Candidate Entity Ranking

Candidate Entity Ranking

- In some standard benchmark datasets (TAC-KBP2010,2011), entity mentions were linked on average to about 13 different entities in the Knowledge Base

How can we rank them?

- Two types of methods:
  - *Supervised Ranking Methods*
  - *Unsupervised Ranking Methods*

Candidate Entity Ranking

- In some standard benchmark datasets (TAC-KBP2010,2011), entity mentions were linked on average to about 13 different entities in the Knowledge Base

How can we rank them?

- Another different classification:
  - *Independent Ranking Methods:* Assume mentions are independent
  - *Collective Ranking Methods:* Assume a document refers to coherent entities
  - *Collaborative Ranking Methods:* Leverage cross-document context to disambiguate

Features

What can we leverage in order to identify the correct mapping <mention, entity> (“Jordan”, en.wikipedia.org/wiki/Michael_Jordan)?

- Two types of features:
  - **Context-independent features**, which just rely on the surface-form of the entity mention
  - **Context-dependent features,** which also look at the context around the entity mention

Features

Context-independent features

- *Name string comparison:* We can simply check whether the mention and the entity label in the KB match
  - Exact matching
  - Dice coefficient score
  - Hamming distance
  - Combinations of the above

Features

Context-independent features

- *Entity popularity:* Given a certain mention, pick the entity which is the most *popular*
- One way to calculate popularity consists of using Wikipedia. Let E_m be the set of potential entities, e_i is a candidate entity and e_m the entity mention

$$Pop(e_i) = \frac{count_m(e_i)}{\sum_{e_j \in E_m} count_m(e_j)},$$

- $count_m$ is the count of links to e_i where the anchor text is e_m

Features

Context-independent features

- *Entity type:* NER can return a broad type for a given mention (ie. Person, Organization, Location, etc)
- We can check whether there is a match between the entity type in the KB and the one recognized by the NER

Features

Context-dependent features

- The context around the entity offers valuable information
- *Bag of words (BOW):*
  - All words in the document that contains the entity mention (or in a suitable window) and match with the words associated to the entity (e.g. the Wikipedia page)

Features

Context-dependent features

- The context around the entity offers valuable information
- *Concept vectors:*
  - From the document that contains the mention, we can extract **key-phrases, anchor texts, named entities,** etc.
  - We can use these features to create a vector that represents both entities and mentions
  - The similarity between the entity and mention can be calculated using
    - *cosine similarity*

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$

Jaccard Similarity

$$J(\mathbf{x}, \mathbf{y}) = \frac{\sum_i \min(x_i, y_i)}{\sum_i \max(x_i, y_i)}$$

VU VRIJE UNIVERSITEIT AMSTERDAM

Features

Context-dependent features

- The context around the entity offers valuable information
- *Coherence between mappings:*
  - Main principle: entities in one document are coherent with one or few topics
  - We can measure how related are two candidate entities for two mentions. In Wikipedia, we can do it by counting how many articles link to both entities
    - E.g. Normalized Google Distance

$$Coh_G(u_1, u_2) = 1 - \frac{log(max(|U_1|, |U_2|)) - log(|U_1 \cap U_2|)}{log(|WP|) - log(min(|U_1|, |U_2|))}$$

Features

Context-dependent features

- The context around the entity offers valuable information
- *Coherence between mappings:*
  - Main principle: entities in one document are coherent with one or few topics
  - We can measure how related are two candidate entities for two mentions. In Wikipedia, we can do it by counting how many articles link to both entities
    - E.g. Jaccard Similarity

$$Coh_J(u_1, u_2) = \frac{|U_1 \cap U_2|}{|U_1 \cup U_2|}$$

Features

Context-dependent features

- The context around the entity offers valuable information

In general, context-dependent features are expensive to calculate because we must consider many different possibilities.

There is no perfect set of features!

Supervised Ranking Methods

Candidate Entity Ranking

- In some standard benchmark datasets (TAC-KBP2010,2011), entity mentions were linked on average to about 13 different entities in the Knowledge Base

How can we pick the right one among them?

