Knowledge Graphs: In Theory and Practice

Sumit Bhatia\textsuperscript{1} and Nitish Aggarwal\textsuperscript{2}

\textsuperscript{1} IBM Research, New Delhi, India
\textsuperscript{2} IBM Watson, San Jose, CA

sumitbhatia@in.ibm.com, nitish.aggarwal@ibm.com

November 10, 2017
Knowledge Graphs Analytics
Knowledge Graph Analytics

- Finding Entities of Interest
  - Entity Search and Recommendation
  - Entity Linking and Disambiguation
- Entity exploration: Knowing more about the entities
  - Relationship Search
  - Path Ranking
- Upcoming challenges
Finding the Right Entities

Google

steve jobs birthday

About 1,84,00,000 results (0.59 seconds)

Steve Jobs / Date of birth

24 February 1955
Finding the Right Entities

Google search for "Singapore telephone code" shows the result +65, which is the dialing code for Singapore.
Entities are the *fundamental units* of a Knowledge graph. How to get to the right entities in the graph?
Entities are the *fundamental units* of a Knowledge graph. How to get to the right entities in the graph? Given a Knowledge Base, $K = \{\mathcal{E}, \mathcal{R}\}$, a document corpus $\mathcal{D}$, and a named entity mention $m$, map/link the mention $m$ to its corresponding entity $e \in \mathcal{E}$. 
Entities are the *fundamental units* of a Knowledge graph. How to get to the right entities in the graph?
Given a Knowledge Base, \( K = \{ \mathcal{E}, \mathcal{R} \} \), a document corpus \( \mathcal{D} \), and a named entity mention \( m \), map/link the mention \( m \) to its corresponding entity \( e \in \mathcal{E} \).
Entities are the *fundamental units* of a Knowledge graph. How to get to the right entities in the graph?

Given a Knowledge Base, $K = \{\mathcal{E}, \mathcal{R}\}$, a document corpus $\mathcal{D}$, and a named entity mention $m$, map/link the mention $m$ to its corresponding entity $e \in \mathcal{E}$.

**Web Queries:**
```
steve jobs birthday
```
Entities are the *fundamental units* of a Knowledge graph. How to get to the right entities in the graph? Given a Knowledge Base, $K = \{E, R\}$, a document corpus $D$, and a named entity mention $m$, map/link the mention $m$ to its corresponding entity $e \in \mathcal{E}$. 

**Web Queries:**
steve jobs birthday

**NL Questions:**
When did Steve resign from Microsoft?
Entities are the *fundamental units* of a Knowledge graph. How to get to the right entities in the graph? Given a Knowledge Base, $K = \{E, R\}$, a document corpus $D$, and a named entity mention $m$, map/link the mention $m$ to its corresponding entity $e \in \mathcal{E}$.

**Web Queries:**
steve jobs birthday

**NL Questions:**
When did Steve resign from Microsoft?

**NL Text:**
....Jobs and Wozniak started Apple Computers from their garage...
• Same entity can be represented by multiple surface forms
Challenges

- **Same entity can be represented by multiple surface forms**
  Barack Obama, Barack H. Obama, President Obama, Senator Obama

- **Same surface form could refer to multiple entities**
  Michael Jordan – Basketball player or Berkeley professor

- **Out of KG mentions**
Challenges

- Same entity can be represented by multiple surface forms
  Barack Obama, Barack H. Obama, President Obama,
  Senator Obama
  President of the United States
Challenges

• Same entity can be represented by multiple surface forms
  Barack Obama, Barack H. Obama, President Obama,
  Senator Obama
  President of the United States

• Same surface form could refer to multiple entities
Challenges

• Same entity can be represented by multiple surface forms
  Barack Obama, Barack H. Obama, President Obama, Senator Obama
  President of the United States

• Same surface form could refer to multiple entities
  Michael Jordan – Basketball player or Berkeley professor
Challenges

- **Same entity can be represented by multiple surface forms**
  - Barack Obama, Barack H. Obama, President Obama, Senator Obama
  - President of the United States

- **Same surface form could refer to multiple entities**
  - Michael Jordan – Basketball player or Berkeley professor
  - when did steve leave apple?
Challenges

• Same entity can be represented by multiple surface forms
  Barack Obama, Barack H. Obama, President Obama, Senator Obama
  President of the United States

• Same surface form could refer to multiple entities
  Michael Jordan – Basketball player or Berkeley professor
  when did steve leave apple?

