Query-Driven Knowledge Base Completion with Multimodal Fusion

Yang Peng
Data Science Research Lab
University of Florida
Outline

• Introduction
• Recap of Work before Candidacy
• Knowledge Base Completion
  – Query-Driven Web-Based Question Answering
  – Augmented Rule Inference based on Question Answering
  – Ensemble Fusion
• Conclusions
Introduction

• Multimodal Data
The University of Florida (commonly referred to as Florida or UF) is an American public land-grant, sea-grant, and space-grant research university on a 2,000-acre (8.1 km²) campus in Gainesville, Florida. It is a senior member of the State University System of Florida and traces its origins to 1853,[7] and has operated continuously on its Gainesville campus since September 1906.[8]

Introduction

• Multimodal Data
  – Unstructured text
  – Structured knowledge
  – Images
  – etc.
Multimodal Fusion

- Multimodal fusion is the use of algorithms to combine information from different kinds of data with the purpose of achieving better performance.

- This dissertation focuses on employing multimodal fusion on multimodal data to improve performance for various tasks, as well as providing scalability and high efficiency.
Recap

• Discovered the correlative and complementary relations between different modalities

• Multimodal data can either provide additional information or emphasize the same information

• Multimodal ensemble fusion model for word sense disambiguation and information retrieval
  – Achieve better performance than image-only or text-only approaches
Recap

• **Streaming fact extraction**
  – Process 5TB text data in less than 1 hour
  – Extract hundreds of thousands facts for 13 relations

• **Scalable image retrieval on Hadoop**
  – Process millions of images in less than 2 hour
  – Propose two distributed K-Means algorithms, hundreds of times faster than normal K-Means
Knowledge Bases

• Huge knowledge bases such as Freebase, NELL, and YAGO

• A large knowledge graph of entities, relations and facts
Google Knowledge Graph
Knowledge Bases

• Facts stored in triples
  – <subject, relation, object>, e.g., <Marvin_Minsky, wasBornIn, New_York_City>
  – Marvin Lee Minsky (August 9, 1927 -- January 24, 2016) was an American cognitive scientist concerned largely with research of artificial intelligence (AI), co-founder of the Massachusetts Institute of Technology’s AI laboratory, and author of several texts concerning AI and philosophy. [https://en.wikipedia.org/wiki/Marvin_Minsky](https://en.wikipedia.org/wiki/Marvin_Minsky)
Knowledge Base Completion

• Huge knowledge bases such as Freebase, NELL, and YAGO

• Despite large size, they are highly incomplete
Knowledge Base Completion

- Known Place of Birth: 30%
- Unknown Place of Birth: 70%
Knowledge Base Completion

- Knowledge Base Completion
  - Knowledge base completion (KBC) is the task to fill in the gaps in knowledge bases in a targeted way
Knowledge Base Completion

- Knowledge Base Completion
  - Knowledge base completion (KBC) is the task to fill in the gaps in knowledge bases in a targeted way
  - \(<\texttt{subject}, \texttt{relation}, \texttt{?} >\): given the subject and relation, what is the corresponding object value(s)?
  - Example
    - Query: \(<\texttt{Marvin\_Minksy}, \texttt{wasBornIn}, \texttt{?} >\)
    - Answer: \texttt{New\_York\_City}
Related Work

• Knowledge Base Construction/Population
  – Fact extraction from unstructured text datasets
  – Human workers to add information
Related Work

• **Knowledge Base Construction/Population**
  – Fact extraction from unstructured text datasets
  – Human workers to add information

• **Drawbacks**
  – Time-consuming
  – Intense human labor
Related Work

• Inference and Learning
  – Logical rules
  – Probabilistic graphical models
  – Embeddings and Matrix Factorization
Related Work Cont.

