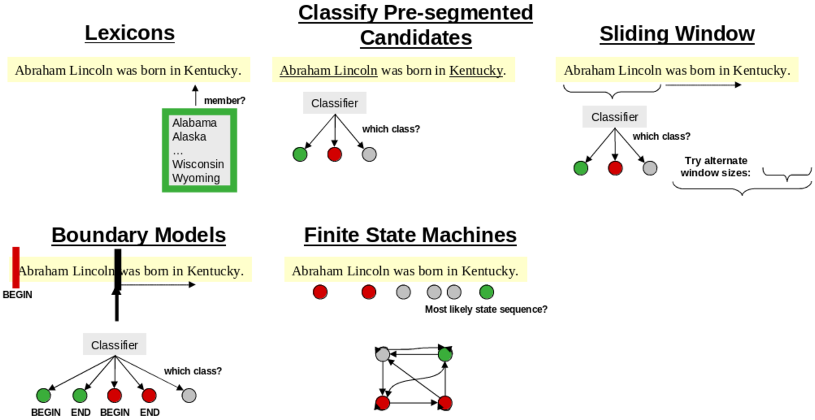
NER : Problem Statement

Definition (Named entity recognition[a])

[a] from 4

Named-entity recognition (NER) (also known as entity identification and entity extraction) is a subtask of information extraction that seeks to locate and classify atomic elements in text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.

NER : Solutions



[2]

---

[2] from 3

NER as a sequence labeling problem

- Observation sequence : Text
- State sequence : Labeling of the sequence with elements in (PER, LOC, ORG) etc.
- Find $\text{argmax}_S P(S|O)$
- Candidates : HMM, MEMM, CRF

NER as a sequence labeling problem

- **HMM**
  - Generative
  - Makes strong independence assumption
  - Myopic (Refer to label bias problem in William Cohen's Survey)
- **MEMM**
  - Discriminative
  - No independence assumptions are made, by formulation
  - Allows the use of feature functions
  - Myopic
- **CRF**
  - Discriminative
  - MEMM + non myopic, avoids local normalization
  - Talks of "compatibility", not independence (CS 728)

Named Entity Disambiguation : Outline

- Techniques
  - Local Disambiguation
  - Collective Disambiguation
  - Robust Disambiguation of Named Entities
- Quick Demo

Local Disambiguation

- Resolve each mention oblivious to the other disambiguations
- Need to disambiguate a mention by collecting the local evidences
- **Evidences** POS tags, gender information, dictionary lookup
- **Local** We cannot use the disambiguation information for any of the other entities for solving the problem
- Techniques
  - Machine Learning Based
  - Rule Based
  - Recent Rule based

Local Disambiguation : Rule Based

- Stems from the classical problem of word sense disambiguation
- Example : Lesk's Algorithm

Lesk's Algorithm

For each mention, pick the candidate sense for which there is maximum overlap in the gloss (definition) of the candidate and the context

- Note that the possible mentions are those that are identified by the named entity recognizer

Local Disambiguation : Rule Based (Example)

- Consider the same example

Figure: Disambiguating "Page"

- Disambiguating "Page"

Local Disambiguation : Rule Based (Example)

- **Jimmy Page**[3] James Patrick "Jimmy" Page, OBE (born 9 January 1944) is an English musician, songwriter and record producer who achieved international success as the guitar player and leader of the rock band Led Zeppelin.
- **Larry Page**[4] Lawrence "Larry" Page[2] (born March 26, 1973) is an American Business magnate and computer scientist who is the co-founder of Google, alongside Sergey Brin. On April 4, 2011, Page succeeded Eric Schmidt as the chief executive officer of Google.[3][4] As of 2014, Page's personal wealth is estimated to be US$32.3 billion, ranking him #19 on the Forbes list of billionaires.[1]
- **Context** played kashmir at Knebworth, his Les paul was uniquely tuned

Pick the candidate that is most likely given the context

[3] First para of the Wikipedia Entry
[4] First para of the Wikipedia Entry

Local Disambiguation : Rule based : Drawbacks

- Context can be misleading *Amazon* saw a flood of visitors
- Context can be insufficient (or even absent!)
- Rule based disambiguation has made a comeback with AIDA
- ML based local disambiguation to come

Collective Annotation of WikiPedia Entities in Web Text

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti

Key Intuition : Topical Coherence

- A document is usually about one topic
- Disambiguating each entity using the local clues misses out on a major piece of information : Topic of a page
- A page is usually has one topic, you can expect all the entities to be *related* to the topic *somehow*

Michael Jackson : 30 Disambiguations
John Paul : 10 disambiguations
But if they are mentioned on the **same page**, the page is most likely about Christianity, A big hint towards disambiguating **both** of them

Topical Coherence

Challenges

- Capturing local compatibility
  - Create a scoring function to rank possible candidates
- Inculcating topical coherence in the overall objective
  - Define Topical coherence

Local compatibility

- s : Spot, an Entity to be disambiguated (Christian leader John Paul)
- γ : An entity label value (`http://en.wikipedia.org/wiki/Po-pe_John_Paul_II`)
- $f_s(\gamma)$: A feature function that creates a vector of features

Local compatibility : Feature design

- 1. Take
  - Text from the first descriptive paragraph of γ
  - Text from the whole page for γ
  - Anchor text within Wikipedia for γ.
  - Anchor text and 5 tokens around γ
- 2. Apply each of the following operation with one argument as Spot
  - Dot-product between word count vectors
  - Cosine similarity in TFIDF vector space
  - Jaccard similarity between word sets