• Out of KG mentions
Related problems:

- Record linkage/de-duplication in databases
- Entity Resolution/name matching
- Co-reference resolution, Word Sense disambiguation
Entity Linking Process

Well studied in NLP [17]

open source software like Stanford NLP toolkit [16]

Use of dictionaries

Ranking target entities based on:
• graph based features
• text/document based features
Entity Linking Process

Entity Recognition → Target List Generation → Ranking

Well studied in NLP [17]
Open source software like Stanford NLP toolkit [16]

Use of dictionaries

Ranking target entities based on:
• graph based features
• text/document based features
Entity Linking Process

Named Entity Recognition
Well studied in NLP [17]
open source software like Stanford NLP toolkit [16]
Entity Linking Process

Named Entity Recognition
Well studied in NLP [17]
open source software like
Stanford NLP toolkit [16]

Use of dictionaries
Entity Linking Process

Named Entity Recognition
Well studied in NLP [17]
open source software like Stanford NLP toolkit [16]

Use of dictionaries

Ranking target entities based on:
- graph based features
- text/document based features
Candidate Entity List Generation

• Much of the variation between different entity linking algorithms could be explained by quality of candidate search components [12]
• Acronym expansions and coreference resolutions lead to significant performance gains [12]
• The candidate set should be exhaustive enough but not too big to affect efficiency
• Much of the variation between different entity linking algorithms could be explained by quality of candidate search components [12]
• Much of the variation between different entity linking algorithms could be explained by quality of candidate search components [12]

• Acronym expansions and coreference resolutions lead to significant performance gains [12]
Candidate Entity List Generation

- Much of the variation between different entity linking algorithms could be explained by quality of candidate search components [12]

- Acronym expansions and coreference resolutions lead to significant performance gains [12]

- The candidate set should be exhaustive enough but not too big to affect efficiency
Dictionary based Methods
An offline dictionary of entity names created out of external sources mapping different possible surface forms of entity names to their corresponding entities in the KG
Dictionary based Methods
An offline dictionary of entity names created out of external sources mapping different possible surface forms of entity names to their corresponding entities in the KG
  
  - Domain specific sources like Gene name dictionary [18]
Candidate Entity List Generation

Dictionary based Methods
An offline dictionary of entity names created out of external sources mapping different possible surface forms of entity names to their corresponding entities in the KG

- Domain specific sources like Gene name dictionary [18]
- Wikipedia/DBPedia
  - Page Titles
  - Disambiguation/Redirect pages
  - Anchor text of Wikipedia in links
Candidate Entity List Generation

Dictionary based Methods
An offline dictionary of entity names created out of external sources mapping different possible surface forms of entity names to their corresponding entities in the KG

- Domain specific sources like Gene name dictionary [18]
- Wikipedia/DBPedia
  - Page Titles
  - Disambiguation/Redirect pages
  - Anchor text of Wikipedia in links
- Anchor text from Web pages to Wikipedia articles
Dictionary based Methods

An offline dictionary of entity names created out of external sources mapping different possible surface forms of entity names to their corresponding entities in the KG

- Domain specific sources like Gene name dictionary [18]
- Wikipedia/DBPedia
  - Page Titles
  - Disambiguation/Redirect pages
  - Anchor text of Wikipedia in links
- Anchor text from Web pages to Wikipedia articles
- Acronym expansions
<table>
<thead>
<tr>
<th>Surface Form</th>
<th>Entity Canonical Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barack Obama</td>
<td>&lt;Barack Obama, Person&gt;</td>
</tr>
<tr>
<td>Barack H. Obama</td>
<td>&lt;Barack Obama, Person&gt;</td>
</tr>
<tr>
<td>USA</td>
<td>&lt;United States of America, Country&gt;</td>
</tr>
<tr>
<td>America</td>
<td>&lt;United States of America, Country&gt;</td>
</tr>
<tr>
<td>Big Apple</td>
<td>&lt;New York, City&gt;</td>
</tr>
<tr>
<td>NYC</td>
<td>&lt;New York, City&gt;</td>
</tr>
<tr>
<td>NY</td>
<td>&lt;New York, City&gt;</td>
</tr>
</tbody>
</table>
## Candidate Entity List Generation

