DEANNA: Natural Language Questions for the Web of Data

Mohamed Yahya†
Maya Ramanath‡
† Max Planck Institute for Informatics

Klaus Berberich†
Volker Tresp#
* Qatar Computing Research Institute

Shady Elbassiousni*
Gerhard Weikum†
‡ IIT Delhi
# Siemens
“Classical” QA

“Who played in Casablanca and was married to a writer born in Rome?”
Ingrid Bergman (29 August 1915 – 29 August 1982) was a Swedish actress who starred in a variety of European and American films. She won three Academy Awards, two Emmy Awards, and the Tony Award for Best Actress. She is ranked as the fourth greatest female star of American cinema of all time by the American Film Institute. She is best remembered for her roles as Ilsa Lund in *Casablanca* (1942), a World War II drama co-starring Humphrey Bogart and as Alicia Huberman in *Notorious* (1946), an Alfred Hitchcock thriller co-starring Cary Grant.
Ingrid Bergman (29 August 1915 – 29 August 1982) was a Swedish actress who starred in a variety of European and American films. She won three Academy Awards, two Emmy Awards, and the Tony Award for Best Actress. She is ranked as the fourth greatest female star of American cinema of all time by the American Film Institute. She is best remembered for her roles as Ilsa Lund in *Casablanca* (1942), a World War II drama co-starring Humphrey Bogart, and as Alicia Huberman in *Notorious* (1946), an Alfred Hitchcock thriller co-starring Cary Grant. In 1950, after a decade of stardom in American films, she starred in the Italian film *Stromboli*, which led to a love affair with director Roberto Rossellini while they were both already married. The affair and then marriage with Rossellini created a scandal that forced her to remain in Europe until 1956, when she made a successful Hollywood return in *Anastasia*, for which she won her second Academy Award, as well as the forgiveness of her fans. Many of her personal and film documents can be seen in the Wesleyan University Cinema Archives. [4]
**Ingrid Bergman** (29 August 1915 – 29 August 1982) was a Swedish actress who starred in a variety of European and American films. She won three Academy Awards, two Emmy Awards, and the Tony Award for Best Actress. She is ranked as the fourth greatest female star of American cinema of all time by the American Film Institute. She is best remembered for her roles as Ilsa Lund in *Casablanca* (1942), a World War II drama co-starring Humphrey Bogart, and as Alicia Huberman in *Notorious* (1946), an Alfred Hitchcock thriller co-starring Cary Grant.

In 1950, after a decade of stardom in American films, she starred in the Italian film *Stromboli*, which led to a love affair with director Roberto Rossellini while they were both already married. The affair and then marriage with Rossellini created a scandal that forced her to remain in Europe until 1956, when she made a successful Hollywood return in *Anastasia*, for which she won her second Academy Award, as well as the forgiveness of her fans. Many of her personal and film documents can be seen in the Wesleyan University Cinema Archives.

**Roberto Gastone Zeffiro Rossellini** (8 May 1906 – 3 June 1977) was an Italian film director and screenwriter. Rossellini was one of the directors of the Italian neorealist cinema, contributing films such as *Roma città aperta* (*Rome, Open City* 1945) to the movement.
Ingrid Bergman (29 August 1915 – 29 August 1982) was a Swedish actress who starred in a variety of European and American films. She won three Academy Awards, two Emmy Awards, and the Tony Award for Best Actress, ranking as the fourth greatest female star of American cinema of all time by the American Film Institute. She is best remembered for her roles as Ilsa Lund in *Casablanca* (1942), a World War II drama co-starring Humphrey Bogart, and as Alicia Huberman in *Notorious* (1946), an Alfred Hitchcock thriller co-starring Cary Grant. In 1950, after a decade of stardom in American films, she starred in the Italian film *Stromboli*, which led to a love affair with director Roberto Rossellini while they were both already married. The affair and their subsequent marriage with Rossellini created a scandal that forced her to remain in Europe until 1956, when she made a successful Hollywood return in *Anastasia* for which she won her second Academy Award, as well as the forgiveness of her fans. Many of her personal and film documents can be seen in the Wesleyan University Cinema Archives.