Total 12 Features (3 operations, 4 argument pairs) + Sense Probability Prior[5]

[5]Obtained by counting intra wiki links

Compatibility Score

- Local compatibility score between a spot s and a candidate is given by $w^T f_s(\gamma)$
- Thus, candidate is picked by $argmax_{\gamma \in \Gamma} w^T f_s(\gamma)$
- w is trained using an SVM like training objective

$$\text{Minimize } ||w||^2 + C\Sigma_s \varepsilon_s \text{ under the constraints}$$
$$w^T f_s(\gamma) - w^T f_s(\gamma) \geq 1 - \varepsilon_s$$

Defining topic Relatedness

- We need some notion of capturing the fact that 2 topics are related to each other
- Given
  - $g(\gamma)$: Set of wikipedia pages that link to γ
  - c : Total number of Wikipedia pages
  - $r(\gamma, \gamma')$: Relatedness of topics γ and γ'
- Define $r(\gamma, \gamma') = \frac{log|g(\gamma) \bigcap g(\gamma')| - log(max\{|g(\gamma)|, |(\gamma')|\})}{logc - log(min\{|g(\gamma)|, |(\gamma')|\})}$ (The Milne and Witten Score)

The Dominant Topic Model

- Need to define a collective score based on pairwise topical coherence of all γ_s used for labeling.
- The pairwise topical coherence, $r(\gamma_s, \gamma_s')$ is as defined above.
- For a page, overall topical coherence :

$$\Sigma_{s \neq s' \in S_0} r(\gamma_s, \gamma_s')$$

- Can be written as clique potential as in case of node potential

$$exp(\Sigma_{s \neq s' \in S_0} r(\gamma_s, \gamma_s'))$$

The Optimization objective

$$\frac{1}{\binom{|S_0|}{2}} \Sigma_{s \neq s' \in S_0} r(\gamma_s, \gamma'_s) + \frac{1}{|S_0|} \Sigma_{s \in S_0} w^T f_s(\gamma)$$

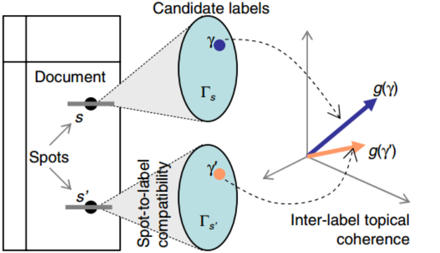


Figure 3: Labels $\gamma \in \Gamma_s, \gamma' \in \Gamma_{s'}$ have to be chosen for spots s, s' to maximize a combination of spot-to-label compatibility scores $\mathrm{NP}_s(\gamma), \mathrm{NP}_{s'}(\gamma')$ as well as topical similarity between γ and γ', say, $g(\gamma)^\top g(\gamma')$. [6]

[6] From 1

Solving the optimization objective

- LP rounding approach
- Hill climbing

```
initialize some assignment y^(0)
for k = 1, 2, ... do
    select a small spot set S_Δ
    for each s ∈ S_Δ do
        find new γ that improves objective
        change y_s^(k-1) to y_s^(k) = γ greedily
    if objective could not be improved then
        return latest solution y^(k)
```

Experiments : Data preparation

- August 2008 version of WikiPedia used, 5.15 million entity IDs.
- Filter out IDs composed of verbs, adverbs, conjunctions etc.
- Create a trie from IDs.
- Identify spots (*NER*) by tokenizing the document and then matching spots with the trie.

Experiments : Preparing Ground Truth Collection

- Need data annotated with links to Wikipedia
- Done manually, pages obtained from popular links across various domains
- 19, 000 annotations marked, 40% marked NA, 3800 distinct entities used

Number of documents	107
Total number of spots	17,200
Spot per 100 tokens	30
Average ambiguity per Spot	5.3

Results : Only Local disambiguation

- Local approach performs well

$$\gamma_0 \leftarrow argmax_{\gamma \in \Gamma_s} w^T f_s(\gamma)$$

$$\text{if } w^T f_s(\gamma_0) > \rho_{NA} \text{ then return } \gamma_0 \text{ else return NA}$$

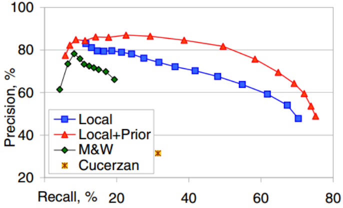


Figure 9: Even a non-collective Local approach that only uses trained node potential dominates both Cucerzan and M&W's algorithms wrt both recall and precision (IITB data). [7]

[7] from 1

LP vs Hill climbing approach

- Hill climbing and LP are equivalent

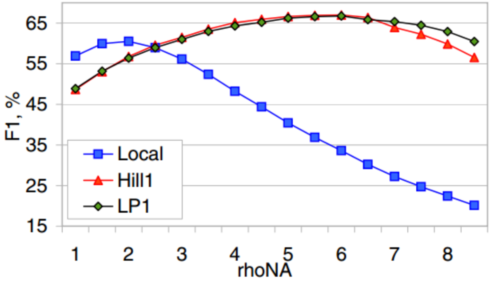


Figure 12: Hill1 attains almost the same F_1 score as LP1; both are better than Local (IITB data). [8]

[8] from 1

Recall precision for various approaches

- Exploiting topical coherence improves precision by 9
- Adding topic prior also helps