<table>
<thead>
<tr>
<th>Surface Form</th>
<th>Entity Canonical Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barack Obama</td>
<td>&lt;Barack Obama, Person&gt;</td>
</tr>
<tr>
<td>Barack H. Obama</td>
<td>&lt;Barack Obama, Person&gt;</td>
</tr>
<tr>
<td>USA</td>
<td>&lt;United States of America, Country&gt;</td>
</tr>
<tr>
<td>America</td>
<td>&lt;United States of America, Country&gt;</td>
</tr>
<tr>
<td>Big Apple</td>
<td>&lt;New York, City&gt;</td>
</tr>
<tr>
<td>NYC</td>
<td>&lt;New York, City&gt;</td>
</tr>
<tr>
<td>NY</td>
<td>&lt;New York, City&gt;</td>
</tr>
<tr>
<td>NY</td>
<td>&lt;New York, State&gt;</td>
</tr>
</tbody>
</table>

---

Simple term match – partial or exact... Obama visited Singapore in 2016... Matches: Barack Obama, Mount Obama, Michelle Obama,..., etc.
### Candidate Entity List Generation

<table>
<thead>
<tr>
<th>Surface Form</th>
<th>Entity Canonical Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barack Obama</td>
<td>&lt;Barack Obama, Person&gt;</td>
</tr>
<tr>
<td>Barack H. Obama</td>
<td>&lt;Barack Obama, Person&gt;</td>
</tr>
<tr>
<td>USA</td>
<td>&lt;United States of America, Country&gt;</td>
</tr>
<tr>
<td>America</td>
<td>&lt;United States of America, Country&gt;</td>
</tr>
<tr>
<td>Big Apple</td>
<td>&lt;New York, City&gt;</td>
</tr>
<tr>
<td>NYC</td>
<td>&lt;New York, City&gt;</td>
</tr>
<tr>
<td>NY</td>
<td>&lt;New York, City&gt;</td>
</tr>
<tr>
<td>NY</td>
<td>&lt;New York, State&gt;</td>
</tr>
</tbody>
</table>

Simple term match – partial or exact

...Obama visited Singapore in 2016...

Matches: Barack Obama, Mount Obama, Michelle Obama,..., etc.
Candidate Entity Ranking

The candidate entity set can be big!

For KORE50 dataset:
- 631 candidates on an average per mention in YAGO [23]

Approaches for ranking can be clubbed under two broad categories:
- Text based
- Graph structure based
The candidate entity set can be big!
The candidate entity set can be big!
For KORE50 dataset:

- **631** candidates on an average per mention in YAGO [23]
- **2000+** in Watson KG [4]
The candidate entity set can be big!
For KORE50 dataset:

- 631 candidates on an average per mention in YAGO [23]

Approaches for ranking can be clubbed under two broad categories:

- Text based
- Graph structure based
Candidate Entity Ranking

...Obama is in Hawaii this week...
{Barack Obama, Michelle Obama, Mt. Obama}
Candidate Entity Ranking

...Obama is in Hawaii this week...
{Barack Obama, Michelle Obama, Mt. Obama}

• Similarity between entity name and mention
  • Term overlap, edit distance, etc.
Candidate Entity Ranking

...Obama is in Hawaii this week...
{Barack Obama, Michelle Obama, Mt. Obama}

- Similarity between entity name and mention
  - Term overlap, edit distance, etc.
- Entity Popularity – Wikipedia page views [11, 10]
...Obama is in Hawaii this week...
{Barack Obama, Michelle Obama, Mt. Obama}