• Inference and Learning
  – Logical rules
  – Probabilistic graphical models
  – Embeddings and Matrix Factorization

• Drawbacks
  – Learning effective, efficient and expressive models is difficult
  – Knowledge bases are highly incomplete
• **Question Answering**
  - Robert West, Evgeniy Gabrilovich, Kevin Murphy, etc. 2014. Knowledge base completion via search-based question answering. In *Proceedings of the 23rd international conference on World wide web (WWW '14)*
  - Parse structured queries to natural language questions and use an in-house question answering system to get answers
Related Work Cont.

• Question Answering
  – Robert West, Evgeniy Gabrilovich, Kevin Murphy, etc. 2014. Knowledge base completion via search-based question answering. In Proceedings of the 23rd international conference on World wide web (WWW '14)
  – Parse structured queries to natural language questions and use an in-house question answering system to get answers

• Drawbacks
  – Low recall for unpopular relations and entities
Main Problems with Related Work

• Uses either only unstructured textual information or only structured information
  – Lower recall/precision

• Batch-oriented KBC
  – User query answers may still be missing
Our Approach

• Propose a query-driven pipeline with multimodal fusion for knowledge base completion

• Enhance web-based question answering and rule inference using both unstructured text and structured knowledge for knowledge base completion

• To our best knowledge, this is the first system for query-time knowledge base completion fusing unstructured and structured data
Novelty

• Fuse unstructured text and structured knowledge bases to improve performance

• Combine novel text-based approaches and knowledge-based approaches

• Query-driven vs batch-oriented
System Overview

• Design a query-driven web-based question answering system to extract missing facts from textual snippets crawled from the Web
System Overview

• Design a query-driven web-based question answering system to extract missing facts from textual snippets crawled from the Web

• Build an augmented rule inference system based on web-based question answering, logical rules and existing facts inside KBs to infer missing facts
System Overview

• Design a query-driven web-based question answering system to extract missing facts from textual snippets crawled from the Web

• Build an augmented rule inference system based on web-based question answering, logical rules and existing facts inside KBs to infer missing facts

• Use ensemble fusion approaches to combine question answering and rule inference
System Pipeline

Web-Based Question Answering

Knowledge Bases

Rule Inference

Ensemble Fusion

relevant textual snippets

metadata, entity semantics, etc.

logical rules, existing facts, etc.

final results
System Pipeline

Web-Based Question Answering

Knowledge Bases

Rule Inference

Ensemble Fusion

System Pipeline

Web
relevant textual snippets

candidate answers

metadata, entity semantics, etc.

logical rules, existing facts, etc.

final results

<subject, relation, ?>

<subject, relation, ?>

candidate answers

candidate answers
Web-Based Question Answering (WebQA)

• Parse queries to natural language questions, search on the Web and extract candidate answers from snippets
WebQA Pipeline

Web-Based Question Answering

- Question Generation
- Data Collection
- Answer Extraction
- Answer Ranking

Set of questions
Set of snippets
Set of candidate answers
Ranked answers with confidence scores

<subject, relation, ?>
Example: WebQA Pipeline

**Query:** `<Marvin_Minsky, wasBornIn, ?>`

- **Questions:**
  - Marvin Minsky born;
  - Marvin Minsky birth;

- **Snippets:**
  - Marvin Lee Minsky was born in New York City, to an eye surgeon father, Henry, and to a mother, Fannie ...

- **Candidate Answers:**
  - New-York-City
  - Boston

- **Ranked Answers:**
  - New_York_City: 0.95
  - Boston: 0.4

Web-Based Question Answering
• Use **question templates** to transform queries to questions
  – Each relation has multiple question templates
  – wasBornIn: *born, birth* and *birthplace*

**Snippets:**
*Marvin Lee Minsky was born in New York City, to an eye surgeon father, Henry, and to a mother, Fannie.*

**Questions:**
- Marvin Minsky born;
- Marvin Minsky birth;
...

**Candidate Answers:**
- New_York_City
- Boston
...

**Ranked Answers:**
- New_York_City: 0.95
- Boston: 0.4
...
Data Collection

• Use search engines to crawl snippets by searching questions on the Web
  – Another example snippet: **Marvin Minsky** - A.M. Turing Award Winner, **BIRTH**: New York City, August 9, 1927. **DEATH**: Boston, January 24, 2016 ...
**Answer Extraction**

Snippets:
*Marvin Lee Minsky* was born in *New York City*, to an eye surgeon father, Henry, and to a mother, Fannie ...

