Question Answering Systems

Named entity recognition and disambiguation

Rishiraj Saha Roy
Max Planck Institute for Informatics, Germany
Lecture overview

- Logistics
- Named entity basics
- NERD System 1: TAGME
- NERD System 2: AIDA
Logistics: Exam

- All 21 papers → Syllabus
- ~15 minutes per student
- ~3-5 topics → questions content covered in class
- Question test your understanding
- Relative grading → Difficult
Logistics: Assignments

- **6 ECTS**
  - Self-assessment, no tutorials
  - Grades available every week
  - Sample good reviews (try)
  - Extensive guidelines (subtle factors)
- Assignments contribute to final grade
- Exact formula only at end
- Our decision is final, vetted by two
Logistics: Material

- Reading material
  - https://www.mpi-inf.mpg.de/question-answering-systems/

- Slides and recordings
  - https://drive.google.com/drive/folders/1Z0ljVSjymCD6IX_TCN4dNlnuz8Kb42BL

- Assignment grades
  - https://docs.google.com/spreadsheets/d/e/2PACX-1vRN0qyr-ooE1jGLfLoPM89ipdPRBprwRUAKLkaRXPqCmDcj0Ht9T5LfqlqEe3gLk3sHS9YI
    HndJRLI/pubhtml?gid=1036917168&single=true&urp=gmail_link
Question of the day

How can we disambiguate named entities present in questions?
You’ll find this covered in

1. Fast and Accurate Annotation of Short Texts with Wikipedia Pages
   - Ferragina and Scaiella
   - CIKM 2010 + IEEE Software 2011

2. Robust Disambiguation of Named Entities in Text
   - Hoffart et al.
   - EMNLP 2011
Entities: Basics

- Entities and relationships
- Entities and named entities
  - People
  - Organizations
  - Locations
  - More (ever-expanding: movies, food, animals...)
- Concepts and classes
- Here: What is there in Wikidata/Wikipedia 😊
Entities: Tasks

- Named entity recognition: NER in test
- Named entity disambiguation / linking: NED/NER
- Named entity typing: NET
- Main ideas: Similarity and coherence

Rishiraj Saha Roy

26 May 2020
Entities: Applications

1. Question understanding
2. News readability
3. Information extraction
   - Many many more in IR + NLP
     - Can you think of some..?
Fast and Accurate Annotation of Short Texts with Wikipedia Pages

Fast and accurate annotation of short texts with wikipedia pages
P Ferragina, U Scaiella
IEEE software 29 (1), 70-75
Try it out!

TAGME is a powerful tool that is able to identify on-the-fly meaningful short-phrases (called “spots”) in an unstructured text and link them to a pertinent Wikipedia page in a fast and effective way. This annotation process has implications which go far beyond the enrichment of the text with explanatory links because it concerns with the contextualization and, in some way, the understanding of the text.

Try TAGME now!

You can play with the demo interface below or check the TAGME RESTful API we are currently supporting.

Currently TAGME is available in English, German and in Italian and it is based on Wikipedia snapshots of April, 2016.

NEWS! TAGME is now hosted by the D4Science infrastructure. Check the RESTful API page for details.

Developed by Paolo Ferragina and Ugo Scialetta at A² Lab
Dipartimento di Informatica, University of Pisa.

On this day 24 years ago Maradona scored his infamous "Hand of God" goal against England in the quarter-final of the 1986 World Cup.

https://tagme.d4science.org/tagme/

Thanks to Prof. Paolo Ferragina for the slides.
On this day 24 years ago Maradona scored his infamous "Hand of God" goal against England in the quarter-final of the 1986 World Cup.
One issue: Synonymy

He is using Microsoft’s browser

He is a fan of Internet Explorer
Another issue: Polysemy

the paparazzi photographed the star

the astronomer photographed the star
Wikipedia is a rich source of instances

Steve Jobs

From Wikipedia, the free encyclopedia

For the biography, see Steve Jobs (book).