- Similarity between entity name and mention
  - Term overlap, edit distance, etc.
- Entity Popularity – Wikipedia page views [11, 10]
- Wikipedia/web anchor text/ inlinks [20, 13]
Candidate Entity Ranking

...Obama is in Hawaii this week...
{Barack Obama, Michelle Obama, Mt. Obama}

• Similarity between entity name and mention
  • Term overlap, edit distance, etc.
• Entity Popularity – Wikipedia page views [11, 10]
  • Wikipedia/web anchor text/inlinks [20, 13]

...when did Steve leave apple...
{Steve Jobs, Steve Wozniak, Steve Ballmer}
Candidate Entity Ranking

...Obama is in Hawaii this week...
{Barack Obama, Michelle Obama, Mt. Obama}
  • Similarity between entity name and mention
    • Term overlap, edit distance, etc.
  • Entity Popularity – Wikipedia page views [11, 10]
  • Wikipedia/web anchor text/ inlinks [20, 13]

...when did Steve leave apple...
{Steve Jobs, Steve Wozniak, Steve Ballmer}
Context Matters!
Role of Context

...when did Steve leave apple...

{Steve Jobs,Steve Wozniak,Steve Ballmer}
Role of Context

...when did Steve leave apple...

{Steve Jobs, Steve Wozniak, Steve Ballmer}

- Mention context
  - text of the document/paragraph in which the mention appears
  - a window of terms around the mention
Candidate Entity Ranking

Role of Context

...when did Steve leave apple...

{Steve Jobs, Steve Wozniak, Steve Ballmer}

- Mention context
  - text of the document/paragraph in which the mention appears
  - a window of terms around the mention
- Entity context representations
  - Wikipedia article
  - Text around anchors
  - Domain specific models: abstracts of papers containing gene name in titles
Role of Context

...when did Steve leave apple...

\{Steve Jobs,Steve Wozniak,Steve Ballmer\}

- Mention context
  - text of the document/paragraph in which the mention appears
  - a window of terms around the mention
- Entity context representations
  - Wikipedia article
  - Text around anchors
  - Domain specific models: abstracts of papers containing gene name in titles

Compute similarity between mention and entity context representations
Graph Based Features Focus on strength between entities, often useful in collective entity linking.
Graph Based Features Focus on strength between entities, often useful in collective entity linking

- Simplest graph based measure – Entity Popularity

\[
pop(e) = \frac{\text{nbrCount}(e)}{\sum_{e' \in \mathcal{E}} \text{nbrCount}(e')}
\]  

In Wikipedia graph, inlinks and outlinks can be used to compute popularity
Graph Based Features Focus on strength between entities, often useful in collective entity linking

• Simplest graph based measure – Entity Popularity

\[
pop(e) = \frac{\text{nbrCount}(e)}{\sum_{e' \in \mathcal{E}} \text{nbrCount}(e')} \tag{1}
\]

In Wikipedia graph, inlinks and outlinks can be used to compute popularity

Next we review some measures useful for collective entity linking
Candidate Entity Ranking

Linking/Resolving/Disambiguating Multiple Entities simultaneously

Image Source: [26]
Brad and Angelina were holidaying in Paris.
Brad and Angelina were holidaying in Paris.

- Jaccard Index

\[ J(a, b) = \frac{|A \cap B|}{|A \cup B|} \]  

(2)

- Milne-Witten Similarity [26]

\[ MW(a, b) = \log(\max(|A|, |B|)) - \log(|A \cap B|) \]

\[ \frac{\log(|N|) - \log(\min(|A|, |B|))}{\log(\max(|A|, |B|))} \]

(3)

where, A and B are the set of neighbors of entities a and b, respectively.

- Adamic Adar [1]

\[ AA(a, b) = \sum_{n \in A \cup B} \log(\frac{1}{\text{degree}(n)}) \]

(4)
Brad and Angelina were holidaying in Paris.