- Two types of methods:
  - ***Supervised Ranking Methods***
  - *Unsupervised Ranking Methods*

Binary Classification Methods

- Given in input <*mention, entity*>, we can train a classifier that returns 1 if the mapping is appropriate or 0 if it is not appropriate
- We can use off-the-shelf classifiers, e.g. SVM, Naïve Bayes Classifiers, etc. EXAMPLE SVM
- To describe *mention* and *entity,* we can use the features described before

Probabilistic Methods

- Instead of using classifiers, we can use a probabilistic models to express the likelihood that a given mention and surrounding context is connected to a entity
  - The work [1] propose a generative *entity-mention* model. Accuracy is 86% on TAC_KBP 2009
  - Another option consists of using *crowds.* Whenever the prediction is deemed too uncertain, [2] proposes to exploit crowdsourcing services and let humans disambiguate it

[1] X. Han and L. Sun, “A generative entity-mention model for linking entities with knowledge base,” in *ACL*, 2011, pp. 945–954.
[2] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux, “ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking,” in *WWW*, 2012, pp. 469–478.

Unsupervised Ranking Methods

Candidate Entity Ranking

- In some standard benchmark datasets (TAC-KBP2010,2011), entity mentions were linked on average to about 13 different entities in the Knowledge Base

How can we rank them?

- Two types of methods:
  - *Supervised Ranking Methods*
  - ***Unsupervised Ranking Methods***

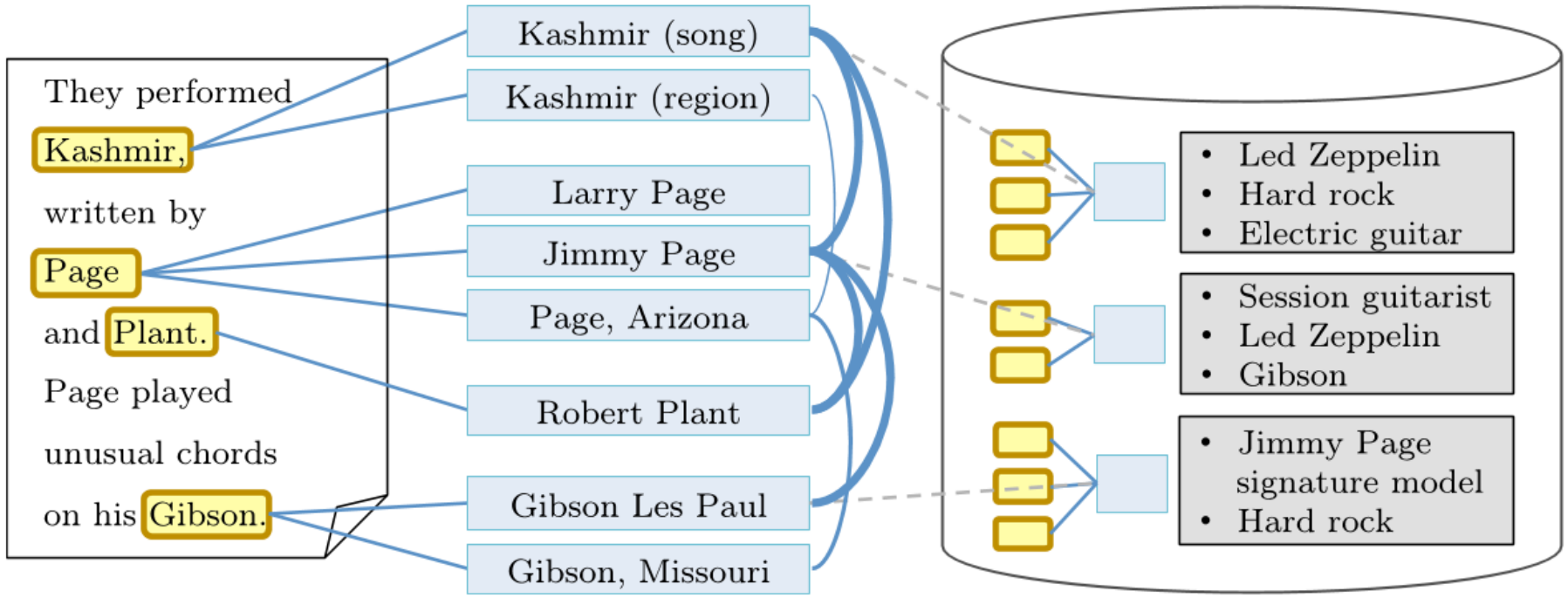
Graph Based Approaches

- The AIDA system [1] models the entity-mention and entity-entity relations as a graph
- The system gives weights to the edges according to some measures that estimate the likelihood of the entity (typically it is a sort of entity popularity measure)

[1] J. Hoffart *et al.*, “Robust disambiguation of named entities in text,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 782–792.

