Semantic Search (Part IV)

[DAT640] Information Retrieval and Text Mining

Krisztian Balog
University of Stavanger

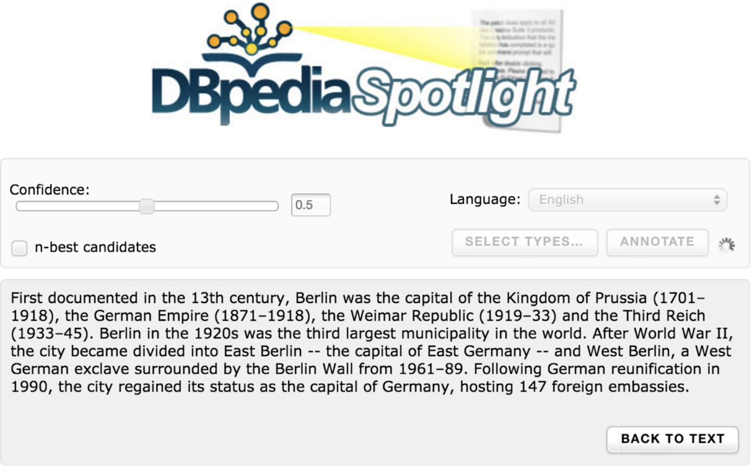
October 29, 2019

Entity linking

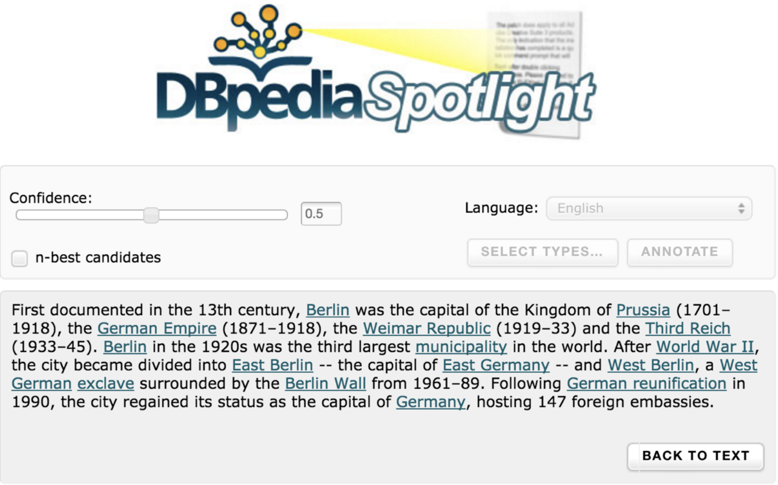
Entity linking

- Task: recognizing entity mentions in text and linking them to the corresponding entries in a knowledge base (KB)
    - Limited to recognizing entities for which a target entry exists in the reference KB; each KB entry is a candidate
    - It is assumed that the document provides sufficient context for disambiguating entities

Entity linking in action



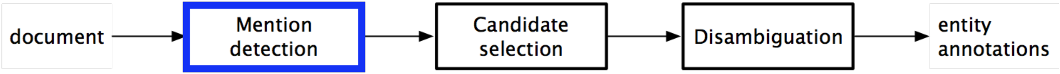
Entity linking in action



Anatomy of an entity linking system

- **Mention detection**: Identification of text snippets that can potentially be linked to entities
- **Candidate selection**: Generating a set of candidate entities for each mention
- **Disambiguation**: Selecting a single entity (or none) for each mention, based on the context

Mention detection



Mention detection

- Goal: Detect all "linkable" phrases
- Challenges
  - Recall oriented
    - Do not miss any entity that should be linked
  - Find entity name variants
    - E.g. "jlo" is name variant of Jennifer Lopez
  - Filter out inappropriate ones
    - E.g. "new york" matches >2k different entities

Common approach

1. Build a dictionary of entity surface forms
   - Entities with all names variants
2. Check all document n-grams against the dictionary
   - The value of n is set typically between 6 and 8
3. Filter out undesired entities
   - Can be done here or later in the pipeline

Example

Surface form dictionary

Surface form s	Entities $\mathcal{E}_s$
Empire	British_Empire Empire_(magazine) First_French_Empire Galactic_Empire_(Star_Wars) Holy_Roman_Empire Roman_Empire …
Empire State	Empire_State_(band) Empire_State_Building Empire_State_Film_Festival
Empire State Building	Empire_State_Building
…	…
Times Square	Times_Square Times_Square_(Hong_Kong) Times_Square_(IRT_42nd_Street_Shuttle) …
…	…

Home to the **Empire State Building**, **Times Square**, **Statue of Liberty** and other iconic sites, **New York City** is a fast-paced, globally influential center of art, culture, fashion and finance.