Questions:
Marvin Minsky born; Marvin Minsky birth; ...

Candidate Answers:
New_York_City; Boston ...

Ranked Answers:
New_York_City: 0.95; Boston: 0.4 ...

• Extract noun phrases from snippets and link them to entities in KBs, for example:
  – **New_York_City**: a city entity
  – **Henry_Minsky**: a person entity
  – Category filtering
Answer Ranking

- Extract **multimodal features** from textual snippets and knowledge bases for candidate answers and use a **classifier** to classify and rank them.
Details on Answer Ranking (I)
Multimodal Features

• Fuse information from unstructured textual snippets and structured knowledge bases
  – **count**: the number of times a candidate answer appearing in snippets
  – **average rank**: the average rank of the snippets a candidate answer appears in
  – **keyword count**: the number of times question keywords appearing together with the candidate answer in snippets
  – **context similarity** between the context of the candidate answer and the abstract of the query entity
  – **abstract similarity** between the abstracts of the candidate answer and the query entity
  – **relatedness** between the candidate answer and the query entity inside the knowledge base
Details on Answer Ranking (II) Classification & Ranking

• **Imbalanced datasets**
  – # negative samples / # positive samples > 30
  – Resampling (better)
  – Cost reweighting: false negatives have higher cost

• **Classifiers**
  – Logistic regression (best)
  – Decision tree
  – Support vector machines
Query-Driven Techniques

• Question Template Selection

• Query-Driven Snippet Filtering
Question Template Selection

• Each relation has multiple templates
  – Increase the chance of finding correct answers

• Issuing all possible questions is problematic
  – Long running time
  – Performance deterioration
Question Template Selection

• Propose a greedy selection algorithm to select a minimal set of templates with the highest performance

• Greedy strategy was the best strategy according to previous work (WWW’14)
Greedy Selection Algorithm

• Four templates, $t_1$, $t_2$, $t_3$, $t_4$, in descending order of individual performance

• Previous approach greedily selects the templates according to their performance ranking
  – $\{t_1\}$
  – $\{t_1, t_2\}$
  – $\{t_1, t_2, t_3\}$
  – $\{t_1, t_2, t_3, t_4\}$
Greedy Selection Algorithm

• However, previous work ignored the correlation between templates
• Our greedy strategy is to choose the templates which work best together
  – \{t_1\}
  – \{t_1, t_3\}: \{t_1\} works better with \(t_3\) than \(t_2\) and \(t_4\)
  – \{t_1, t_3, t_4\}: \{t_1, t_3\} works better with \(t_4\) than \(t_2\)
  – \{t_1, t_3, t_4, t_2\}
Greedy Selection Algorithm

**Algorithm 1** Greedy selection algorithm

1. $T = \{t_1, t_2, \ldots, t_n\}$: the set of n question templates
2. $Q = \emptyset$: current selected question templates
3. $QS = \emptyset$: the set of different sets of question templates
4. **for** $i = 1; i \leq n; i++$ **do**
5. Select $t_j$ from $T$ such that $Q \cup \{t_j\}$ has the highest performance for all possible $t$ in $T$
6. $Q = Q \cup \{t_j\}$
7. $QS = QS \cup \{Q\}$
8. $T = T - \{t_j\}$
9. Select $Q_m$ from $QS$ with the highest performance and smallest size
10. **return** $Q_m$
Datasets, Benchmark and Metric

• **Dataset:** Yago2 knowledge base
  – more than 10 million entities (like persons, organizations, cities, etc.) and more than 120 million facts about these entities