Steven Paul "Steve" Jobs (/ˈstɛbə/; February 24, 1955 – October 5, 2011) was an Arab-American entrepreneur and inventor, who was the co-founder, chairman, and CEO of Apple Inc. Through Apple, he was widely recognized as a charismatic pioneer of the personal computer revolution, and for his influential career in the computer and consumer electronics fields, transforming "one industry after another, from computers and smartphones to music and movies." Jobs also co-founded and served as chief executive of Pixar Animation Studios; he became a member of the board of directors of The Walt Disney Company in 2006, when Disney acquired Pixar. Jobs was among the first to see the commercial potential of a mouse-driven graphical user interface, which led to the creation of the Apple Lisa and, a year later, the Macintosh. He also played a role in introducing the LaserWriter, one of the first widely available laser printers, to the market.

After a power struggle with the board of directors in 1985, Jobs left Apple and founded NeXT, a computer platform development company specializing in the higher-education and business markets. In 1986, he acquired the computer graphics division of Lucasfilm, which was spun off as Pixar. He was credited in Toy Story (1995) as an executive producer. He served as CEO and majority shareholder until Disney's purchase of Pixar in 2006. In 1996, after Apple had failed to deliver its operating system, Copland, Gil Amelio turned to NeXT Computer, and the NeXTSTEP platform became the foundation for the Mac OS X. Jobs returned to Apple as an advisor, and took control of the company as an interim CEO. Jobs brought Apple from near bankruptcy to profitability by 1998.
Wikipedia’s categories contain classes

Categories typically form a taxonomic DAG
DAG of categories
Topic-based annotation

"Diego Maradona won against Mexico"

Mexico’s football team

Ex-Argentina’s player

Find anchors and annotate them with articles drawn from Wikipedia!
He is using Microsoft’s browser
She plays with Internet Explorer
Polysemy

the paparazzi photographed the **star**

the astronomer photographed the **star**

**Celebrity** is a person who is famously recognized ...

**Star** is a massive, luminous ball of plasma ...
Why is it a difficult problem?

“Diego Maradona won against Mexico”

- Ex-Argentina’s coach
- His nephew
- Maradona Stadium
- Maradona Movie
- ...

- Mexico nation
- Mexico state
- Mexico football team
- Mexico baseball team
- ...

Don’t annotate!
The literature

- TagMe (Univ. Pisa)
- DBPedia Spotlight (Univ. Berlin)
- Illinois Wikifier (Univ. Illinois)
- AIDA (Max Planck Institute for Informatics)
- CNMS (Univ. Amsterdam)
- Wikipedia Miner (Univ. Waikato)

Many commercial software: AlchemyAPI, DBpedia Spotlight, Extractiv, Lupedia, OpenCalais, Saplo, SemiTags, TextRazor, Wikimeta, Yahoo! Content Analysis, Zemanta.
The TAGME system

- Designed for **short texts**
  - news, blogs, search-results snippets, tweets, ads, etc
  - competitive on long texts too

- Achieves **high accuracy**
  - Massive experimental test on millions of short texts

- **Fast**
  - More than 10x faster than others
  - 100% Java

---

*Good research problem*
- Everything done??
- Look deeper!
- Problem + quality of results
- Speed? Memory?

*often unaided paper*
TAGME: Distilled information

- Anchor dictionary
- Page catalog
- In-link graph
TAGME: Overview

- Anchor parsing
- Anchor disambiguation
- Anchor pruning

TagME is $\text{NERD} \rightarrow \text{NER} + \text{NED}$

\[
\begin{array}{cccc}
  w_1 & w_2 & w_3 & w_4 \\
\end{array}
\]

nested strings

\[ a_1 \in a_2 \]

overlapping mentions?

\[ \text{Tom cruise ship scene} \]

Trojan War

\[ p(a_1) > p(a_2) \Rightarrow \text{link}(a_1) \Rightarrow \text{link}(a_2) \]