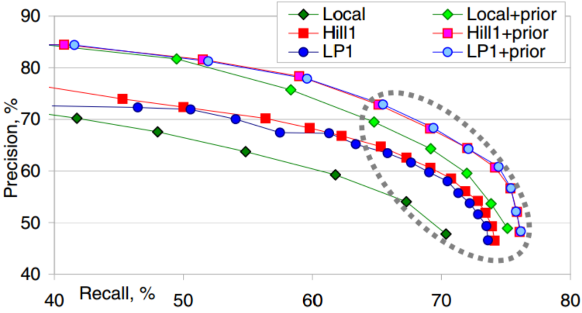


Figure 14: Recall/precision on IITB data. [9]

[9] from 1

Contributions

- *Selectively* using prior, similarity (entity - mention) and coherence (entity - entity) depending on the text
- Keyphrase based mention - entity similarity
- Modeling of the problem

Mention Entity Graph

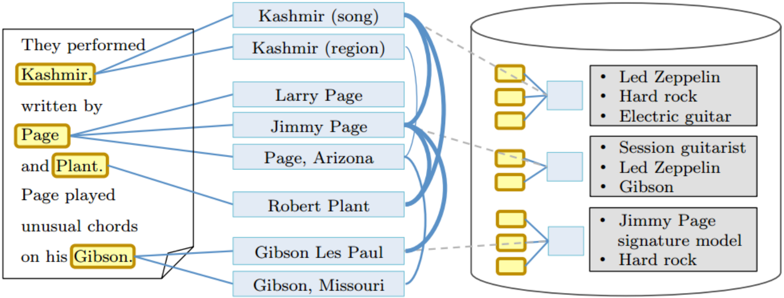


Figure: Mention Entity Graph

Recognition and Selecting Candidate Entities : Nodes of the graph

- Stanford's NER is used for finding potential named entity
- Yago provides short names and paraphrases for each entity via the "means" relation
- The list can be huge. Eg., For Afghanistan
  - Dari Persian
  - Third Anglo Afghan War
  - Republic of Afghanistan
  - Afghanistan at the Asian Games

Keyphrase based similarity : Edge weights

- Keyphrases for entities
  - Link anchor text
  - Category Names, citation titles, external references
  - Titles of articles linking to the entity
- **How important is each word?**
- $weight(w) = \frac{|w \in (KP(e)_{e' \in IN_e} KP(e'))|}{N}$ Here, IN refers to the set of entities that have in links to e
- Higher the weight, more indicative is a word of the topic. Statistics will have a higher weight for Prof. Michael Jordan.

Keyphrase based similarity

Given a mention, m, we have all the candidate entities (E) with their respective keyphrase sets. We need to find sim(m, e) for all $e \in E$

- For a given entity, for each keyphrase, find its *cover*.
- *Cover* : Shortest window of words that contains maximal number of words of the keyphrase.
- Keyphrase : "Grammy Award Winner"
- "He has been the winner of many awards during his long career including the Grammy"

Keyphrase based similarity : Scoring

$$score(q) = z * \frac{\Sigma_{w \in cover} weight(w)}{\Sigma_{w \in q} weight(w)}$$

q : Partially matching phrase $simscore(m, e) = \Sigma_{q \in KP(e)} score(q)$

For entity - entity similarity, Milne-Witten similarity measure was used.

Robustness : Selectively picking prior, similarity and concreteness

- Use prior only if prior for some candidate is above 90%
- Invoke coherence only if there is *scope* for coherence to improve something
- $diff = \Sigma_{i=1 \dots k} |prior(m, e_i) - simscore(m, e_i)|$
- If diff is not >0.9, choose the best candidate entity using only prior and simscore