- Jaccard Index
  \[ J(a, b) = \frac{|A \cap B|}{|A \cup B|} \]  

- Milne-Witten Similarity [26]
  \[ MW(a, b) = \frac{\log(max(|A|, |B|)) - \log(|A \cap B|)}{\log(|\mathcal{N}|) - \log(min(|A|, |B|))} \]  

where, \( A \) and \( B \) are the set of neighbors of entities \( a \) and \( b \), respectively.
Brad and Angelina were holidaying in Paris.

- Jaccard Index
  \[ J(a, b) = \frac{|A \cap B|}{|A \cup B|} \]  

- Milne-Witten Similarity [26]
  \[ MW(a, b) = \frac{\log(max(|A|, |B|)) - \log(|A \cap B|)}{\log(|N|) - \log(min(|A|, |B|))} \]

  where, A and B are the set of neighbors of entities a and b, respectively.

- Adamic Adar [1]
  \[ AA(a, b) = \sum_{n \in A \cup B} \log\left(\frac{1}{\text{degree}(n)}\right) \]
These features can be used in supervised or unsupervised settings.

Choice of features depend on data/domain at hand. Many features are specific for Wikipedia, that may not be applicable to other textual data.

Trade off between accuracy and efficiency while designing your systems.
Which search algorithm did Sergey and Larry invent?

Which search algorithm did Sergey and Larry invent?

---

Entity Exploration
We found the entity of interest.

Knowing more about the entity

- Finding entities related to entity of interest
- Properties of entities
- Going beyond immediate neighborhood of the entity
Entity Retrieval

- Entity Box in web queries
- Lots of useful information about the query entity
- \( \approx 40\% \) of all web queries are entity queries [19]
- Many QA queries can be answered by the underlying Knowledge Base
Related Entity Finding track at TREC [3]
Related Entity Finding track at TREC [3]
Input: Entity Name and Search Intent
Related Entity Finding track at TREC [3]
Input: Entity Name and Search Intent
Output: Ranked list of entity documents – entities embedded in documents
Related Entity Finding track at TREC [3]
Input: Entity Name and Search Intent
Output: Ranked list of entity documents – entities embedded in documents
Example:

Query: Blackberry
Intent: Carriers that carry Blackberry phones
Example Answers: Verizon, AT&T, etc.
For a given input entity $e$, type $T$ of target entity, and a relation description $R$, we wish to rank the target entities as follows:

$$P(e | e_s, T, R) \propto P(R | e_s, e)$$

Components of Related Entity Ranking \cite{7}\footnote{M. Bron, K. Balog, and M. De Rijke. “Ranking related entities: components and analyses”. In: Proceedings of the 19th ACM international conference on Information and knowledge management. ACM. 2010, pp. 1079–1088.}
Components of Related Entity Ranking [7]²

For a given input entity \( e_s \), type \( T \) of target entity, and a relation description \( R \), we wish to rank the target entities as follows:

\[
P(e|e_s, T, R) \propto P(R|e_s, e) \times P(e|e_s) \times P(T|e)
\]

(5)

Components of Related Entity Ranking [7]

For a given input entity $e_s$, type $T$ of target entity, and a relation description $R$, we wish to rank the target entities as follows:

$$P(e|e_s, T, R) \propto P(R|e_s, e) \times P(e|e_s) \times P(T|e) \quad (5)$$

---

Co-occurrence

\[
P(e|e_s) = \frac{\text{cooc}(e, e_s)}{\sum_{e' \in E} \text{cooc}(e', e_s)}
\]
Co-occurrence

\[ P(e|e_s) = \frac{cooc(e, e_s)}{\sum_{e' \in E} cooc(e', e_s)} \]

Type Filtering

- Wikipedia categories
- Named entity recognizer tools
Entity Retrieval

Co-occurrence

\[ P(e|e_s) = \frac{\text{cooc}(e, e_s)}{\sum_{e' \in E} \text{cooc}(e', e_s)} \]

Type Filtering

- Wikipedia categories
- Named entity recognizer tools

Context Modeling

Co-occurrence language model \( \Theta_{ee_s} \) approximated by documents in which \( e, E_s \) co-occur

\[ P(R|e, e_s) = \prod_{t \in R} P(t|\Theta_{ee_s}) \]
Entity recommendations for web search queries[6]³