Roberto Gastone Zeffiro Rossellini (8 May 1906 – 3 June 1977) was an Italian film director and screenwriter. Rossellini was one of the directors of the Italian neorealist cinema, contributing films such as *Roma città aperta* (*Rome, Open City* 1945) to the movement. His mother, Elettra (née Bellan), was a housewife, and his father, Angiolo Giuseppe "Beppino" Rossellini, owned a construction firm. His mother was of part French descent, from immigrants who had arrived in Italy during the Napoleonic Wars. He lived on the Via Ludovisi, where Benito Mussolini had his first Roman hotel in 1922 when Fascism obtained power in Italy.
QA, meet the (semantic) Web of Data
QA, meet the (semantic) Web of Data

As of September 2011

Natural Language Questions for the Web of Data - Yahya et al.
QA, meet the (semantic) Web of Data
Subject: Rome  
City

Predicate: isA  
SubclassOf

Object: city  
Location

Roberto_Rossellini  
marriedTo  
Ingrid_Bergman
QA, meet the (semantic) Web of Data

WordNet

+ 

Ingrid Bergman

Roberto Rossellini

WordNet + YAGO2: 120 million facts on 10 million entities

Roman isA city subclassOf location

Roberto_Rossellini marriedTo Ingrid_Bergman
WordNet +

Subject: Rome
Predicate: isA
Object: city

Subject: Roberto_Rossellini
Predicate: marriedTo
Object: Ingrid_Bergman

YAGO2: 120 million facts on 10 million entities
QA, meet the (semantic) Web of Data

