Entity Linking

Genevieve Gorrell and Johann Petrak
Session Overview

• A short introduction to entity linking
• YODIE – Entity Linking with GATE
• BioYODIE – Linking of Biomedical entities with GATE
Introduction to Entity Linking
What is Entity Linking

- Entity linking is the task of identifying all mentions in text of a specific entity from a database or ontology.
- Also referred to as entity disambiguation: there may be several different with the same name in the database.
- Link mentions to the concept in the KB that best matches the meaning in the given context.
- Do this efficiently for a KB with millions of concepts and with dozens or hundreds of concept candidates per mention.

KAGAWA will be allowed to rejoin Borussia Dortmund in January in a swap deal which would see defender @NSubotic4 join #MUFC http://tiny.cc/4t19ux
Why are we doing it?

- Rather than just annotate the words “Berlusconi” and “Берлускони” as a Person (NER), link it to a specific ontology instance (entity)
  - Differentiate between Silvio Berlusconi, Marina Berlusconi, etc.
  - Ontologies tell us that this particular Berlusconi is a Politician, which is a type of Person. He is based in Italy, which is part of the EU. He was a prime minister, etc. This is all helpful to disambiguate and link the mention in the text to the correct entity URI in the ontology
- Having identified the unique entity enables further tasks like relation detection, semantic search a.o.
Why is it hard?

Entity linking needs to handle:

- Name variations (entities are referred to in many different ways, including colloquial variants)
- Entity ambiguity (the same string can refer to more than one entity)
  Sometimes dozens or hundreds of possible alternatives!
- Missing entities – there is no target entity in the entity knowledge base/database
Data Sources for EL

• Entity Linking is based on a datasource/knowledge base to which to link (or several)

• Researchers have used Wikipedia (e.g. TAC KBP, WikipediaMiner), Linked Open Data (in particular DBpedia, YAGO, and Freebase) and the UMLS metathesaurus for medical concepts (Metamap, Bio-YODIE)

• Some datasources/knowledge bases have names in several languages (Wikipedia, UMLS), but coverage is often very different between languages.

• Ideally adapt one system to different languages, different domains
• The entity linking system can either return a matching entry from the target knowledge base or “NIL” to indicate there is no matching entry in the entity database.

• Some entity linking systems make the closed world assumption (CWA) that there is always a target entity in the database.

• Often still focused on entities of type PER, LOC, ORG and often focused on English documents.
• LOD = Linked Open Data: 1000s of sources available
• Dbpedia is probably the most prominent: derived from Wikipedia
• Machine readable knowledge on 4.85 million entities and topics (as of Jun 2015), including:
  – 735,000 places/locations,
  – 1,445,000 persons
  – 241,000 organisations
• For each entity there are:
  – Entity name variants (e.g. IBM, Int. Business Machines)
  – a textual abstract
  – reference(s) to corresponding Wikipedia page(s)
  – entity-specific properties (e.g. death-date of a person)
The Thames Barrier is the world's second-largest movable flood barrier and is located downstream of central London, United Kingdom. Its purpose is to prevent London from being flooded by exceptionally high tides and storm surges moving up from the sea. It needs to be raised (closed) only during high tide; at ebb tide it can be lowered to release the water that backs up behind it.

Links to GeoNames And Freebase

Latitude & Longitude
LOD Sources – UMLS Metathesaurus

- Unified Medical Language System (from the US National Library of Medicine)
- Contains over two million biomedical and health-related concepts
- Combines many thesauri (“source vocabularies”)
- Much ambiguity! E.g. “unknown” is associated with 39 concepts
- The NLP “view” (subset) is more pragmatic, excluding unhelpfully ambiguous and spurious concepts
- “Cold” has six associated concepts
- “CUI” means concept unique identifier and is the unique code for the concept
A concept has a type (in this case “Disease or Syndrome”)

It has definitions from different vocabularies

It also has concept relations, i.e. things it is related to. “Common Cold” is related to “Respiration Disorders” for example
YODIE

Yet another Open Data Information Extraction System
YODIE Overview

- Link entities against DBpedia using DBpedia URIs
- Also adapt to other domains → BioYODIE
- Planned to make (parts) available as Open Source Software
- A typical approach
  - Entity Linking is a crowded marketplace! Lots of exciting new approaches
  - YODIE aims to provide robust choices as (eventually!) freely available, modifiable GATE components
  - Solid performance on varied input types, including tweets
  - Also a motivation and testbed for development of GATE components and plugins
YODIE Overview

• **WHAT** does YODIE do?