Surface form dictionary construction from Wikipedia

- **Page title**
  - Canonical (most common) name of the entity

Surface form dictionary construction from Wikipedia

- Page title
- **Redirect pages**
    - Alternative names that are frequently used to refer to an entity

Surface form dictionary construction from Wikipedia

- Page title
- Redirect pages
- **Disambiguation pages**
    - List of entities that share the same name

Surface form dictionary construction from Wikipedia

- Page title
- Redirect pages
- Disambiguation pages
- **Anchor texts**
  - of links pointing to the entity's Wikipedia page

Surface form dictionary construction from Wikipedia

- Page title
- Redirect pages
- Disambiguation pages
- Anchor texts
- **Bold texts from first paragraph**
  - generally denote other name variants of the entity

Surface form dictionary construction from other sources

- Anchor texts from external web pages pointing to Wikipedia articles
- Problem of *synonym discovery*
  - Expanding acronyms
  - Leveraging search results or query-click logs from a web search engine
  - ...

Filtering mentions

- Objective is to filter our mentions that are unlikely to be linked to any entity
- **Keyphraseness**

$$P(\text{keyphrase}|m) = \frac{|D_{link}(m)|}{|D(m)|}$$

  - $|D_{link}(m)|$ is the number of Wikipedia articles where m appears as an anchor text of a link
  - $|D(m)|$ is the number of Wikipedia articles that contain m

Filtering mentions (cont'd)

- **Link probability**

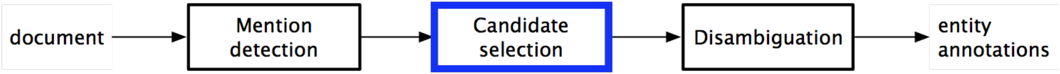
$$P(\text{link}|m) = \frac{link(m)}{freq(m)}$$

  - $link(m)$ is the number of times mention m appears as an anchor text of a link
  - $freq(m)$ is the total number of times mention m occurs in Wikipedia (as a link or not)

Overlapping entity mentions

- Dealing with them in this phase
  - E.g., by dropping a mention if it is subsumed by another mention
- Keeping them and postponing the decision to a later stage (candidate selection or disambiguation)

Candidate selection



Candidate selection

- Goal: Narrow down the space of disambiguation possibilities
- Balances between precision and recall (effectiveness vs. efficiency)
- Often approached as a ranking problem
  - Keeping only candidates above a score/rank threshold for downstream processing

Commonness

- Perform the ranking of candidate entities based on their overall popularity, i.e., "most common sense"

$$P(e|m) = \frac{n(m,e)}{\sum_{e' \in \mathcal{E}} n(m,e')}$$

  - $n(m,e)$ the number of times entity e is the link destination of mention m
- Can be pre-computed and stored in the entity surface form dictionary
- Follows a power law with a long tail of extremely unlikely senses; entities at the tail end of the distribution can be safely discarded
  - E.g., 0.001 is a sensible threshold

Example

Entity e	**Commonness** $P(e\|m)$
Times_Square	0.940
Times_Square_(film)	0.017
Times_Square_(Hong_Kong)	0.011
Times_Square_(IRT_42nd_Street_Shuttle)	0.006
…	…

Home to the **Empire State Building**, **Times Square**, **Statue of Liberty** and other iconic sites, **New York City** is a fast-paced, globally influential center of art, culture, fashion and finance.

Example #2

Entity	Commonness
FIFA_World_Cup	0.2358
FIS_Apline_Ski_World_Cup	0.0682
2009_FINA_Swimming_World_Cup	0.0633
World_Cup_(men's_golf)	0.0622
...	

Bulgaria's best **World Cup** performance was in the **1994 World Cup** where they beat **Germany**, to reach the semi-finals, losing to Italy, and finishing in fourth ...

Entity	Commonness
1998_FIFA_World_Cup	0.9556
1998_IAAF_World_Cup	0.0296
1998_Alpine_Skiing_World_Cup	0.0059
...	

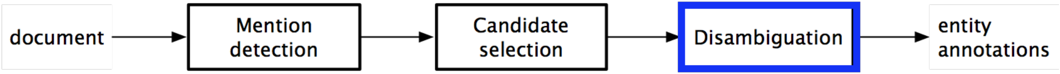
Entity	Commonness
Germany	0.9417
Germany_national_football_team	0.0139
Nazi_Germany	0.0081
German_Empire	0.0065
...	