Entity recommendations for web search queries[6]³

• Co-occurrence features
  • query logs, user sessions
  • flickr and twitter tags

Entity recommendations for web search queries[6]³

- Co-occurrence features
  - query logs, user sessions
  - flickr and twitter tags
- frequency

Entity recommendations for web search queries[6]\(^3\)

- Co-occurrence features
  - query logs, user sessions
  - flickr and twitter tags
- frequency
- Graph theoretic features
  - Page rank on entity graph
  - Common neighbors between two entities

Learning to rank using text and graph based features[21]\

\[4\]

Learning to rank using text and graph based features[21]⁴

- Given a web query, retrieve relevant documents,
Learning to rank using text and graph based features[21]⁴

- Given a web query, retrieve relevant documents,
- Identify entities present in them using entity linking methods

Learning to rank using text and graph based features[21]  

- Given a web query, retrieve relevant documents,
- Identify entities present in them using entity linking methods
- Rank these entities using graph theoretic and text based features

---

Learning to rank using text and graph based features[21]\(^4\)

- Given a web query, retrieve relevant documents,
- Identify entities present in them using entity linking methods
- Rank these entities using graph theoretic and text based features
- Reformulates entity retrieval/recommendation as ad hoc document retrieval

Till now, we have focused on finding entities
Till now, we have focused on finding entities
Let us focus our attention now on finding *about* entities
Till now, we have focused on finding entities
Let us focus our attention now on finding *about* entities
Till now, we have focused on finding entities. Let us focus our attention now on finding *about* entities.

Relationships of similar types can be clustered and then explored based on user requirements [27].
What are the most important facts about an entity? \(^5\)

What are the most important facts about an entity? Given a source entity $e_s$, we wish to compute the probability $P(r, e_t|e_s)$

$$P(r, e_t|e_s) \propto P(e_t) \times P(e_s|e_t) \times P(r|e_s, e_t)$$

(6)

Entity Prior: $P(e_t) \propto \text{relCount}(e_t)$

Entity Affinity: $P(e_s|e_t) = \sum_{r_i \in R(e_s, e_t)} w(r_i) \times r_i \sum_{r_i \in R(e_s)} w(r_i) \times r_i$

Relationship Strength: $P(r|e_s, e_t) = \text{mentionCount}(r, e_s, e_t) \sum_{r \in R(e_s, e_t)} \text{mentionCount}(r, e_s, e_t)$

---

What are the most important facts about an entity? Given a source entity $e_s$, we wish to compute the probability $P(r, e_t | e_s)$

$$P(r, e_t | e_s) \propto P(e_t) \times P(e_s | e_t) \times P(r | e_s, e_t)$$  \hspace{1cm} (6)$$

**Entity Prior:**

$$P(e_t) \propto \text{relCount}(e_t)$$  \hspace{1cm} (7)$$

---

What are the most important facts about an entity? Given a source entity $e_s$, we wish to compute the probability $P(r, e_t|e_s)$

$$P(r, e_t|e_s) \propto P(e_t) \times P(e_s|e_t) \times P(r|e_s, e_t)$$

(6)

**Entity Prior:**

$$P(e_t) \propto \text{relCount}(e_t)$$

(7)

**Entity Affinity**

$$P(e|e_t) = \frac{\sum_{r_i \in R(e_s,e_t)} w(r_i) \times r_i}{\sum_{r_i \in R(e_t)} w(r_i) \times r_i}$$

(8)

---

What are the most important facts about an entity? Given a source entity \( e_s \), we wish to compute the probability \( P(r, e_t|e_s) \)

\[
P(r, e_t|e_s) \propto P(e_t) \times P(e_s|e_t) \times P(r|e_s, e_t)
\]

Entity Prior:

\[
P(e_t) \propto \text{relCount}(e_t)
\]

Entity Affinity

\[
P(e|e_t) = \frac{\sum_{r_i \in R(e_s,e_t)} w(r_i) \times r_i}{\sum_{r_i \in R(e_t)} w(r_i) \times r_i}
\]