WordNet

+ 

MusicBrainz

YAGO2: 120 million facts on 10 million entities

Subject
Rome

city

Roberto_Rossellini

Predicate
isA

subclassOf

location

Object
city

Ingrid_Bergman

marriedTo

Roberto_Rossellini

marriedTo

Ingrid_Bergman

Rome

isA

city

class

WordNet

+ 

MusicBrainz

YAGO2: 120 million facts on 10 million entities

Subject
Rome

city

Roberto_Rossellini

Predicate
isA

subclassOf

location

Object
city

Ingrid_Bergman

marriedTo

Roberto_Rossellini

marriedTo

Ingrid_Bergman

Rome

isA

city

WordNet

+ 

MusicBrainz

YAGO2: 120 million facts on 10 million entities

Subject
Rome

city

Roberto_Rossellini

Predicate
isA

subclassOf

location

Object
city

Ingrid_Bergman

marriedTo

Roberto_Rossellini

marriedTo

Ingrid_Bergman

Rome

isA

city

WordNet

+ 

MusicBrainz

YAGO2: 120 million facts on 10 million entities

Subject
Rome

city

Roberto_Rossellini

Predicate
isA

subclassOf

location

Object
city

Ingrid_Bergman

marriedTo

Roberto_Rossellini

marriedTo

Ingrid_Bergman

Rome

isA

city

WordNet

+ 

MusicBrainz

YAGO2: 120 million facts on 10 million entities

Subject
Rome

city

Roberto_Rossellini

Predicate
isA

subclassOf

location

Object
city

Ingrid_Bergman

marriedTo

Roberto_Rossellini

marriedTo

Ingrid_Bergman

Rome

isA

city

WordNet

+ 

MusicBrainz

YAGO2: 120 million facts on 10 million entities

Subject
Rome

city

Roberto_Rossellini

Predicate
isA

subclassOf

location

Object
city

Ingrid_Bergman

marriedTo

Roberto_Rossellini

marriedTo

Ingrid_Bergman

Rome

isA

city

WordNet

+ 

MusicBrainz

YAGO2: 120 million facts on 10 million entities

Subject
Rome

city

Roberto_Rossellini

Predicate
isA

subclassOf

location

Object
city

Ingrid_Bergman

marriedTo

Roberto_Rossellini

marriedTo

Ingrid_Bergman

Rome

isA

city

WordNet

+ 

MusicBrainz

YAGO2: 120 million facts on 10 million entities

Subject
Rome

city

Roberto_Rossellini

Predicate
isA

subclassOf

location

Object
city

Ingrid_Bergman

marriedTo

Roberto_Rossellini

marriedTo

Ingrid_Bergman

Rome

isA

city

WordNet

+ 

MusicBrainz

YAGO2: 120 million facts on 10 million entities

Subject
Rome

city

Roberto_Rossellini

Predicate
isA

subclassOf

location

Object
city

Ingrid_Bergman

marriedTo

Roberto_Rossellini

marriedTo

Ingrid_Bergman

Rome

isA

city

WordNet

+ 

MusicBrainz

YAGO2: 120 million facts on 10 million entities

Subject
Rome

city

Roberto_Rossellini

Predicate
isA

subclassOf

location

Object
city

Ingrid_Bergman

marriedTo

Roberto_Rossellini

marriedTo

Ingrid_Bergman

Rome

isA

city

WordNet

+ 

MusicBrainz

YAGO2: 120 million facts on 10 million entities

Subject
Rome

city

Roberto_Rossellini

Predicate
isA

subclassOf

location

Object
city

Ingrid_Bergman

marriedTo

Roberto_Rossellini

marriedTo

Ingrid_Bergman

Rome

isA

city
QA, meet the (semantic) Web of Data

WordNet +

LOD: > 31 BILLION triples

YAGO2: 120 million facts on 10 million entities

Subject Rome city Roberto_Rossellini
Predicate isA subclassOf marriedTo
Object city location Ingrid_Bergman
<table>
<thead>
<tr>
<th><strong>U.S. Cities</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>ITS LARGEST AIRPORT WAS NAMED FOR A WORLD WAR II HERO; ITS SECOND LARGEST, FOR A WORLD WAR II BATTLE</td>
</tr>
</tbody>
</table>

**What is Toronto??**
“Who played in Casablanca and was married to a writer born in Rome?”
“Who played in Casablanca and was married to a writer born in Rome?”

```sparql
SELECT ?p WHERE{
  ?p type person .
  ?p actedIn Casablanca_(film) .
  ?p isMarriedTo ?w .
  ?w type writer .
  ?w bornIn Rome }
```
“Who played in *Casablanca* and was married to a writer born in Rome?”

```
SELECT ?p WHERE{
  ?p type person .
  ?p actedIn Casablanca_(film).
  ?p isMarriedTo ?w .
  ?w type writer .
  ?w bornIn Rome }
```
“Who played in *Casablanca* and was married to a writer born in Rome?”

```
SELECT ?p WHERE{
  ?p type person .
  ?p actedIn Casablanca_(film) .
  ?p isMarriedTo ?w .
  ?w type writer .
  ?w bornIn Rome }
```

Dot: conjunction
“Who played in *Casablanca* and was married to a writer born in Rome?”

```
SELECT ?p WHERE{
  ?p type person .
  ?p actedIn Casablanca_(film) .
  ?p isMarriedTo ?w .
  ?w type writer .
  ?w bornIn Rome }
```

- **Dot:** conjunction
- **?p, ?w:** variables to be bound
CRASH COURSE: QUERYING THE WEB OF DATA

"Who played in Casablanca and was married to a writer born in Rome?"

```
SELECT ?p WHERE{
  ?p type person .
  ?p actedIn Casablanca_(film).  
  ?p isMarriedTo ?w .
  ?w type writer .
  ?w bornIn Rome }
```

Dot: conjunction
?p, ?w: variables to be bound
Same variable → same binding (join)

EMNLP
July 12, 2012
Natural Language Questions for the Web of Data - Yahya et al.
“Who played in *Casablanca* and was married to a writer born in Rome?”

```
SELECT ?p WHERE{
?p type person .
?p actedIn Casablanca_(film). 
?p isMarriedTo ?w .
?w type writer .
?w bornIn Rome }
```

- Dot: conjunction
- ?p, ?w: variables to be bound
- Same variable $\rightarrow$ same binding (join)
What is DEANNA?