Exception: the act vs. act
Features used to link a $\rightarrow$ p

**Commonness of a page p wrt an anchor a**

$$\Pr(p | a) = \frac{\# a \text{ linked to } p}{\# a \text{ as anchor}}$$

**Context of a around the mention**

$$T = \ldots w_1 w_2 w_3 a w_4 w_5 w_6 \ldots$$

and the content of a *page/entity*

Page $p = z_1 z_2 z_3 z_4 z_5 z_6 \ldots$

**Link probability of an anchor a**

$$lp(a) = \frac{\text{freq of a as anchor}}{\text{freq of a in the text}}$$

**Graph-based features**

- a is a mention-node
- p is an entity-node
- Links a $\rightarrow$ p and paths between pages
Relatedness between pages

\[ \text{rel}(P_a, P_b) = \frac{\log(\max(|A|,|B|)) - \log(|A \cap B|)}{\log(|W|) - \log(\min(|A|,|B|))} \]

\[ sr(a,b) = \log \left( \frac{\max(|A|,|B|)}{|W|} \right)^6 \]
How TAGME works

“Diego Maradona won against Mexico”

- Diego A. Maradona
- Diego Maradona Jr.
- Maradona Stadium
- Maradona Film
- ...

- Mexico nation
- Mexico state
- Mexico football team
- Mexico baseball team
- ...

No Annotation

PARSING

DISAMBIGUATION by a voting scheme

PRUNING 2 simple features
Disambiguation: The Voting Scheme

Collective agreement among topics via voting

\[ vote_b(p_a) = \frac{\sum_{p_b \in P_b(b)} rel(p_b, p_a) \cdot Pr(p_b|b)}{|P_b(b)|} \]

\[ rel_a(p_a) = \sum_{b \in A \setminus \{a\}} rel(p_a, p_b) \]

Wiki-articles

- Mexico
- State of Mexico
- **Mexico National Football Team**
- Mexico

26 May 2020
Disambiguation: All steps

\[ \tau = \text{trade-off speed vs. recall} \]

Pruning by commonness < $\tau$

Voting scheme

Select top-$\varepsilon$ pages

Select the best in commonness

Commonness of a page $p$ wrt an anchor $a$

\[ \Pr(p \mid a) = \frac{\# a \text{ linked to } p}{\# a \text{ as anchor}} \]
Pruning

- Use 2 features:
  1. link probability
  2. coherence wrt context

- Compute a $\rho$ score via
  - 3 classifiers, avg, linear combination

$\text{if } \rho < \rho_{NA} \text{ then prune!}$
Robust Disambiguation of Named Entities in Text

Robust disambiguation of named entities in text
J Hoffart, MA Yosef, I Bordino, H Fürstenau, M Pinkal, M Spaniol, ...
Proceedings of the Conference on Empirical Methods in Natural Language ...

Aida: An online tool for accurate disambiguation of named entities in text and tables
MA Yosef, J Hoffart, I Bordino, M Spaniol, G Weikum
Proceedings of the VLDB Endowment 4 (12), 1450-1453
Disambiguating to KG entities

- TAGME is fast and effective
- Works well for short texts
- Does not go all the way!
- Wikipedia more general, but we need KG-linking!
- Lookup KG entities using Wikipedia links?
- Harness KG properties! Enter AIDA.
Try it out!

AmbiverseNLU Demo

Jack founded Alibaba in Hangzhou with investments from SoftBank and Goldman.

Enter any text in English, German, Spanish, or Chinese.

ANALYZE

concept

https://ambiversenlu.mpi-inf.mpg.de/
When Page played Kashmir at Knebworth, his Les Paul was uniquely tuned.
When Page played "Kashmir" at Knebworth, his Les Paul was uniquely tuned.
Entities in knowledge bases

Classes

- **song**
  - type
  - 1975 in
  - created
  - was played at

- **musician**
  - type
  - happens in

- **artifact**
  - type

- **guitar**
  - type

Entities

- **Classes**: C
- **Entities**: E

- **Yago select knowledge**
- **Question Answering Systems**
- **Saarland University, Summer Semester 2020**
- **26 May 2020**
AIDA features for disambiguation

When Page played *Kashmir* at Knebworth, his Les Paul was uniquely tuned.

- India
- Pakistan
- Pashmina

**Prior**
- Led Zeppelin
- Jimmy Page
- Knebworth Festival
- ...
  - 91%

**Context**
- 0.0

**Coherence**
  - India
  - Pakistan
  - 0.0

Images taken from Wikipedia under CC BY-SA 3.0

How often did “Kashmir” link to this entity in Wikipedia? Are the disambiguated entities related?
Kashmir (song)
From Wikipedia, the free encyclopedia

"Kashmir" is a song by the English rock band Led Zeppelin from their sixth album Physical Graffiti, released in 1975. It was written by Jimmy Page and Robert Plant (with contributions from John Bonham) in the first years, with the group acting since 1973.