NED : State of the art

Too fancy is not always good

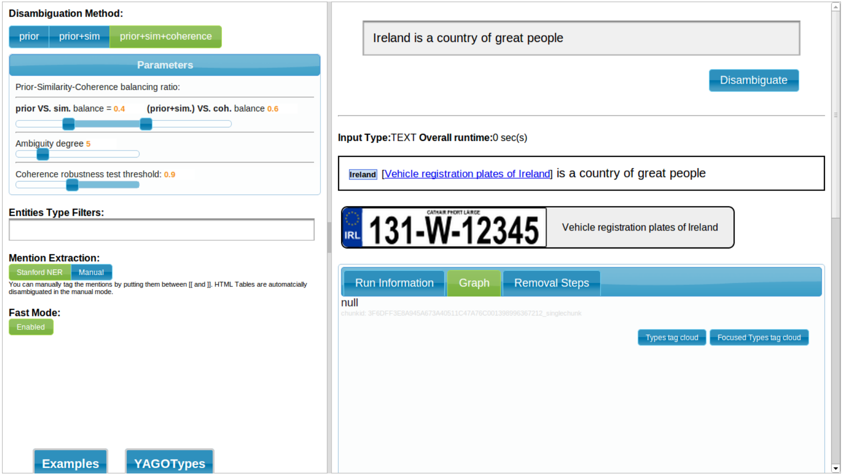


Figure: Prior + Sim + Coherence

NED : State of the art

Too fancy is not always good

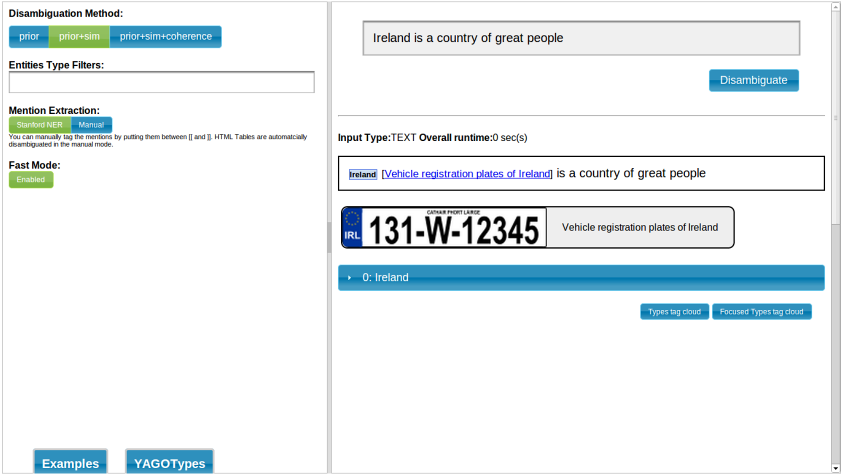


Figure: Prior + Sim

NED : State of the art

Too fancy is not always good

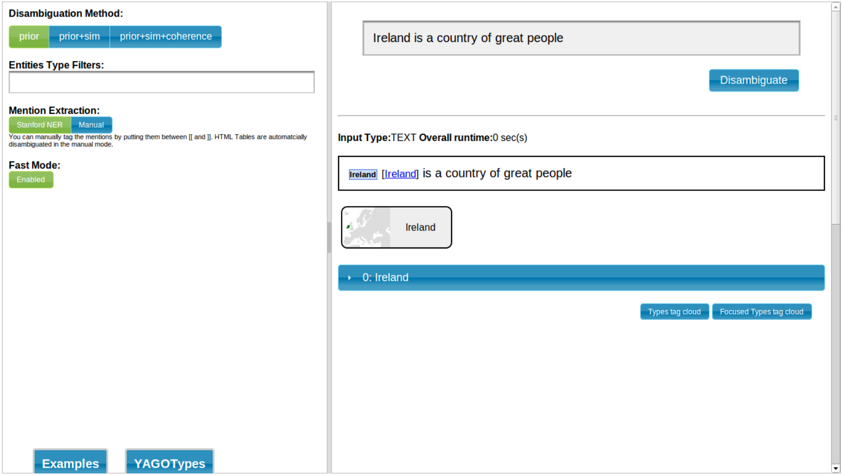
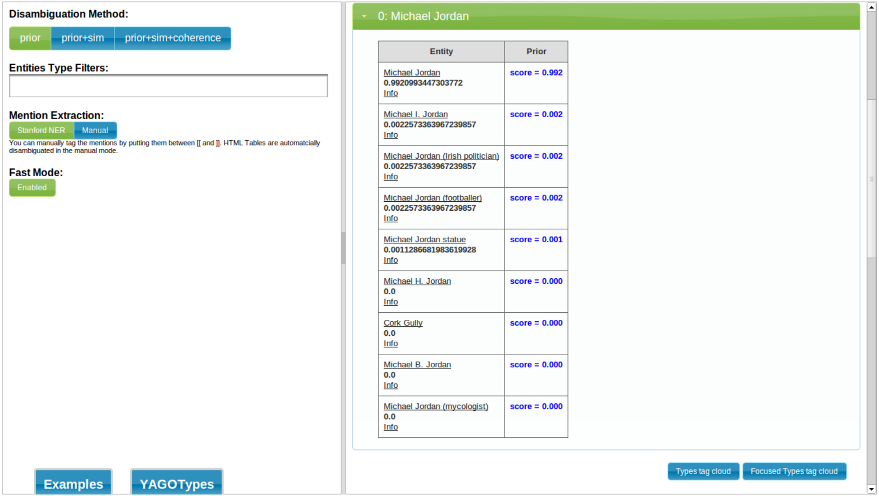


Figure: Prior

NED : State of the art

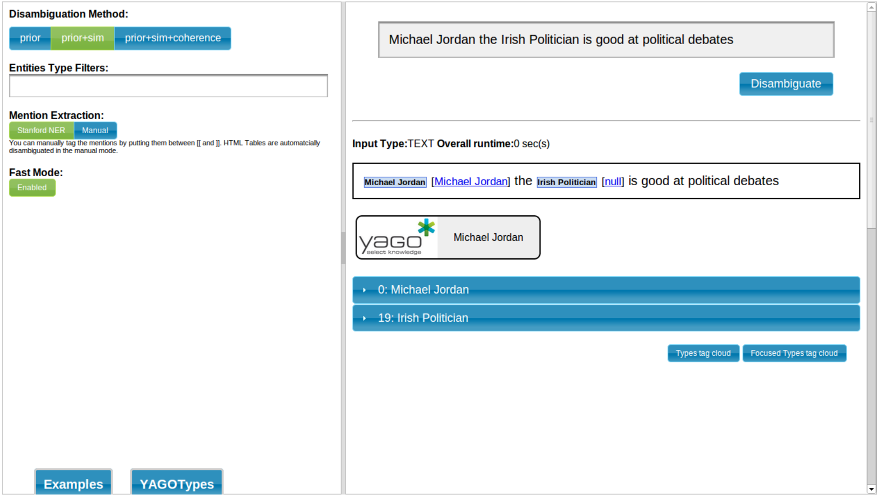
Prior is hard to beat



Entity	Prior
Michael Jordan 0.9920993447303772 Info	score = 0.992
Michael I. Jordan 0.0022573363967239857 Info	score = 0.002
Michael Jordan (Irish politician) 0.0022573363967239857 Info	score = 0.002
Michael Jordan (footballer) 0.0022573363967239857 Info	score = 0.002
Michael Jordan statue 0.0011286681983619928 Info	score = 0.001
Michael H. Jordan 0.0 Info	score = 0.000
Cork Gully 0.0 Info	score = 0.000
Michael B. Jordan 0.0 Info	score = 0.000
Michael Jordan (mycologist) 0.0 Info	score = 0.000