Question

DEANNA

SPARQL

KB

Answers
What is DEANNA?

Question

“Who played in Casablanca and was married to a writer born in Rome?”

DEANNA

SPARQL

KB

Answers
What is DEANNA?

Question: “Who played in Casablanca and was married to a writer born in Rome?”

DEANNA

SPARQL

KB

Answers

?p type person.
?p actedIn Casablanca_(film).
?p isMarriedTo ?w.
?w type writer.
?w bornIn Rome.
What is DEANNA?

Question: “Who played in Casablanca and was married to a writer born in Rome?”

DEANNA

SPARQL

KB

Answers

?p type person.
?p actedIn Casablanca_(film).
?p isMarriedTo ?w.
?w type writer.
?w bornIn Rome.
What is DEANNA?

“Who played in Casablanca and was married to a writer born in Rome?”

DEANNA

SPARQL

KB

Answers

?p type person.
?p actedIn Casablanca_(film).
?p isMarriedTo ?w.
?w type writer.
?w bornIn Rome.
Inside DEANNA

Phrase detection

Phrase mapping

Dependency detection

Joint Disambig.

Query Generation

Question

DEANNA

SPARQL

KB

Answers
Inside DEANNA

- Rome
- born
- was born
- a writer
- Casablanca
- played
- played in
- Who
- married
- married to
- was married to

Question

DEANNA

SPARQL

KB

Answers

Phrase detection

Phrase mapping

Dependency detection

Joint Disambig.

Query Generation

Phrase detection

Phrase mapping

 Dependency detection

Query Generation

Joint Disambig.
Inside DEANNA

Phrase detection
Phrase mapping
Dependency detection
Joint Disambig.
Query Generation

Question
DEANNA
SPARQL
KB
Answers

Phrase detection
Phrase mapping
Dependency detection
Joint Disambig.
Query Generation

Rome
born
was born
a writer
Casablanca
played
played in
Who
married
married to
was married to

e:Rome
e:Sydne_Rome
e:Born_(film)
e:Max_Born
r:bornOnDate
r:bornInPlace
c:writer
e:White_House
e:Casablanca
e:Casablanca_(film)
e:Played_(film)
r:actedIn
r:hasMusicalRole
c:person
e:Married_(series)
c:married_person
r:isMarriedTo
Inside DEANNA

Phrase detection
Phrase mapping
Dependency detection
Joint Disambig.
Query Generation

Question
DEANNA
SPARQL
KB
Answers

Phrase detection
Phrase mapping
Dependency detection
Joint Disambig.
Query Generation

DEANNA
SPARQL
KB
Answers
Inside DEANNA

Question

DEANNA

SPARQL

KB

Answers

Phrase detection

Phrase mapping

Dependency detection

Joint Disambig.

Query Generation

Phrase detection

Phrase mapping

Dependency detection

Joint Disambig.

Query Generation
Inside DEANNA

Phrase detection

Phrase mapping

Dependency detection

Joint Disambig.

Query Generation

Question

DEANNA

SPARQL

KB

Answers
Structured Query Generation

SELECT ?p WHERE {
  ?p type person.
  ?p actedIn Casablanca_(film).
  ?p isMarriedTo ?w.
  ?w type writer.
  ?w bornIn Rome
}
Outline

1. Phrase detection
2. Phrase mapping
3. Dependency detection
5. Query Generation

Experiments & Results

Disambiguation graph construction

Disambiguation graph processing

Question

DEANNA

SPARQL

KB

Natural Language Questions for the Web of Data - Yahya et al.
Outline

1. Phrase detection
2. Phrase mapping
3. Dependency detection
5. Query Generation

Experiments & Results
Phrase Detection

- **Concepts**: entities & classes: Dictionary-based

<table>
<thead>
<tr>
<th>Concept</th>
<th>Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Casablanca</td>
<td>Casablanca</td>
</tr>
<tr>
<td>Casablanca</td>
<td>Casablanca, Morocco</td>
</tr>
<tr>
<td>Casablanca_(film)</td>
<td>Casablanca the film</td>
</tr>
<tr>
<td>Casablanca_(film)</td>
<td>Casablanca</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- **Relations**: Mainly use Reverb [Fader et al. EMNLP’11]: $V$ | $VP$ | $VW*P$