- Commonness works in many of the cases, but not in all
- Other entities help to disambiguate which entity is being referred to

Exercise #1

- Entity linking based on commonness (paper-based)

Disambiguation



Disambiguation

- Baseline approach: most common sense
- Consider additional types of evidence
  - **Prior importance** of entities and mentions
  - **Contextual similarity** between the text surrounding the mention and the candidate entity
  - **Coherence** among all entity linking decisions in the document
- Combine these signals
  - Using supervised learning or graph-based approaches
- Optionally perform pruning
  - Reject low confidence or semantically meaningless annotations

Prior importance features

- **Context-independent features**
  - Neither the text nor other mentions in the document are taken into account
- Keyphraseness
- Link probability
- Commonness

Prior importance features (cont'd)

- **Link prior**
  - Popularity of the entity measured in terms of incoming links

$$P_{link}(e) = \frac{|\mathcal{L}_e|}{\sum_{e' \in \mathcal{E}} |\mathcal{L}_{e'}|}$$

  - $|\mathcal{L}_e|$ is the total number of incoming links entity e has
- **Page views**
  - Popularity of the entity measured in terms traffic volume

$$P_{pageviews}(e) = \frac{pageviews(e)}{\sum_{e' \in \mathcal{E}} pageviews(e')}$$

  - $pageviews(e)$ is the total number of page views (measured over a certain time period)

Contextual features

- Compare the surrounding *context* of a mention with the (textual) representation of the given candidate entity
- Context of a mention
  - Window of text (sentence, paragraph) around the mention
  - Entire document
- Entity's representation
  - Wikipedia entity page, first description paragraph, terms with highest TF-IDF score, etc.
  - Entity's description in the knowledge base

Contextual similarity

- Commonly: bag-of-words representation
- **Cosine similarity**

$$sim_{cos}(m,e) = \frac{\vec{d}_m \cdot \vec{d}_e}{||\vec{d}_m||\,||\vec{d}_e||}$$

- Many other options for measuring similarity
  - Dot product, KL divergence, Jaccard similarity
- Representation does not have to be limited to bag-of-words
  - Concept vectors (named entities, Wikipedia categories, anchor text, keyphrases, etc.)

Entity-relatedness features

- It can reasonably be assumed that a document focuses on one or at most a few topics
- Therefore, entities mentioned in a document should be topically related to each other
- Capturing *topical coherence* by developing some measure of *relatedness* between (linked) entities
  - Defined for pairs of entities

Wikipedia Link-based Measure (WSM)

- Often referred to simply as *relatedness*
- A close relationship is assumed between two entities if there is a large overlap between the entities linking to them

$$WLM(e, e') = 1 - \frac{\log(\max(|\mathcal{L}_e|, |\mathcal{L}_{e'}|)) - \log(|\mathcal{L}_e \cap \mathcal{L}_{e'}|)}{\log(|\mathcal{E}|) - \log(\min(|\mathcal{L}_e|, |\mathcal{L}_{e'}|))}$$

  - $\mathcal{L}_e$ is the set of entities that link to e
  - $|\mathcal{E}|$ is the total number of entities

Wikipedia Link-based Measure (WSM)

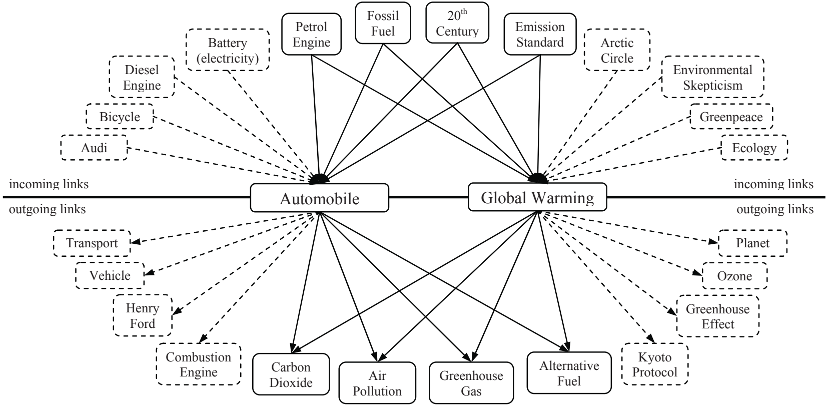


Figure: Image taken from Milne and Witten (2008). An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links. In: AAAI WikiAI Workshop.