Graph Based Approaches

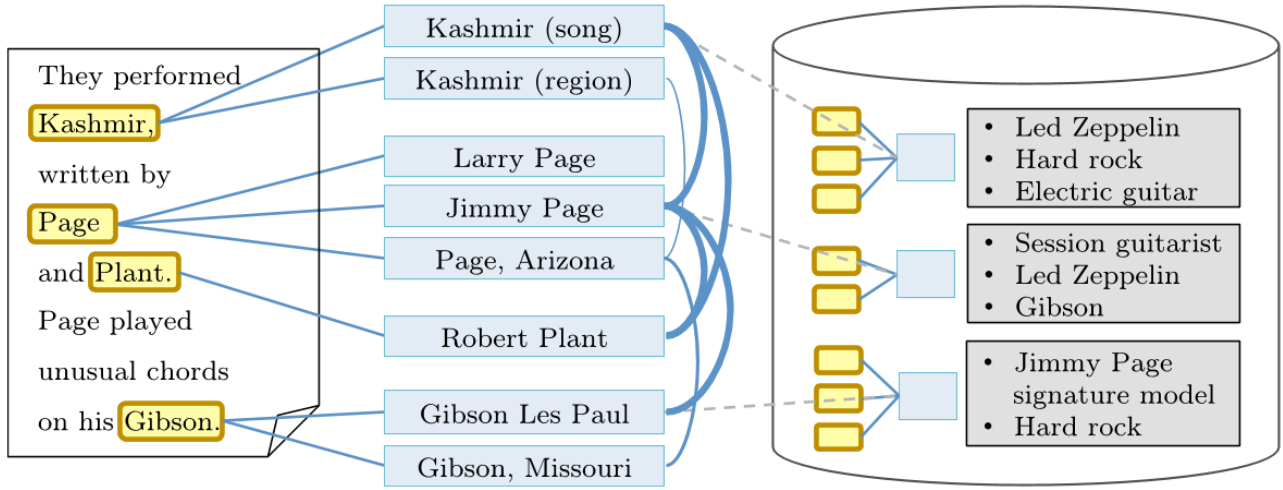
- Example of graph produced by the AIDA system



VU VRIJE UNIVERSITEIT AMSTERDAM

Graph Based Approaches

- Find a subgraph where only one entity-mention edges with maximum weight
- This task is a generalized version of the *Steiner-tree* problem
- The task is NP-hard =>
  they use a greedy algorithm



Unsupervised Ranking Methods

VSM based models

- *Problem*: Obtaining a good training data is hard and expensive
- We can represent both entities and mentions using vectors (VSM – vector space model)
- Pick a similarity measure and choose candidate entity which is the closest to the mention

Unlinkable Mention Prediction

VU VRIJE UNIVERSITEIT AMSTERDAM

Unlinkable Mention Prediction

- How can we detect the case when none of the candidate entities is suitable?
- We can:
  - *Ignore this problem* and assume that recall of Candidate Entity Generation is 1
  - Assume the entity is unlinkable if the entity candidate set *is empty*
  - Use a *threshold value* on the ranking score

Unlinkable Mention Prediction

- How can we detect the case when none of the candidate entities is suitable?
- We can:
  - *Train a binary classifier*
    - Input <mention, top_entity> output 1 if ok, 0 is bad
  - *Add NIL as a special entity* and consider it during the ranking process. If NIL will get the highest score, then the entity is considered unlinkable

State-of-the-Art

VU VRIJE UNIVERSITEIT AMSTERDAM

State-of-the-Art

AIDA

- Uses YAGO2 as KB, Stanford NER to detect the entities
- Works with English and other languages
- Offers public web demo https://gate.d5.mpi-inf.mpg.de/webaida/
- Public API: curl --data text="Dylan was born in Duluth." https://gate.d5.mpi-inf.mpg.de/aida/service/disambiguate

State-of-the-Art

DBPedia Spotlight

- Uses DBpedia as KB, LingPipe as NER
- Relies on VSM for disambiguation
- Works with English and other languages
- Offers public web demo https://dbpedia-spotlight.github.io/demo/