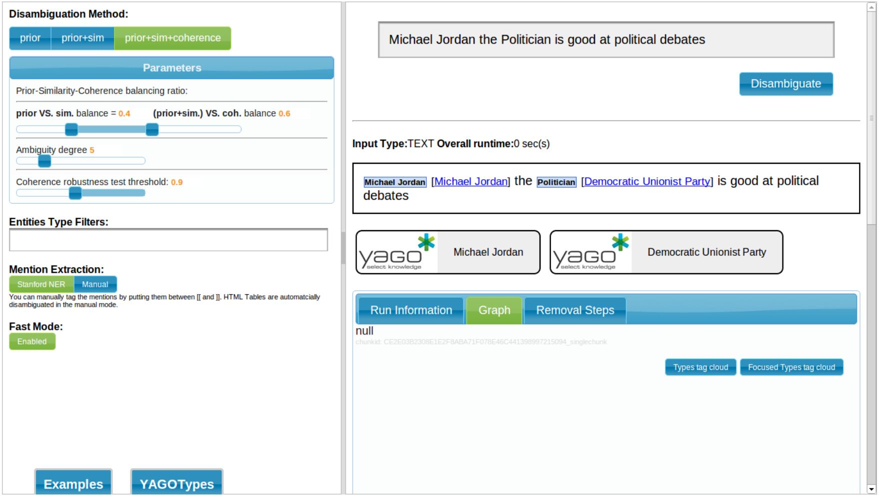
NED : State of the art

Prior is hard to beat



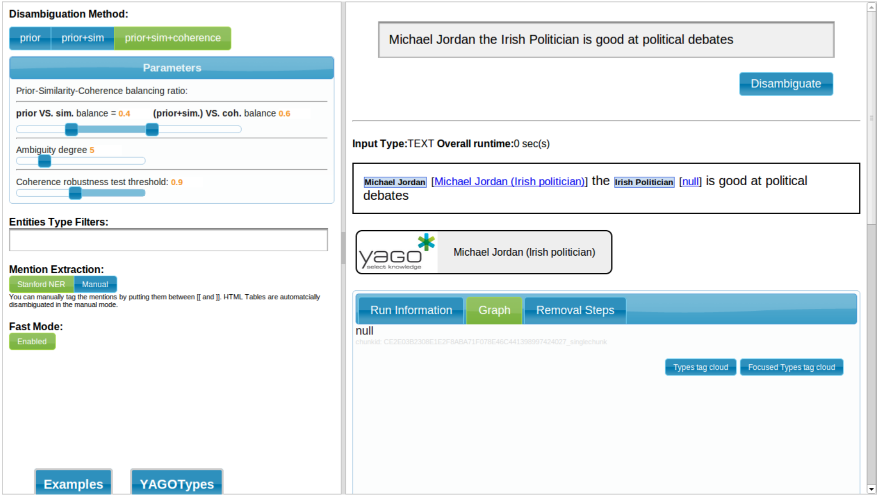
NED : State of the art

Prior is hard to beat



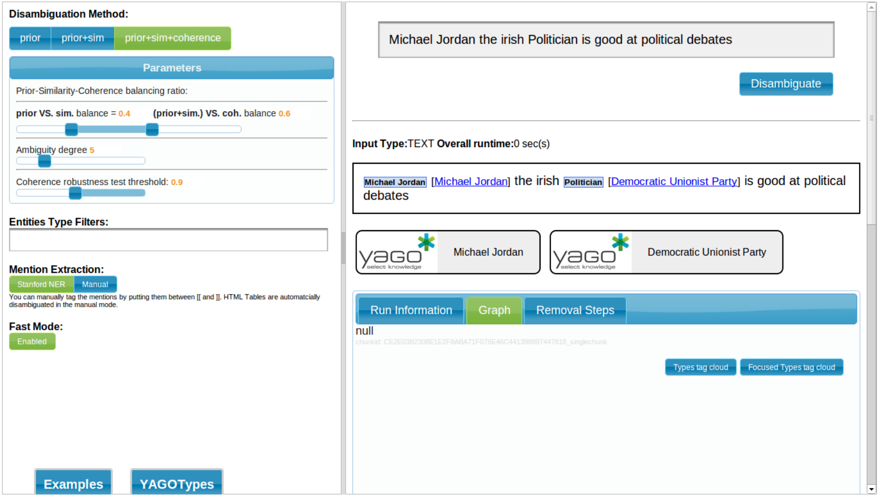
NED : State of the art

Prior is hard to beat



NED : State of the art

Prior is hard to beat



NED : State of the art

Research Focus

- Some of the recent papers like AIDA[10] report F scores brushing 90%
- Lack of a common framework to judge these tools
- Speed is a matter of concern
- Lack of training data [11]

[10] https://www.mpi-inf.mpg.de/yago-naga/aida/

[11] http://www.cs.ucsb.edu/~xyan/papers/kdd13-name_wikification.pdf

- **Quality Focused Systems** AIDA and Wikifer
- **Quantity Focused Systems** TagMe and very recently, AIDA-light
- **CSAW**