Entity-relatedness features

- Numerous ways to define relatedness
  - Consider not only incoming, but also outgoing links or the union of incoming and outgoing links
  - Jaccard similarity, Pointwise Mutual Information (PMI), or the Chi-square statistic, etc.
- A relatedness function does not have to be symmetric
  - E.g., the relatedness of the United States given Neil Armstrong is intuitively larger than the relatedness of Neil Armstrong given the United States
  - **Conditional probability**

$$P(e'|e) = \frac{|\mathcal{L}_{e'} \cap \mathcal{L}_e|}{|\mathcal{L}_e|}$$

- Having a single relatedness function is preferred, to keep the disambiguation process simple
- Various relatedness measures can effectively be combined into a single score using a machine learning approach

Disambiguation approaches

- Consider *local compatibility* (including prior evidence) and *coherence* with the other entity linking decisions
- Overall objective function:

$$\Gamma^* = \mathrm{arg\,max}_\Gamma \Big(\sum_{(m,e)\in\Gamma} \phi(m,e) + \psi(\Gamma) \Big)$$

  - $\phi(m,e)$ is the local compatibility between the mention and the assigned entity
  - $\psi(\Gamma)$ is the coherence function for all entity annotations in the document
  - Γ is a solution (set of mention-entity pairs)
- **This optimization problem is NP-hard!**
  - Need to resort to approximation algorithms and heuristics

Disambiguation strategies

- **Individually**, one-mention-at-a-time
  - Rank candidates for each mention, take the top ranked one (or NIL)
  - Interdependence between entity linking decisions may be incorporated in a pairwise fashion

$$\Gamma(m) = \operatorname{arg\,max}_{e \in \mathcal{E}_m} score(e, m)$$

- **Collectively**, all mentions in the document jointly

Disambiguation approaches

Approach	Context	Entity interdependence
Most common sense	none	none
Individual local disambiguation	text	none
Individual global disambiguation	text & entities	pairwise
Collective disambiguation	text & entities	collective

Individual local disambiguation

- Early entity linking approaches
- Local compatibility score can be written as a linear combination of features

$$\phi(e, m) = \sum_i \lambda_i f_i(e, m)$$

  - $f_i(e, m)$ can be either a context-independent or a context-dependent feature
- Learn the "optimal" combination of features from training data using machine learning

Individual global disambiguation

- Consider what other entities are mentioned in the document
- True global optimization would be NP-hard
- Good approximation can be computed efficiently by considering pairwise interdependencies for each mention independently
    - Pairwise entity relatedness scores need to be aggregated into a single number (how coherent the given candidate entity is with the rest of the entities in the document)

TAGME (Ferragina & Scaiella, 2010)

- Combine the two most important features (*commonness* and *relatedness*) using a voting scheme
- The score of a candidate entity for a particular mention:

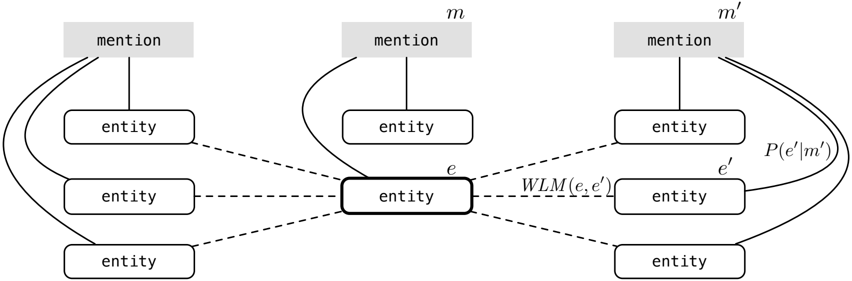
$$score(e, m) = \sum_{\substack{m' \in \mathcal{M}_d \\ m' \neq m}} vote(m', e)$$

- The vote function estimates the agreement between e and all candidate entities of all other mentions in the document

TAGME (voting mechanism)

- Average relatedness between each possible disambiguation, weighted by its commonness score

$$vote(m', e) = \frac{\sum_{e' \in \mathcal{E}_{m'}} WLM(e, e')P(e'|m')}{|\mathcal{E}_{m'}|}$$



TAGME (final score)

- Final decision uses a simple but robust heuristic
  - The top entities with the highest score are considered for a given mention and the one with the highest commonness score is selected