Relationship Strength

\[
P(r|e_s, e_t) = \frac{\text{mentionCount}(r, e_s, e_t)}{\sum_{r \in R(e_s,e_t)} \text{mentionCount}(r, e_s, e_t)}
\]

---

Entity Exploration - Fact Ranking
Till now, we have limited our attention to relations of the entity and its immediate neighborhood.
Till now, we have limited our attention to relations of the entity and it’s immediate neighborhood. What lies after that?
Discovering and Explaining Higher Order Relations Between Entities
Discovering and Explaining Higher Order Relations Between Entities

Charlie Hebdo Shooting Attack
Discovering and Explaining Higher Order Relations Between Entities
Discovering and Explaining Higher Order Relations Between Entities

Can we tell how are they connected?
Path Ranking

Thousands of such paths

Too generic – obvious relations

CIKM 2017 Knowledge Graphs: In Theory and Practice
• Thousands of such paths
• Too generic – obvious relations
Path Ranking

Three components for ranking possible paths [2]

Specificity: Popular entities given lower scores

\[ \text{spec}(p) = \sum_{e \in p} \text{spec}(e) \]

where:

\[ \text{spec}(e) = \log(1 + \frac{1}{\text{docCount}(e)}) \] (10)

Reduces generic paths, but boosts noise entities

Connectivity: A strongly connected path consists of strong edges.

\[ \text{score}(e_a, e_b) = \vec{d}_{ea} \cdot \vec{d}_{eb} \] (11)

Cohesiveness:

\[ \text{score}(p) = n - 1 \sum_{i=2}^{n} \text{score}(e_i) = n - 1 \sum_{i=2}^{n} \vec{d}_{ei} - 1 \cdot \vec{d}_{ei} + 1 \] (12)
Path Ranking

Three components for ranking possible paths [2]

**Specificity**: Popular entities given lower scores

$$
\text{spec}(p) = \sum_{e \in p} \text{spec}(e); \text{ where: } \text{spec}(e) = \log(1 + 1/\text{docCount}(e)) \quad (10)
$$

Reduces generic paths, but boosts noise entities
Path Ranking

Three components for ranking possible paths [2]

**Specificity:** Popular entities given lower scores

\[
spec(p) = \sum_{e \in p} spec(e); \text{ where: } spec(e) = \log(1 + 1/docCount(e)) \tag{10}\]

Reduces generic paths, but boosts noise entities

**Connectivity:** A strongly connected path consists of strong edges.

\[
\text{score}(e_a, e_b) = \vec{d}_{e_a} \cdot \vec{d}_{e_b} \tag{11}\]
Path Ranking

Three components for ranking possible paths [2]

Specificity: Popular entities given lower scores

\[ \text{spec}(p) = \sum_{e \in p} \text{spec}(e); \text{ where: } \text{spec}(e) = \log(1 + 1/\text{docCount}(e)) \]  (10)

Reduces generic paths, but boosts noise entities

Connectivity: A strongly connected path consists of strong edges.

\[ \text{score}(e_a, e_b) = \vec{d}_{ea} \cdot \vec{d}_{eb} \]  (11)

Cohesiveness:

\[ \text{score}(p) = \sum_{i=2}^{n-1} \text{score}(e_i) = \sum_{i=2}^{n-1} \vec{d}_{ei-1} \cdot \vec{d}_{ei+1} \]  (12)
Wikipedia:
Aamir Khan's Bollywood movie Ghajini was the remake of Hollywood movie Memento directed by Christopher Nolan.
Path Ranking

Wikipedia:
Shortly after the Syrian uprising began against the Syrian administration headed by Syrian president Bashar al-Assad, al-Julani moved into Syrian territory and, fully supported by al-Baghdadi.... Bashar was supported by major general Qasem Soleimani....
Application Example from Life Sciences

Predicting Drug-Drug Interactions (DDI)\textsuperscript{6}

Predicting Drug-Drug Interactions (DDI)\(^6\)

- DDI are a major cause of preventable adverse drug reactions

---

Predicting Drug-Drug Interactions (DDI)\textsuperscript{6}

- DDI are a major cause of preventable adverse drug reactions
- Clinical studies can not accurately determine all possible DDIs

Predicting Drug-Drug Interactions (DDI)\textsuperscript{6}

- DDI are a major cause of preventable adverse drug reactions
- Clinical studies can not accurately determine all possible DDIs
- Can we utilize knowledge about drugs to predict possible DDIs?