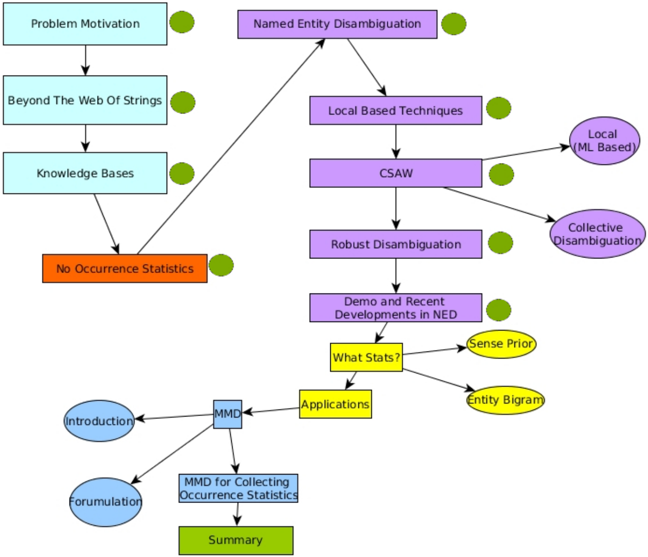


Figure: Progress

Statistics of Interest

Sense Prior

- **Sense Prior** The number of times a particular "sense" of an entity is used
- There are several "Gingerbreads" (Android 2.3, The novel)
- Sense prior would tell us how frequent is *Gingerbread the OS* compared with *Gingerbread the novel*
- $SensePrior(Si, E) = P(\text{E appears as the ith sense}) = P(Si|E)$
- Different from mention prior! (Number of times a mention links to a particular entity)

Sense Prior

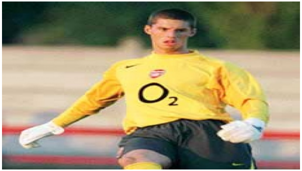
Example (Hypothetical)

Basketballer (60%)

Professor (30%)

Michael Jordan

Botanist (8%)



Footballer (2%)

Entity Bigrams

Motivation

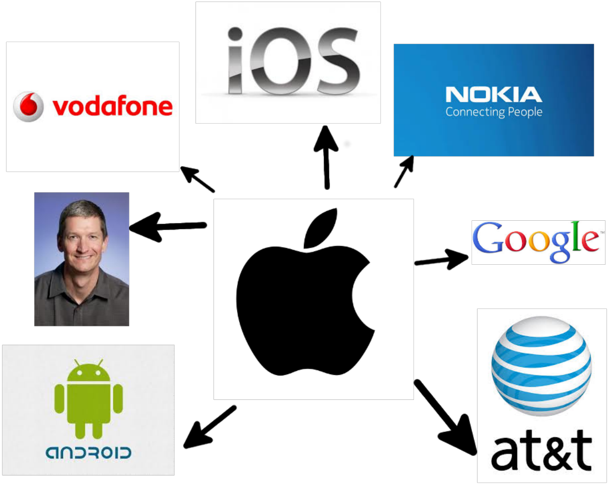


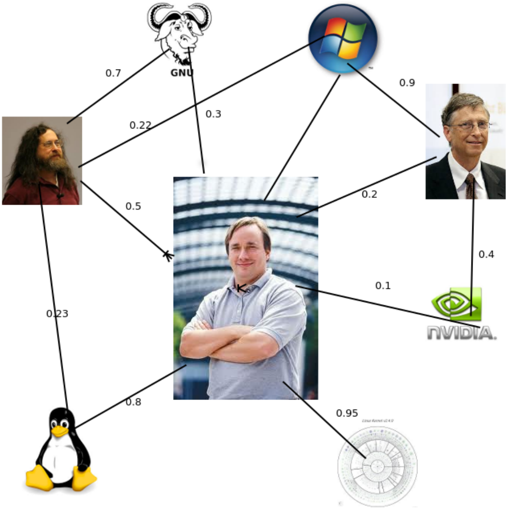
Figure: Entities Frequently Appear with related entities

Entity Bigrams

- **Entity Bigrams** Counts the number of times two given entities, taking two given senses appear together.
- Eg. : Number of times Nokia `http://en.wikipedia.org/wiki/Nokia` appears with Gingerbread `http://en.wikipedia.org/wiki/Gingerbread_(operating_system)`
- Entity Bi Gram$(E2|E1) = P(E2 \text{ } \textit{follows} \text{ } E1) = P(E2|E1)$