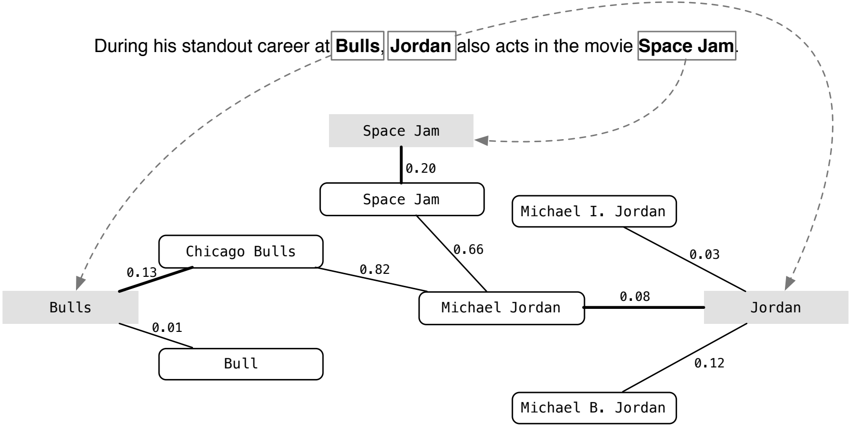
$$\Gamma(m) = \arg\max_{e \in \mathcal{E}_m} \{P(e|m) : e \in \mathsf{top}_\epsilon[score(e, m)]\}$$

- Note that $score$ merely acts as a filter
  - Only entities in the top ϵ percent of the scores are retained ($\epsilon = 0.3$)
  - Out of the remaining entities, the most common sense of the mention will be finally selected

Collective disambiguation

- Graph-based representation
- **Mention-entity edges** capture the local compatibility between the mention and the entity
  - Measured using a combination of context-independent and context-dependent features
- **Entity-entity edges** represent the semantic relatedness between a pair of entities
  - Common choice is *relatedness* (WLM)
- Use these relations jointly to identify a single referent entity (or none) for each of the mentions

Example



During his standout career at Bulls, Jordan also acts in the movie Space Jam.
Space Jam
0.20
Space Jam
Michael I. Jordan
0.66
0.03
Chicago Bulls
0.13
0.82
Bulls
Michael Jordan
0.08
Jordan
0.01
Bull
0.12
Michael B. Jordan

AIDA (Hoffart et al., 2011)

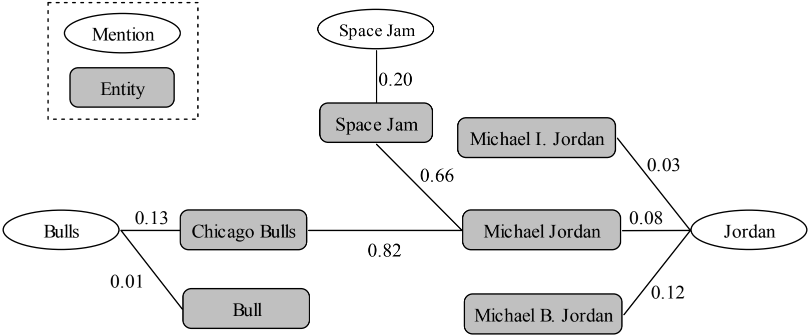
- Problem formulation: find a dense subgraph that contains all mention nodes and exactly one mention-entity edge for each mention
- Greedy algorithm iteratively removes edges

AIDA algorithm

- Start with the full graph
- Iteratively remove the entity node with the lowest *weighted degree* (along with all its incident edges), provided that each mention node remains connected to at least one entity
  - Weighted degree of an entity node is the sum of the weights of its incident edges
- The graph with the highest *density* is kept as the solution
  - The density of the graph is measured as the minimum weighted degree among its entity nodes

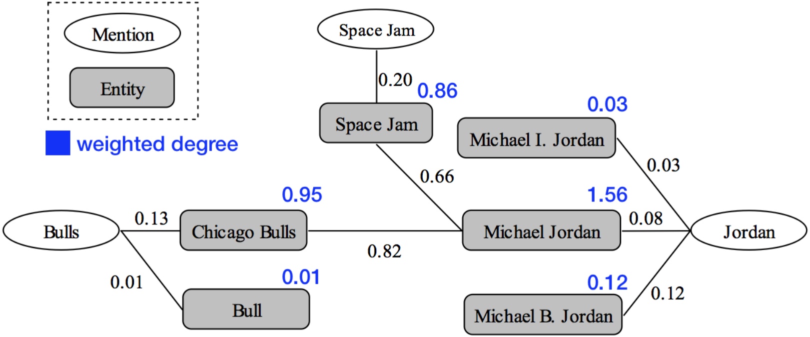
Example iteration #1

- Which entity should be removed first?



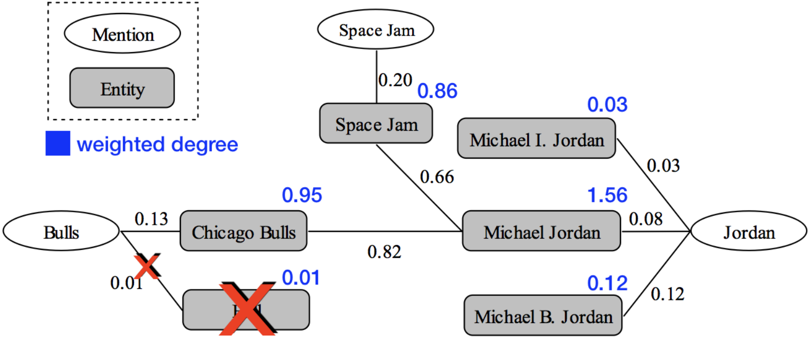
Example iteration #1

- Which entity should be removed first?