... was/VBD married/VBN to/TO a/DT...
**Concepts:** entities & classes:

- Dictionary-based

<table>
<thead>
<tr>
<th>Concept</th>
<th>Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Casablanca</td>
<td>Casablanca</td>
</tr>
<tr>
<td>Casablanca</td>
<td>Casablanca, Morocco</td>
</tr>
<tr>
<td>Casablanca_(film)</td>
<td>Casablanca the film</td>
</tr>
<tr>
<td>Casablanca_(film)</td>
<td>Casablanca</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Relations:**

- Mainly use Reverb [Fader et al. EMNLP’11]: $V \mid VP \mid VW*P$
- ... was/VBD married/VBN to/TO a/DT
Phrase Detection

- **Concepts:** entities & classes:
  Dictionary-based

<table>
<thead>
<tr>
<th>Concept</th>
<th>Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Casablanca</td>
<td>Casablanca</td>
</tr>
<tr>
<td>Casablanca</td>
<td>Casablanca, Morocco</td>
</tr>
<tr>
<td>Casablanca_(film)</td>
<td>Casablanca the film</td>
</tr>
<tr>
<td>Casablanca_(film)</td>
<td>Casablanca</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- **Relations:**
  Mainly use **Reverb** [Fader et al. EMNLP’11]: V | VP | VW*P
  ...
  **was/VBD married/VBN to/TO a/DT**...
Outline

1. Phrase detection
2. Phrase mapping
3. Dependency detection
5. Query Generation

DEANNA
SPARQL
KB

Experiments & Results
Outline

1. Phrase detection
2. Phrase mapping
3. Dependency detection
5. Query Generation

Experiments & Results
### Phrase Mapping

**Concepts**: entities & classes: Dictionary-based

<table>
<thead>
<tr>
<th>Concept</th>
<th>Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Casablanca</td>
<td>Casablanca</td>
</tr>
<tr>
<td>Casablanca</td>
<td>Casablanca, Morocco</td>
</tr>
<tr>
<td>Casablanca_(film)</td>
<td>Casablanca the film</td>
</tr>
<tr>
<td>Casablanca_(film)</td>
<td>Casablanca</td>
</tr>
<tr>
<td>Played_(film)</td>
<td>Played</td>
</tr>
</tbody>
</table>
Phrase Mapping

Concepts: entities & classes: Dictionary-based

<table>
<thead>
<tr>
<th>Concept</th>
<th>Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>e:White_House</td>
<td>Casablanca</td>
</tr>
<tr>
<td>e:Casablanca</td>
<td>Casablanca, Morocco</td>
</tr>
<tr>
<td>e:Casablanca_(film)</td>
<td>Casablanca the film</td>
</tr>
<tr>
<td>e:Played_(film)</td>
<td>Played</td>
</tr>
<tr>
<td>Casablanca</td>
<td>Casablanca</td>
</tr>
<tr>
<td>Casablanca_(film)</td>
<td>Casablanca</td>
</tr>
</tbody>
</table>
Phrase Mapping

Concepts: entities & classes: Dictionary-based

<table>
<thead>
<tr>
<th>Concept</th>
<th>Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Casablanca</td>
<td>Casablanca</td>
</tr>
<tr>
<td>Casablanca</td>
<td>Casablanca, Morocco</td>
</tr>
<tr>
<td>Casablanca_(film)</td>
<td>Casablanca the film</td>
</tr>
<tr>
<td>Casablanca_(film)</td>
<td>Casablanca</td>
</tr>
<tr>
<td>Played_(film)</td>
<td>Played</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relation</th>
<th>Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>actedIn</td>
<td>acted in</td>
</tr>
<tr>
<td>actedIn</td>
<td>played in</td>
</tr>
<tr>
<td>hasMusicalRole</td>
<td>plays</td>
</tr>
<tr>
<td>hasMusicalRole</td>
<td>mastered</td>
</tr>
</tbody>
</table>
Phrase Mapping