Entity Bigrams

Application : Finding Closely Related Entities



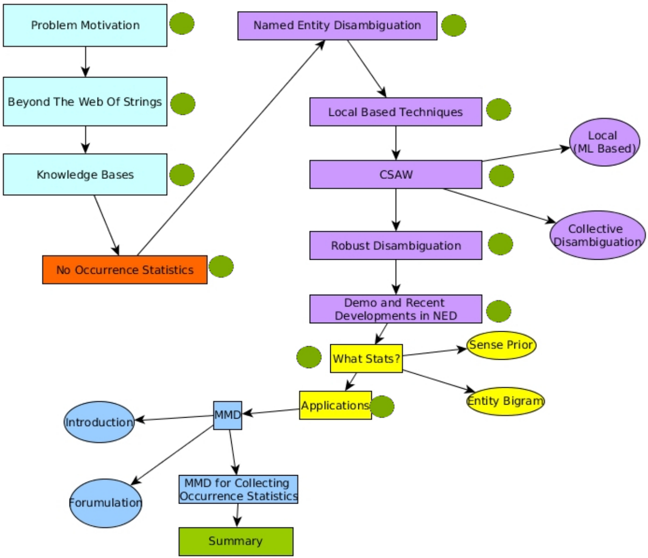
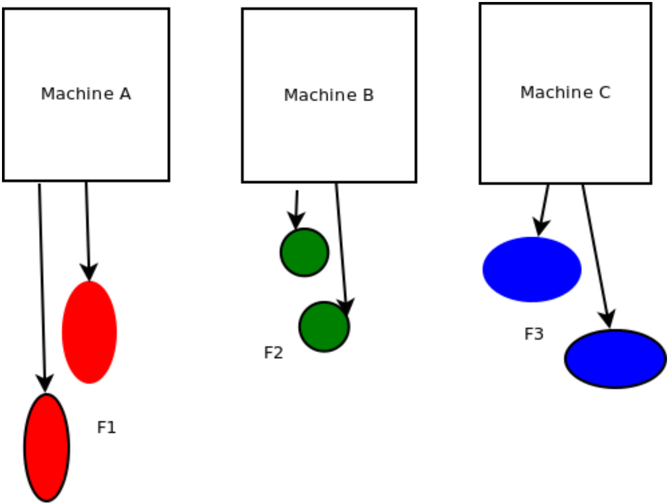


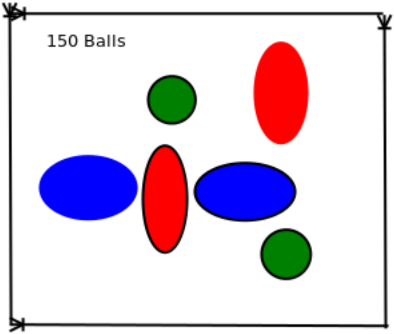
Figure: Progress

Maximum Mean Discrepancy for Collecting Entity Statistics

Introduction



Introduction



For 80 balls, the machine which manufactured is known.
Find out the machine that manufactured the ball for rest of the 20 balls

Introduction
Crux

- Let F_{test} be the average feature vector of the 30 balls
- If we knew the true fraction of balls made by each machine, $\theta_1, \theta_2, \theta_3$, we would expect

$$F_{test} = \theta_1 * F1 + \theta_2 * F2 + \theta_3 * F3 \tag{1}$$

- We don't know the θs, but the above equation tells us how to find them!
- minimize $|F_{test} - \theta_1 * F1 + \theta_2 * F2 + \theta_3 * F3|^2$ while ensuring that
  - The θs are all positive
  - The θs sum to 1

Aim

- Instead of per instance label, interested in the aggregate statistics
- Eg: Fraction of comments on a website that are positive.
- Eg: Fraction of spam mails
- Eg: Fraction of mentions that point to a particular entity.

Problem Formulation

- Let $X = x \in R_d$ be the set of all instances and $Y = 0, 1, ..., c$ be the set of all labels.
- Given a labeled dataset $D(\subset X \times Y)$, design an estimator that for any given set $U(\subset X)$ can estimate the class ratios $\theta = [\theta_0, \theta_1, ..., \theta_c]$ Where θ_y denotes the fraction of instances with class label y in U

Why not train a classifier?

- We can also get the ratio by training a classifier and running it over all of the test data
- Fails because the distribution of class labels over training and test data is usually not the same.
- Occam's Razor : One less assumption

Maximum Mean Discrepancy

- Match two distributions based on the mean of features in the hilbert space induced by a kernel K.
- Assume that distribution of features is same in both training and test data : $P_U(x|y) = P_D(x|y), \forall y \in Y$
- Thus, the test distribution must equal $Q(x) = \Sigma_y P_D(x|y)\theta_y$

Maximum Mean Discrepancy

Objective

- Let $\bar{\phi}_y and \bar{\phi}_u$ denote the true means of the feature vectors of the y th class and the unlabeled data
- Suppose we somehow get the true class ratios θ. The true mean of the feature vector of the unlabeled data can then be obtained by $\Sigma_y \theta_y \bar{\phi}_y$.
- So ideally, $\Sigma_y \theta_y \bar{\phi}_y = \bar{\phi}_u$

The objective thus is

$$min_\theta \Sigma_y \in Y \; ||\Sigma_y \theta_y \bar{\phi}_y - \bar{\phi}_u||^2$$

Such that

- $\forall y, \theta_y \geq 0$
- $\sum_{y=0}^{c} \theta_y = 1$

- But $\bar{\phi}_y$ and $\bar{\phi}_u$ are unknown and thus are approximated from the training dataset by counting.

$$\hat{\phi}_y(n_y) = \Sigma_{(x,y)\in D}\frac{\phi(x)}{n_y} \tag{2}$$

$$\hat{\phi}_U(n_u) = \Sigma_{x\in U}\frac{\phi(\bar{x})_y}{n_u} \tag{3}$$

- The objective can be written in terms of dot products of the mapped features and thus the kernel trick can be applied.