  - Identify potential entity mentions by matching known names of entities against text of documents/tweets/etc.
  - For each mention, get all entity candidates and information related to the entity, mention and entity/mention combination
  - Score entity candidates based on contextual fit, mention type, congruence with other potential entities etc.
  - Select candidate using a machine learning approach that learns to combine the scores to find the best entity
  - Try to identify mentions that do not refer to anything in Dbpedia (NILs)
Identify Mentions: How?

- Find KB concept labels: Gazetteer of known possible labels for all concepts
  - DBpedia instance labels
  - DBpedia name/nickname properties (from WP templates)
  - Labels from redirected WP pages (spelling variations)
  - Labels from WP Disambiguation pages
  - Anchor text from intra-WP links to that concept
- Additional mentions from supplemental gazetteers (e.g. country demonyms)
- Optional additional mentions candidates from NER
Identify Mentions: How?

• Match ALL labels ignoring case with document text
• Millions of labels, even after normalization/filtering
• Often many candidate concepts/URIs (100s) per label
• Will need a lot of information for each candidate concept:
  - URI
  - Original case
  - concept type (Org, Pers, …)
  - Frequency statistics
  - …
=> Cannot directly use a Gazetteer for all of this
=> Separate 1) finding of mention locations and
   2) getting all possible entity candidates for them
Identify Mentions: How?

• Use gazetteer to just identify mentions

• Prepare a database that maps each mention text to all the information we need.

• Prepare as much in advance as possible so we do not need to spend time on it in the pipeline

• Preparation only needed infrequently, may require a lot of computing resources

• Desired output:
  1) gazetteer of all labels/names
  2) all the information for each label/name:
      - information about the label itself (frequencies…)
      - information for all possible entities for the label
Candidate Preparation

- Sources (numbers just for EN):
  - DBpedia labels: ~10M triples
  - DBPedia properties: ~26M triples
  - WP page links: ~172M links
  - DBpedia: 1.6M triples

- Make sure URIs are normalized:
  - proper %-encoding style (varies between DBP version)
  - use IRIs not URIs everywhere

- Make sure labels are normalized
Normalize and filter labels:

- Exclude obvious cases (hundreds of chars, “List of ...”, numbers only, …)
- Normalize for case-insensitive matching
  => but remember original case!
- Canonical representation of umlauts, accents etc.
- Generate common variants, e.g. “ä” → “ae”
- Multiple white-space, punctuation
- Extract parentheses info, e.g
  “Jean Lemaire (painter)”
Label/Candidate Preparation

- Gather information per URI: class, frequency of related WP page link in WP articles, DBpedia properties …

- Gather information per label: original spelling, source (redirected page, disambiguation, canonical page, yago), frequencies, ….

- Gather information per label/URI pair: relative frequency of label used with this URI (WP page link), relative frequency of URI used with this label (“commonness”)

- Combine information and generate a de-normalized representation: key/value where the key is the label and the value is an array of rich URI-information, one element for each URI.
“yorkshire” →

- original_label=”Yorkshire”, all_labels=[“Yorkshire”, ”The Yorkshire Mafia”], uri=”dbp:The_Yorkshire_Mafia”, uriByLabel=0.0, sources=[“dbp_name”],....

- original_label=”Yorkshire”, all_labels=[“Yorkshire”, ”Yorkshire, Ohio”, “Yorkshire, OH”], uri=”dbp:Yorkshire,_Ohio”, sources=[“dbp_labels”], uriByLabel=3.310E-3, airpClass=dbpo:PopulatedPlace, ...