Concepts: entities & classes:
Dictionary-based

Concept | Phrase
--- | ---
Casablanca | Casablanca
Casablanca | Casablanca, Morocco
Casablanca_(film) | Casablanca the film
Casablanca_(film) | Casablanca
Played_(film) | Played

Relation | Phrase
--- | ---
actedIn | acted in
actedIn | played in
hasMusicalRole | plays
hasMusicalRole | mastered
Outline

1. Phrase detection
2. Phrase mapping
3. Dependency detection
4. *Joint Disambig.*
5. Query Generation
6. Experiments & Results

Question

- DEANNA
  - SPARQL
  - KB
Outline

1. Phrase detection
2. Phrase mapping
3. Dependency detection
5. Query Generation

Experiments & Results

Question

DEANNA

SPARQL

KB
Look for specific patterns in dependency parses [de Marneffe et al. LREC’06]
Look for specific patterns in dependency parses [de Marneffe et al. LREC’06]

- Dependency Detection

```
<dependency>
  <dt>Rome</dt>
  <dd>e:Rome</dd>
  <dd>e:Sydne_Rome</dd>
  <dependency>
    <dt>born</dt>
    <dd>e:Born_(film)</dd>
    <dd>e:Max_Born</dd>
    <dependency>
      <dt>was born</dt>
      <dd>r:bornOnDate</dd>
      <dd>r:bornInPlace</dd>
      <dependency>
        <dt>a writer</dt>
        <dd>c:writer</dd>
      </dependency>
    </dependency>
  </dependency>
</dependency>
```
Look for specific patterns in dependency parses [de Marneffe et al. LREC’06]
Look for specific patterns in dependency parses
[de Marneffe et al. LREC’06]
RESULT
Disambiguation Graph

Phrases:
- Rome
- born
- was born
- a writer
- Casablanca
- played
- played in
- Who
- married
- married to
- was married to

Semantic Nodes:
- e:Rome
- e:Sydne_Rome
- e:Born_(film)
- e:Max_Born
- r:bornOnDate
- r:bornInPlace
- c:writer
- e:White_House
- e:Casablanca
- e:Casablanca_(film)
- e:Played_(film)
- r:actedIn
- r:hasMusicalRole
- c:person
- e:Married_(series)
- c: married_person
- r:isMarriedTo

Nodes:
- q1
- q2
- q3
Outline

1. Phrase detection
2. Phrase mapping
3. Dependency detection
5. Query Generation

DEANNA

SPARQL

KB

Experiments & Results
Outline

1. Phrase detection
2. Phrase mapping
3. Dependency detection
5. Query Generation

Experiments & Results

Question

DEANNA

SPARQL

KB

Natural Language Questions for the Web of Data - Yahya et al.
Outline

1. Phrase detection
2. Phrase mapping
3. Dependency detection
5. Query Generation

Experiments & Results
Joint Disambiguation - ILP

- **ILP**: Integer Linear Programming

  \[
  \text{maximize } \alpha \sum_{i,j} w_{i,j} Y_{i,j} + \beta \sum_{k,l} v_{k,l} Z_{k,l} + \ldots
  \]

- **Subject to:**
  - No token in multiple phrases,
  - Triples observe type constraints, …
Joint Disambiguation – Objective

\[ \alpha \sum_{i,j} w_{i,j} Y_{i,j} + \beta \sum_{k,l} v_{k,l} Z_{k,l} \]

Prior
Joint Disambiguation – Objective

\[ \alpha \sum_{i,j} w_{i,j} Y_{i,j} + \beta \sum_{k,l} v_{k,l} Z_{k,l} \]