• **Query Benchmark**
  – 8 relations: wasBornIn, diedIn, hasChild, hasCapital, isMarriedTo, isCitizenOf, graduatedFrom, hasAcademicAdvisor
  – Randomly select 500 queries for training and 100 queries for testing from Yago for each relation
Datasets, Benchmark and Metric

• Mean average precision (MAP) as metric
  – Mean of average precision scores over all queries
  – Average precision (AP): considering both precision and ranking to evaluate a ranked list of answers
    • Ranked list: \{1, 0, 0, 1, 0, 0\}
    • AP = (1/1 + 2/4) / 2 = 0.75
Template Selection Evaluation

$K$ is the number of questions
Template Selection Evaluation

• 2 or 3 templates are enough for most relations

• We use much fewer templates than previous work to achieve better performance

• Compare four relations with previous work
Results for Template Selection

The performance of WebQA vs previous work (WWW’14 paper), measured by mean average precision

<table>
<thead>
<tr>
<th>Template</th>
<th>WebQA</th>
<th>WWW’14</th>
</tr>
</thead>
<tbody>
<tr>
<td>isMarriedTo</td>
<td>0.52</td>
<td>0.5</td>
</tr>
<tr>
<td>hasChild</td>
<td>0.24</td>
<td>0.18</td>
</tr>
<tr>
<td>wasBornIn</td>
<td>0.75</td>
<td>0.67</td>
</tr>
<tr>
<td>isCitizenOf</td>
<td>0.45</td>
<td>0.93</td>
</tr>
</tbody>
</table>

WWW’14 used top searched entities while we use randomly selected entities.
Query-Driven Snippet Filtering

- 50 snippets for each question (WWW’14), 100 – 150 snippets for each KBC query

- Entity linking on all snippets is time-consuming
  - Server lookups
  - Multi-threading is not enough

- Not all snippets are good
Query-Driven Snippet Filtering

• Features
  – The original rank of a snippet returned by a search engine
  – A boolean indicator about whether the keyword in question templates appearing in the snippet or not,
  – The number of words of entity names appearing in the snippet

• Logistic regression with resampling
Results for Snippet Filtering
WebQA Efficiency

• Bottleneck
  – Network delay
  – Server-side delay

• Solution
  – Use much fewer questions
  – Snippet filtering
Runtime Results for Query-Driven Techniques

Average runtime of WebQA for relation wasBornIn with 3 questions

- 12.5 seconds for 150 snippets
- 4.1 seconds for 50 snippets
- 3.2 seconds for 30 snippets
- 3.1 seconds for 20 snippets
System Pipeline

Web-Based Question Answering

Web

relevant textual snippets

metadata, entity semantics, etc.

Rule Inference

logical rules, existing facts, etc.

Knowledge Bases

candidate answers

Web-Based Question Answering

(candidate answers)

Ensemble Fusion

final results

<subject, relation, ?>

<subject, relation, ?>
Rule Inference

- Logical rules pre-learned from knowledge bases to infer missing facts
- Horn-clause rules

\[
\text{isMarriedTo}(x,y), \quad \text{hasChild}(y,z) \implies \text{hasChild}(x,z); \quad 0.62
\]
Rules

• **Length-1 rules**
  
  – e.g. `diedIn(x, City_A) => wasBornIn(x, City_A);`
  
  confidence score = 0.12

• **Length-2 rules**
  
  – e.g. `isMarriedTo(x, y), hasChild(y, z) => hasChild(x, z);`
  
  confidence score = 0.62
Ordinary Rule Inference (OrdRI)

• Facts of KBs are stored in database tables

• Use only existing facts inside knowledge bases
  – SQL query

• Each relation has multiple rules
  – Parallelization using multi-threading
Ordinary Rule Inference (OrdRI)

• Fusion of results from multiple rules
  – Sum score
  – Maximum score
  – Logistic regression (average confidence score, total # of rules, etc.)
Problem