Upper bounds on the error

$$\|\widehat{\theta}(n)-\theta^*\|^2 \leq \frac{R^2\left(\frac{c^2+2c+2}{n_u}+\sum_{y=0}^{c}\frac{2}{n_y}\right)\left(1+\sqrt{\log\frac{2}{\delta}}\right)^2}{mineig(\widehat{A}(n)^\top\widehat{A}(n))} \quad (5)$$

Figure: Upper bound on the difference between the true class ratio and the predicted class ratio

The bound holds with probability atleast δ, n_y : Number of training instances, n_u : Number of test instances, R is the data spread ($max_{x \in X}||\phi(x)||$), c the number of classes

MMD for Estimating Occurrence Statistics of Entities

Required

Given a corpus with mentions identified we want reliable estimates of frequency of each of the entities.

Features

Each mention has several candidate disambiguations. This gives one way of formulating the features. For each mention, we can have a (sparse) feature vector having non zero scores for the candidates.

Training Data

Can be obtained by splicing the named entity disambiguation pipeline of any of the popular named entity disambiguators. [21] discusses how to achieve this for AIDA, a popular named entity disambiguator.

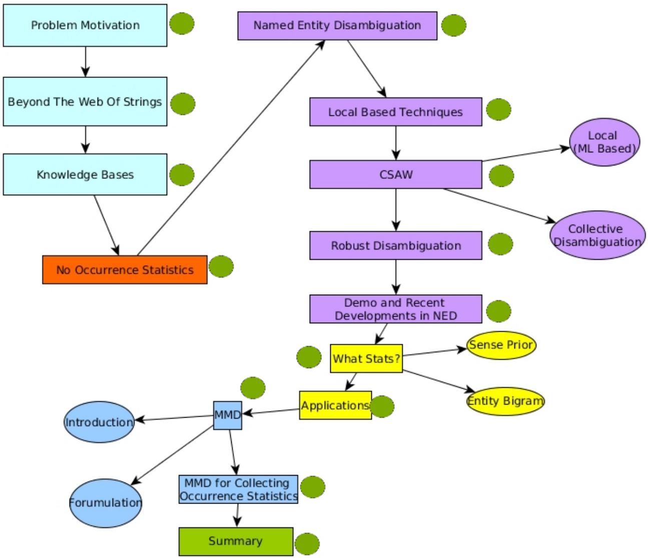


Figure: Progress

Conclusions

- The potential of open web can only be harnessed to its full extent by adding structure to it
- *Web is structured around entities*
- Many such smart applications that rely on structured web will rely on frequencies of occurrence of the entities
- Named entity disambiguators have matured over the last 8 years, with the focus now shifting towards improving speed of such systems
- It remains to be seen how approaches based on direct estimation of entity occurrence ratios perform in comparison with the standard tools, both in terms of speed and accuracy.

References I

[1] Kulkarni, Sayali, et al. "Collective annotation of Wikipedia entities in web text." Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009.

[2] http://www.cse.iitb.ac.in/~soumen/OWI/Slides/

[3] William Cohen's Survey available at 2

[4] http://en.wikipedia.org/wiki/Named-entity_recognition

[5] http://nlp.stanford.edu/software/CRF-NER.shtml

[6]

Milne, David, and Ian H. Witten. "Learning to link with wikipedia." Proceedings of the 17th ACM conference on Information and knowledge management. ACM, 2008.

[7] ws http://www.worldwidewebsize.com/

References II

[8] `http://en.wikipedia.org/wiki/Wikipedia:Statistics`

[9] Mihalcea, Rada, and Andras Csomai. “Wikify!: linking documents to encyclopedic knowledge.” Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. ACM, 2007.

[10] Hoffart, Johannes, et al. “Robust disambiguation of named entities in text.” Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011.

[11] Hoffart, Johannes, et al. ”Kore: keyphrase overlap relatedness for entity disambiguation.” Proceedings of the 21st ACM international conference on Information and knowledge management. ACM, 2012.

References III

[12] Balasubramanian, Niranjan, Stephen Soderland, and Oren Etzioni. "Rel-grams: a probabilistic model of relations in text." Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction. Association for Computational Linguistics, 2012.

[13] Balasubramanian, Niranjan, Stephen Soderland, and Oren Etzioni Mausam. "Generating Coherent Event Schemas at Scale." Proceedings of the Empirical Methods in Natural Language Processing. ACM (2013).

References IV

[14] Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In Proceedings of the 5th annual international conference on Systems documentation (SIGDOC '86), Virginia DeBuys (Ed.). ACM, New York, NY, USA, 24-26. DOI=10.1145/318723.318728 http://doi.acm.org/10.1145/318723.318728

[15] http://wordnet.princeton.edu/wordnet/

[16] Suchanek, Fabian M., Gjergji Kasneci, and Gerhard Weikum. "Yago: a core of semantic knowledge." Proceedings of the 16th international conference on World Wide Web. ACM, 2007.

[17] Auer, Sren, et al. "Dbpedia: A nucleus for a web of open data." The semantic web. Springer Berlin Heidelberg, 2007. 722-735.

References V

[18] Nakashole, Ndapandula, Gerhard Weikum, and Fabian Suchanek. "PATTY: a taxonomy of relational patterns with semantic types." Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, 2012.

[19] Bollacker, Kurt, et al. "Freebase: a collaboratively created graph database for structuring human knowledge." Proceedings of the 2008 ACM SIGMOD international conference on Management of data. ACM, 2008.

[20] Iyer, Arun, Saketha Nath, and Sunita Sarawagi. "Maximum Mean Discrepancy for Class Ratio Estimation: Convergence Bounds and Kernel Selection." Proceedings of The 31st International Conference on Machine Learning. 2014.

References VI

[21] Using Structured learning for named entity disambiguation, www.cse.iitb.ac.in/~amanmadaan/structlearn.pdf