- original_label=”Yorkshire”, uri=”dbp:Yorkshire”, all_labels=[“Yorkshire”, ”Yorks”, ”County of Yorkshire”,...], uriByLabel=0.68, airpClass=dbpo:PopulatedPlace, parentheses =[“UK”, ”England”],...
• Use ExtendedGazetteer PR to just match the labels (~9M cleaned labels, 158M on disk, ~900M memory)
  Creates Mention Candidate/Spot annotation

• For each matched label, look up the candidates data from a database (~21G on disk)

• Each candidate from the list is represented as a separate annotation, the fields as features in the FeatureMap

• Mention annotations link to their candidate annotations (via a feature that contains a list of all their annotation ids) (AnnotationGraph plugin)

• Subsequently, most processing happens on entire candidate lists.
Reducing Candidates – Overall Strategy

• Matching all known labels heavily over-annotates (e.g. several entities for “the”) and generates lots of overlapping annotations

• Try to remove obvious spurious mentions early on: isolated stop words, inside URLs etc.

• Try to deal with overlaps early on: generally use longest match but with exceptions (determiner, possessives, locations)

• Calculate scores for remaining candidates

• Pick best candidate or decide it must be OOKB
Choosing a Candidate

- Have a reduced set of Mentions, each with its list of candidates
- Each candidate has initial features from the prepared data:
  - Label frequency $n(\text{label in WP})$
  - Commonness $p(\text{url|label})$
  - Link probability $p(\text{wikiLink|label})$
  - List of original labels
  - Dbpedia class (type) of candidate
  - Static Page Rank
  - ...
- Additional features based on how well the candidate “fits” into the actual context will get added...
Choosing a Candidate—What next?

- Measure how close each candidate is semantically to the text surrounding the mention: *text-based contextual similarity*: Vector Space Models, Word/Doc. Embeddings

- Measure how well each candidate fits with other candidates in the context, what is the best fit? → LOD-based structural similarity, relatedness, personalized page rank, ...: *coherence*

- Measure ambiguity, calculate relative scores

- Measure which kind of label matches mention and how (preferred, alternate, nickname, acronym, same case ...)

- Choose candidate based on all features/scores: → Machine Learning
Choosing a Candidate

• This can be presented as a machine learning classification problem
  − Each candidate becomes an instance
  − Scores etc. become features for the machine learning
  − Class is true if candidate should be selected for the mention, false otherwise (→ NILs)

• => ML task is to decide if a candidate is the right choice (“true”/1) or not (“false”/0) based on features

• Need to find a good training corpus

• Need to find a good trade-off: number of mentions, size of candidate lists vs. difficulty of the learning task (very unbalanced!)
Choosing a Candidate

• ML algorithms that have been found to work:
  - Support vector classification with RBF kernel (also need probability estimates!)
  - Random Forest

• Training data: TAC data, AIDA training corpus, training portion of the Sheffield NERD Tweet corpus (holding back other corpora for evaluation)

• Assigns true or false to each candidate, with a probability
  – This may mean that we have none, one or multiple “true”
  – Multiple “true”: use the probability assigned by the machine learner to select the best
  – None: remove / treat as NIL. Actual NIL (Entity not in KB) or just spurious mention
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Prec</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>YODIE 2015</td>
<td>0.62</td>
<td>0.65</td>
<td>0.64</td>
</tr>
<tr>
<td>AIDA/2013</td>
<td>0.74</td>
<td>0.34</td>
<td>0.47</td>
</tr>
<tr>
<td>AIDA/2014</td>
<td>0.70</td>
<td>0.74</td>
<td>0.72</td>
</tr>
<tr>
<td>Lupedia</td>
<td>0.50</td>
<td>0.24</td>
<td>0.32</td>
</tr>
<tr>
<td>Spotlight</td>
<td>0.31</td>
<td>0.40</td>
<td>0.35</td>
</tr>
<tr>
<td>TagMe</td>
<td>0.61</td>
<td>0.56</td>
<td>0.58</td>
</tr>
<tr>
<td>TextRazor</td>
<td>0.35</td>
<td>0.58</td>
<td>0.34</td>
</tr>
<tr>
<td>AGDISTIS</td>
<td>0.64</td>
<td>0.56</td>
<td>0.60</td>
</tr>
<tr>
<td>Zemanta</td>
<td>0.51</td>
<td>0.29</td>
<td>0.37</td>
</tr>
</tbody>
</table>
YODIE aims to be robust to varied types of input—recent research includes adding extra processing and context for tweets.