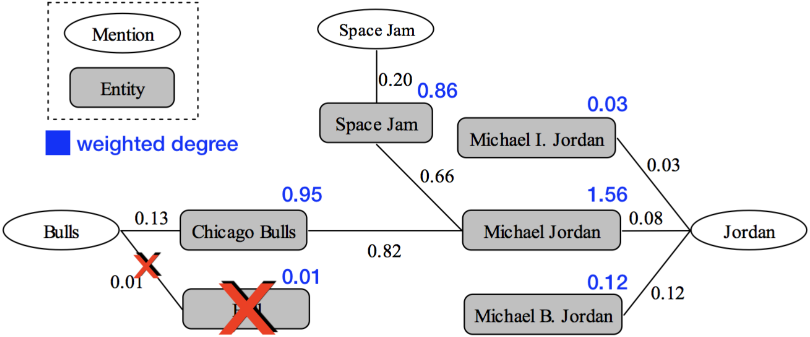
Example iteration #1

- Which entity should be removed first?



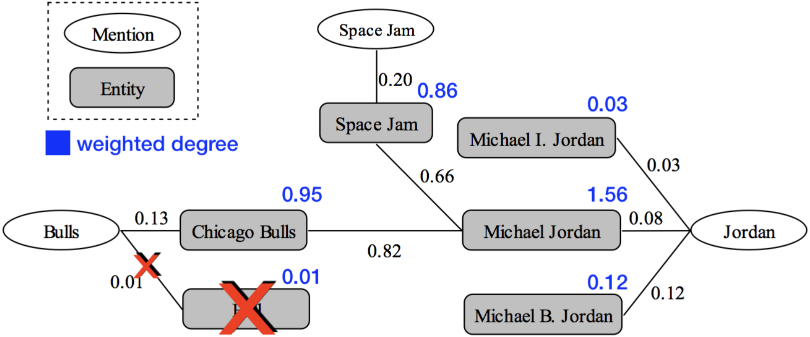
Example iteration #1

- What is the density of the graph?



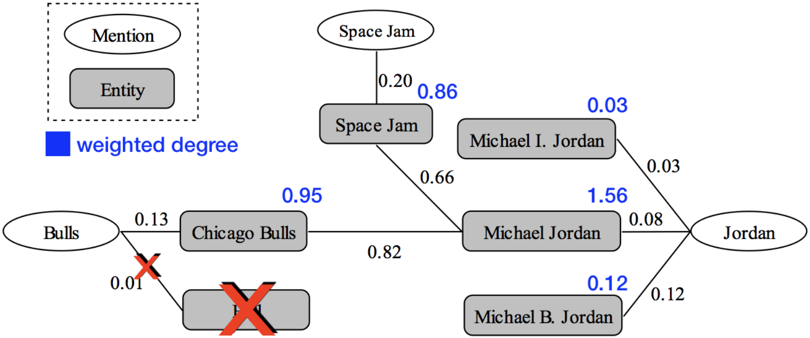
Example iteration #1

- What is the density of the graph? **0.03**



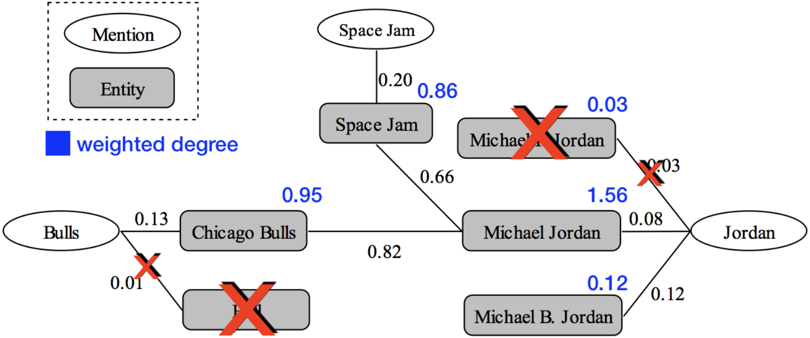
Example iteration #2

- Which entity should be removed next?