References

They performed **Kashmir**, written by **Page** and **Plant**. **Page** played unusual chords on his **Gibson**.

**Kashmir**
- **Kashmir** (song)
- **Kashmir** (region)

**Page**
- Larry Page
- Jimmy Page
- Page, Arizona

**Plant**
- Robert Plant

**Gibson**
- Gibson Les Paul
- Gibson, Missouri

**Led Zeppelin**
- Hard rock
- Electric guitar

**Session guitarist**
- Led Zeppelin
- Gibson

**Jimmy Page**
- Signature model
- Hard rock
When Page played Kashmir at Knebworth, his Les Paul was uniquely tuned.

Goal: Dense subgraph with one mention-entity edge per mention
When Page played Kashmir at Knebworth, his Les Paul was uniquely tuned.

**Objective**: Maximize the minimum weighted degree

**Constraint**: Keep at least one entity per mention
Keyphrase-based similarity

- Keyphrases ($kp$) commonly occur only partially

\[
\text{score}(kp) = \frac{\# \text{ matching words}}{\text{length of cover}(kp)} + \frac{\text{weight}(w)}{\text{sum weight}(w)}
\]

- To score an entity, all keyphrase scores are summed

Kashmir was written by Page and Plant.

“Songs written by Robert Plant”

Account for partial matches

Weight of contained tokens $w$
Keyword weighting

\[
\text{score}(kp) = \frac{\# \text{ matching words}}{\text{length of cover}(kp)} \left( \frac{\sum_{w \in \text{cover}} \text{weight}(w)}{\sum_{w \in kp} \text{weight}(w)} \right)^2
\]

- Global IDF of a keyphrase token \( w \) in Wikipedia
- Mutual Information of a token \( w \) and an associated entity
  - How often does the token occur in the keyphrase set of an entity?

https://en.wikipedia.org/wiki/Mutual_information
Disambiguation by joint inference

**Input**
- Mentions
  - context of mention $cxt(m)$
  - entity candidates $e + cxt(e)$
- Features
  - Prior $prior(m,e)$
  - Similarity $sim(cxt(m), cxt(e))$
  - Coherence $coh(e1,e2)$

**Goal**

$$\alpha \cdot \sum_{i=1}^{k} prior(m_i, e_j)$$
$$+ \beta \cdot \sum_{i=1}^{k} \text{sim}(cxt(m_i), cxt(e_j))$$
$$+ \gamma \cdot \text{coh}(e_{j1}, e_{j2}, \ldots e_{jk})$$
$$= \max!$$

$m_i \leftarrow e_{j_1}$ arg max (score)

26 May 2020
Greedy graph algorithm

- **Input**: weighted graph of mentions and entities
- **Output**: result graph with maximum density

- **Objective**: maximize the minimum weighted degree
- **Constraint**: keep at least one entity per mention

Key Idea

1. Prune entities that are too distant from all mentions
2. While an entity can be removed, remove the one with the lowest weighted degree
   - Keep graph with best minimum weighted degree
Final steps

- Find subgraph maximizing total edge weight
  - If graph is small enough, enumerate all potential mention-entity mappings
  - Otherwise do local search, randomly switching mention-entity mappings for a fixed number of times
Robustness issues

<table>
<thead>
<tr>
<th>Prior may be misleading</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given the <strong>prior probability</strong> for all entity candidates, only use prior when very good indicator for one single entity (&gt;90%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Coherence can get hooked to a wrong subgraph</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given <strong>prior probability</strong> and <strong>similarity distribution</strong> for all entity candidates. If they are reasonably similar, fix entity for mention before running the graph algorithm.</td>
</tr>
</tbody>
</table>

Dataset is available at [http://www.mpi-inf.mpg.de/yago-naga/aida](http://www.mpi-inf.mpg.de/yago-naga/aida)
Conclusions

- Understanding entities is vital to QA
- NERD is a vital (first) cog in the QA wheel
- Often used off-the-shelf
- Many, many innovative techniques
- Mainly based on priors, similarity and coherence
- Applications span beyond QA!