• Using only existing facts in knowledge bases
  – Knowledge bases are highly incomplete, hence many premises or body predicates of rules are missing

• Solution
  – Use WebQA to extract missing body literals from the Web and use rules to infer missing head facts
  – Query-driven
Augmented Rule Inference with WebQA (AugRI)

Working for length-1 rules, such as diedIn(x, y) => wasBornIn(x, y)
Augmented Rule Inference with WebQA (AugRI)

Working for length-2 rules, such as
\text{isMarriedTo}(x, y), \text{hasChild}(y, z) \Rightarrow \text{hasChild}(x, z)
Answer Confidence Score

• Length-1 rule
  – diedIn(x, y) => wasBornIn(x, y); 0.12
  – score(y) = score(diedIn(x, y)) \times score_r
  – If diedIn(x,y) exists inside KBs, its confidence is 1; else its confidence is the score returned by WebQA

• Length-2 rule
  – isMarriedTo(x, y), hasChild(y, z) => hasChild(x, z); 0.62

\[ score_r \times \sum_y \text{score(isMarriedTo}(x, y)) \times \text{score(hasChild}(y, z)) \]
Fusion of Multiple Rules

• Similar to ordinary rule inference
Query-Driven Optimization

• Length-2 Rules
  – Could possibly launch too many WebQA queries for the second literal
Query-Driven Optimization

• Length-2 Rules
  – Could possibly launch too many WebQA queries for the second literal

• Use *threshold* to filter out low-confidence results from the first step

• Only use top $k$ answers from the first step

• Empirical parameter learning
Results Comparing Ordinary Rule Inference and Augmented Rule Inference

MAP of OrdRI vs AugRI on the Yago benchmark

graduatedFrom: OrdRI 0.06, AugRI 0.17

diedIn: OrdRI 0.2, AugRI 0.3

hasChild: OrdRI 0.22, AugRI 0.24

isCitizenOf: OrdRI 0.08, AugRI 0.53

wasBornIn: OrdRI 0.12, AugRI 0.49

isMarriedTo: OrdRI 0.96, AugRI 0.96
System Pipeline

Web

relevant textual snippets

<subject, relation, ?>

Web-Based Question Answering

metadata, entity semantics, etc.

Knowledge Bases

logical rules, existing facts, etc.

Rule Inference

candidate answers
candidate answers
candidate answers

Ensemble Fusion

final results
Ensemble Fusion

- Candidate answers with confidence scores from WebQA and AugRI

- Ensemble approaches
  - Linear rule
  - Maximum rule
  - Sum rule
  - Logistic regression
Ensemble Fusion of Question Answering and Rule Inference

KBC Performance of individual approaches and fusion approaches (measured by MAP). WebQA is conducted with 30 snippets.
System Efficiency

• In average, very fast, about 4 – 5 seconds, because query-driven optimization guarantees only very few WebQA queries are initiated

• Future improvements
  – Use in-house entity linking
  – Use faster search engines
Conclusions

• Propose a query-driven pipeline with multimodal fusion for knowledge base completion, fusing unstructured and structured information
• Design a novel web-based question answering (WebQA) system with multimodal features, question template selection and query-driven snippet filtering
• Build a novel rule inference system with logical rules, WebQA and query-driven optimization
• Achieve state-of-the-art performance
• Provide fast real-time responses to user queries on-the-fly
Publications


Publications


Acknowledgments

Committee
• Prof. Daisy Zhe Wang
• Prof. Shigang Chen
• Prof. Sartaj Sahni
• Prof. Sanjay Ranka
• Prof. Tan Wong

Collaborators & Lab-mates
• Dr. Kun Li
• Dr. Christian Grant
• Dr. Morteza Shahriari Nia
• Dr. Yang Chen
• Xiaofeng Zhou
• Sean Goldberg
• Miguel Rodríguez
• Dihong Gong
• Ali Sadeghian
• etc.
Thank you!

• Questions?