Semantic nodes
- e:Rome
- e:Sydne_Rome
- e:Born_(film)
- e:Max_Born
- r:bornOnDate
- r:bornInPlace
- c:writer

Phrase nodes
- Rome
- born
- was born
- a writer

Similarity Edges

Coherence Edges

Coherence

Coherence

q-nodes

q_1

Natural Language Questions for the Web of Data - Yahya et al.
A phrase node can be assigned to one semantic node:

\[ \alpha \sum_{i,j} w_{i,j} Y_{i,j} + \beta \sum_{k,l} v_{k,l} Z_{k,l} \]
A phrase node can be assigned to one semantic node:

\[ \alpha \sum_{i,j} w_{i,j} Y_{i,j} + \beta \sum_{k,l} v_{k,l} Z_{k,l} \]
Classes translate to type-constrained variables

→ Every semantic triple should have a class to join & project!

person actedIn casablanca_(film)

?x type person . ?x actedIn casablanca_(film)
Classes translate to \texttt{type-constrained variables}.

\[ \text{Every semantic triple should have a class to join \& project!} \]

\[ \text{person} \text{ actedIn} \text{ Casablanca\_(film)} \]

\[ ?x \text{ type person} \quad \text{.} \quad ?x \text{ actedIn} \text{ Casablanca\_(film)} \]
Joint Disambiguation – Constraints

Classes translate to type-constrained variables

→ Every semantic triple should have a class to join & project!

```
?x type person . ?x actedIn Casablanca_(film)
```

Phrase nodes

Semantic nodes:
- e:Rome
- e:Sydne_Rome
- r:bornOnDate
- r:bornInPlace
- e:The_Writer (magazine)
- c:writer
Classes translate to **type-constrained variables**

→ Every semantic triple should have a class to **join & project**!

```
?x type person.
?x actedIn Casablanca_(film)
```

Phrase nodes

Semantic nodes

```
q_1 was born a writer
```

```
Rome e:Sydne_Rome e:Rome
r:bornOnDate r:bornInPlace
e:The_Writer (magazine)
c:writer
```
Joint Disambiguation – Constraints

Classes translate to type-constrained variables

→ Every semantic triple should have a class to join & project!

```
person actedIn casablanca_(film)

?p type person . ?p actedIn casablanca_(film)
```

Phrase nodes

Semantic nodes

- e:Rome
- e:Sydne_Rome
- r:bornOnDate
- r:bornInPlace
- e:The_Writer (magazine)
- c:writer
Outline

1. Phrase detection
2. Phrase mapping
3. Dependency detection
5. Query Generation

Question

DEANNA

SPARQL

KB

Experiments & Results
Outline

1. Phrase detection
2. Phrase mapping
3. Dependency detection
5. Query Generation

Experiments & Results

DEANNA
SPARQL
KB

EMNLP
July 12, 2012
Natural Language Questions for the Web of Data - Yahya et al.
Structured Query Generation

SELECT ?p WHERE {
  ?w type writer .
  ?w bornIn Rome .
  ?p type person.
  ?p actedIn Casablanca_(film).
  ?p isMarriedTo ?w }

EMNLP
July 12, 2012
Outline

1. Phrase detection
2. Phrase mapping
3. Dependency detection
5. Query Generation

Experiments & Results

Question

DEANNA

SPARQL

KB
Evaluation Methodology

- **3-stage** evaluation for more insight:
  1. Disambiguation
  2. Query generation
  3. Query answering

- We rely on **human judges**
Datasets

• **QALD-1**
  - YAGO2
  - 27/50 questions within scope
  - “Which software has been published by Mean Hamster Software?”