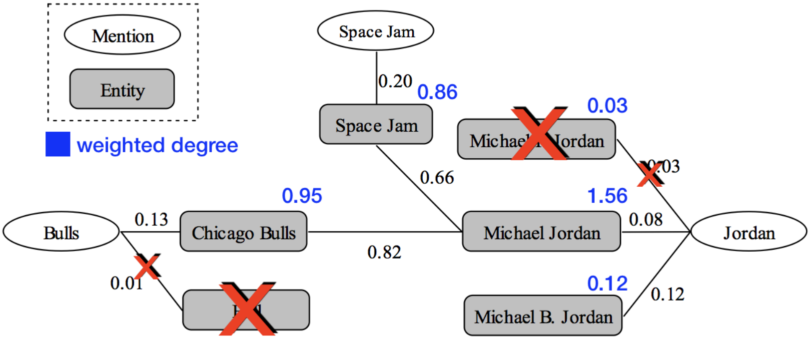
Example iteration #2

- Which entity should be removed next?



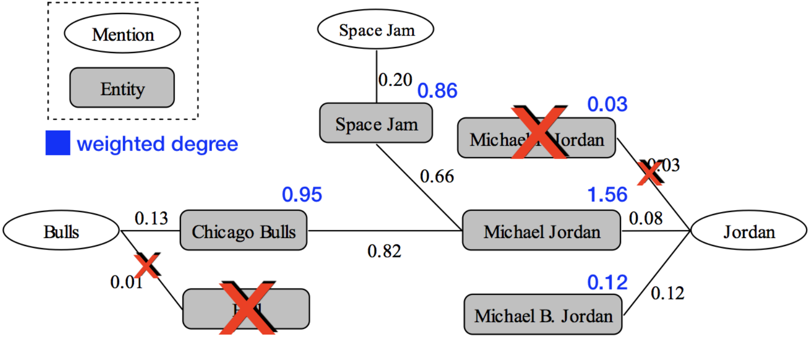
Example iteration #2

- What is the density of the graph?



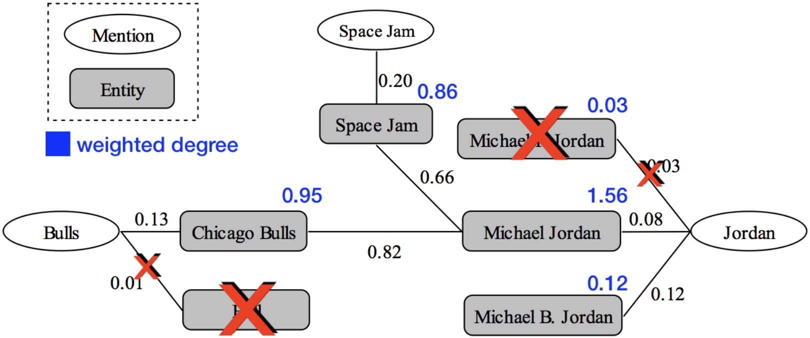
Example iteration #2

- What is the density of the graph? **0.12**



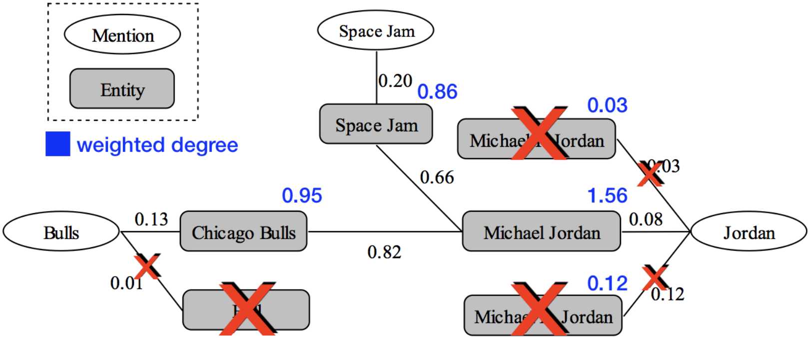
Example iteration #3

- Which entity should be removed next?



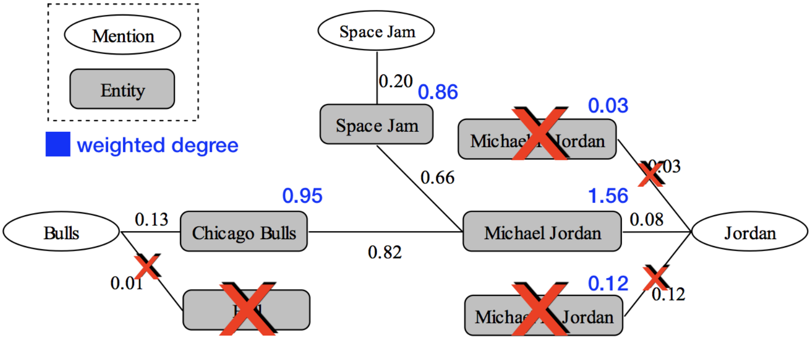
Example iteration #3

- Which entity should be removed next?