Create a KG out of existing information about drugs and their interactions with genes, enzymes, molecules, etc.
Application Example from Life Sciences

• Given a pair of drugs, extract features based on physiological effect, side effect, targets, drug targets, chemical structure, etc.

• Perform supervised classification using logistic regression

• Retrospective Analysis: Known DDIs til January 2011 as training.

• Could predict \( \approx 68\% \) of DDIs discovered after January 2011 till December 2014.
Future Research Directions

• Reasoning over Knowledge Graphs
• KG Completion [8, 22, 15]
• Complex QA Systems
• Explaining relations present in a graph [24, 14]
• Graph and text joint modeling [25, 28]
• Ask domain experts!
Future Research Directions

• Reasoning over Knowledge Graphs
  • KG Completion [8, 22, 15]
  • Complex QA Systems
Future Research Directions

• Reasoning over Knowledge Graphs
  • KG Completion [8, 22, 15]
  • Complex QA Systems

• Explaining relations present in a graph [24, 14]
Future Research Directions

- Reasoning over Knowledge Graphs
  - KG Completion [8, 22, 15]
  - Complex QA Systems
- Explaining relations present in a graph [24, 14]
- Graph and text joint modeling [25, 28]
Future Research Directions

- Reasoning over Knowledge Graphs
  - KG Completion [8, 22, 15]
  - Complex QA Systems
- Explaining relations present in a graph [24, 14]
- Graph and text joint modeling [25, 28]
- Ask domain experts!
DEMO
Conclusions

- KG can provide structure to your unstructured data!
Conclusions

• KG can provide structure to your unstructured data!
• We wanted to provide an overview of tools/techniques that have worked well in the past, and challenges you may face

should help you get started with a pretty strong baseline system

be careful in selecting the KG appropriate for your domain and requirements.

keep in mind the scale and efficiency issues

you will have to work with lots of noisy and erroneous data

but the efforts required are worth it!
Conclusions

• KG can provide structure to your unstructured data!
• We wanted to provide an overview of tools/techniques that have worked well in the past, and challenges you may face
• Should help you get started with a pretty strong baseline system
Conclusions

• KG can provide structure to your unstructured data!
• We wanted to provide an overview of tools/techniques that have worked well in the past, and challenges you may face
• Should help you get started with a pretty strong baseline system
• Be careful in selecting the KG appropriate for your domain and requirements.
Conclusions

• KG can provide structure to your unstructured data!
• We wanted to provide an overview of tools/techniques that have worked well in the past, and challenges you may face
• Should help you get started with a pretty strong baseline system
• Be careful in selecting the KG appropriate for your domain and requirements.
• Keep in mind the scale and efficiency issues
Conclusions

• KG can provide structure to your unstructured data!
• We wanted to provide an overview of tools/techniques that have worked well in the past, and challenges you may face
• Should help you get started with a pretty strong baseline system
• Be careful in selecting the KG appropriate for your domain and requirements.
• Keep in mind the scale and efficiency issues
• You will have to work with lots of noisy and erroneous data
Conclusions

• KG can provide structure to your unstructured data!
• We wanted to provide an overview of tools/techniques that have worked well in the past, and challenges you may face
• Should help you get started with a pretty strong baseline system
• Be careful in selecting the KG appropriate for your domain and requirements.
• Keep in mind the scale and efficiency issues
• You will have to work with lots of noisy and erroneous data
• But the efforts required are worth it!
Thanks!!!
Suggestions and Questions Welcome!

Slides available at http://sumitbhatia.net/source/knowledge-graph-tutorial.html