• **NAGA [Elbassuoni et al. CIKM’09]**
  - YAGO+IMDB
  - 44/87 questions within scope
  - “Which director has won the Academy Award for Best director and is married to an actress that has won the Academy Award for Best Actress?”
Results 1/3: Disambiguation

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>QALD-1</th>
<th>NAGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$cov_{macro}$</td>
<td>0.973</td>
<td>0.934</td>
</tr>
<tr>
<td>$prec_{macro}$</td>
<td>1.000</td>
<td>0.934</td>
</tr>
<tr>
<td>$cov_{micro}$</td>
<td>0.963</td>
<td>0.945</td>
</tr>
<tr>
<td>$prec_{micro}$</td>
<td>1.000</td>
<td>0.941</td>
</tr>
</tbody>
</table>

$\text{cov} = \frac{\text{correct/ideal}}{}$

$\text{prec} = \frac{\text{correct / retrieved}}{}$
### Issues:
- Incorrect disambiguation
- Incorrect dependencies.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>QALD-1</th>
<th>NAGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$cov_{macro}$</td>
<td>0.975</td>
<td>0.894</td>
</tr>
<tr>
<td>$prec_{macro}$</td>
<td>1.000</td>
<td>0.941</td>
</tr>
<tr>
<td>$cov_{micro}$</td>
<td>0.963</td>
<td>0.847</td>
</tr>
<tr>
<td>$prec_{micro}$</td>
<td>1.000</td>
<td>0.906</td>
</tr>
</tbody>
</table>

$cov = \frac{\text{#correct triples}}{\text{#ideal triples}}$

$prec = \frac{\text{#correct triples}}{\text{#retrieved triples}}$
## Results 3/3: Answering

### Benchmark Results

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>QALD-1</th>
<th>NAGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>#questions</td>
<td>27</td>
<td>44</td>
</tr>
<tr>
<td>#queries</td>
<td>20</td>
<td>41</td>
</tr>
<tr>
<td>#satisfactory</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>#relaxed</td>
<td>+3</td>
<td>+3</td>
</tr>
</tbody>
</table>
### Results 3/3: Answering

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>QALD-1</th>
<th>NAGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>#questions</td>
<td>27</td>
<td>44</td>
</tr>
<tr>
<td>#queries</td>
<td>20</td>
<td>41</td>
</tr>
<tr>
<td>#satisfactory</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>#relaxed</td>
<td>+3</td>
<td>+3</td>
</tr>
</tbody>
</table>

**unsatisfactory**: main problem is **empty result set**
Relaxation: Keep type constraints, everything else to keywords

\[ \text{?x type writer . ?x bornIn Rome} \]

\[ \text{?x type writer . ?x bornIn ?y [“Rome”]} \]

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>QALD-1</th>
<th>NAGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>#questions</td>
<td>27</td>
<td>44</td>
</tr>
<tr>
<td>#queries</td>
<td>20</td>
<td>41</td>
</tr>
<tr>
<td>#satisfactory</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>#relaxed</td>
<td>+3</td>
<td>+3</td>
</tr>
</tbody>
</table>

unsatisfactory: main problem is empty result set
Wrap-up

Question

DEANNA

SPARQL

KB

Answers
Wrap-up

Question: “Who played in Casablanca and was married to a writer born in Rome?”

DEANNA

SPARQL

KB

Answers

Natural Language Questions for the Web of Data - Yahya et al.
Wrap-up

Question

“Who played in Casablanca and was married to a writer born in Rome?”

DEANNA

SPARQL

KB

Answers
Who played in Casablanca and was married to a writer born in Rome?

?p type person.
?p actedIn Casablanca_(film).
?p isMarriedTo ?w.
?w type writer.
?w bornIn Rome
“Who played in Casablanca and was married to a writer born in Rome?”

?p type person.
?p actedIn Casablanca_(film).
?p isMarriedTo ?w.
?w type writer.
?w bornIn Rome
“Who played in Casablanca and was married to a writer born in Rome?”

?-p type person.
?-p actedIn Casablanca_(film).
?-p isMarriedTo ?w.
?w type writer.
?w bornIn Rome.
Thank you.

QUESTIONS/COMMENTS?

bit.ly/mpi-deanna