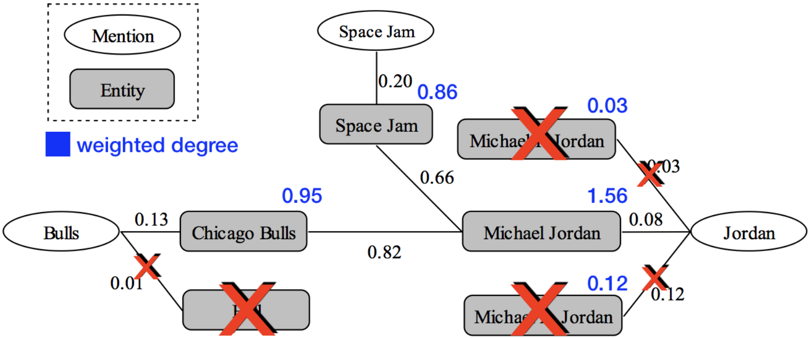
Example iteration #3

- What is the density of the graph?



Example iteration #3

- What is the density of the graph? **0.86**



AIDA pre- and post-processing

- Pre-processing phase: remove entities that are "too distant" from the mention nodes
- At the end of the iterations, the solution graph may still contain mentions that are connected to more than one entity; deal with this in post-processing
  - If the graph is sufficiently small, it is feasible to exhaustively consider all possible mention-entity pairs
  - Otherwise, a faster local (hill-climbing) search algorithm may be used

Pruning

- Discarding meaningless or low-confidence annotations produced by the disambiguation phase
- Simplest solution: use a confidence threshold
- More advanced solutions
  - Machine learned classifier to retain only entities that are "relevant enough" (human editor would annotate them)
  - Optimization problem: decide, for each mention, whether switching the top ranked disambiguation to NIL would improve the objective function

Evaluation

Evaluation (end-to-end)

- Comparing the system-generated annotations against a human-annotated gold standard
- Evaluation criteria
  - **Perfect match**: both the linked entity and the mention offsets must match
  - **Relaxed match**: the linked entity must match, it is sufficient if the mention overlaps with the gold standard

Evaluation with relaxed match

Example #1

Example #2

Evaluation metrics

- Set-based metrics
  - **Precision**: fraction of correctly linked entities that have been annotated by the system
  - **Recall**: fraction of correctly linked entities that should be annotated
  - **F-measure**: harmonic mean of precision and recall
- Metrics are computed over a collection of documents
  - Micro-averaged: aggregated across mentions
  - Macro-averaged: aggregated across documents

Evaluation metrics

- **Micro-averaged**

$$P_{mic} = \frac{|\mathcal{A}_{\mathcal{D}} \cap \hat{\mathcal{A}}_{\mathcal{D}}|}{|\mathcal{A}_{\mathcal{D}}|} \qquad R_{mic} = \frac{|\mathcal{A}_{\mathcal{D}} \cap \hat{\mathcal{A}}_{\mathcal{D}}|}{|\hat{\mathcal{A}}_{\mathcal{D}}|}$$

  - $\mathcal{A}_{\mathcal{D}}$ include all annotations for a set $\mathcal{D}$ of documents
  - $\hat{\mathcal{A}}_{\mathcal{D}}$ is the collection of reference annotations for $\mathcal{D}$

- **Macro-averaged**

$$P_{mac} = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \frac{|\mathcal{A}_d \cap \hat{\mathcal{A}}_d|}{|\mathcal{A}_d|} \qquad R_{mac} = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \frac{|\mathcal{A}_d \cap \hat{\mathcal{A}}_d|}{|\hat{\mathcal{A}}_d|}$$

  - $\mathcal{A}_d$ are the annotations generated by the entity linking system
  - $\hat{\mathcal{A}}_d$ denote the reference (ground truth) annotations for a single document d

- **F1 score**

$$F1 = \frac{2\,P\,R}{P + R}$$

Component-based evaluation

- The pipeline architecture makes the evaluation of entity linking systems especially challenging
  - The main focus is on the disambiguation component, but its performance is largely influenced by the preceding steps
- Fair comparison between two approaches can only be made if they share all other elements of the pipeline

Exercise #2

- Entity linking evaluation (paper-based)

Reading

- Entity-Oriented Search (Balog)[1]
  - Chapter 5

[1] PDF: https://rd.springer.com/content/pdf/10.1007%2F978-3-319-93935-3.pdf