- **Resolve hash-tags:**
  - split multiword hash-tags
  - add hash-tag expansion candidates as text

- **Resolve user screen names:**
  - retrieve user profile information from Twitter and add as annotated text (location, description, language …)

- **Resolve linked web pages:**
  - retrieve content of web page and add as text
  - filter web page content to remove boilerplate/navigation
KAGAWA will be allowed to rejoin Borussia Dortmund in January in a swap deal which would see defender @NSubotic4 join #MUFCT http://tiny.cc/4t19ux

Greatest team in the world, with the best fans

A little known English soccer team, once shunned by Alan Shearer

Most successful team in England

No definition required ... if you can't work it out after five seconds you're either not remotely interested in football or you're not breathing.

The only team in England that lifted 19 league titles

FOOTBALL

Manchester United ready to swap Shinji Kagawa to sign Neven Subotic from Dortmund

MANCHESTER UNITED are lining up a sensational swap deal that would see Shinji Kagawa return to Borussia Dortmund in return for Serbian defender Neven Subotic.

Published: 17:27, Thu, Oct 3, 2013

By: Ben Jefferson

Shinji Kagawa could be on his way back to Dortmund [GETTY]

Japanese international Kagawa had a disappointing first season at United and has struggled to win a regular place under David Moyes this season, leading to speculation he may be open to the

• Additional context provides more text and entities for comparison
### Some Performance Figures on Tweets

<table>
<thead>
<tr>
<th>System</th>
<th>Prec</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>YODIE (Base)</td>
<td>0.44</td>
<td>0.55</td>
<td>0.49</td>
</tr>
<tr>
<td>YODIE (Exp)</td>
<td>0.50</td>
<td>0.62</td>
<td><strong>0.55</strong></td>
</tr>
<tr>
<td>Aida 2014</td>
<td><strong>0.59</strong></td>
<td>0.38</td>
<td>0.46</td>
</tr>
<tr>
<td>Lupedia</td>
<td>0.50</td>
<td>0.24</td>
<td>0.32</td>
</tr>
<tr>
<td>Spotlight</td>
<td>0.09</td>
<td>0.51</td>
<td>0.15</td>
</tr>
<tr>
<td>TagMe</td>
<td>0.10</td>
<td><strong>0.67</strong></td>
<td>0.17</td>
</tr>
<tr>
<td>TextRazor</td>
<td>0.19</td>
<td>0.44</td>
<td>0.26</td>
</tr>
<tr>
<td>Zemanta</td>
<td>0.48</td>
<td>0.56</td>
<td>0.52</td>
</tr>
</tbody>
</table>
BioYODIE
Why semantic annotation?

**Knowledge base (semantics)**
- Hyaluronan
- Protein (BEHAB)
- Brain
- Ventricle

**Knowledge base (annotations)**
- Hyaluronan binds BEHAB
- BEHAB isKindOf Brain
- BEHAB isPartOf Ventricle

**Text**
- BEHAB is expressed at high levels in ventricular tissue

**PMID:** 19535011
Find patient records that mention drugs used to treat disorders commonly associated with depression.
Named Entity Linking

- You have found mentions in text, but there are several possibilities for what something is referring to
- This is the disambiguation problem

Mr ZZZZZ complained of a cold, and was prescribed paracetamol OD.

- Low temperature
- Rhinovirus
- 4-Hydroxyacetanilide
- Oral Daily
- Doctor of Optometry
- Familial Osteochondritis Dissecans
Main approaches similar / identical to YODIE
- Dictionary based term identification based on names of UMLS entities of relevant types
- Retrieval of possible interpretations—any entity that can be called that
- Multiple disambiguation strategies gauging how well the candidates fit
- Scores combined to choose the best
• Unified Medical Language System (from the US National Library of Medicine)

• Contains over two million biomedical and health-related concepts

• Combines many thesauri ("source vocabularies")

• Much ambiguity! E.g. “unknown” is associated with 39 concepts

• The NLP “view” (subset) is more pragmatic, but excludes important acronyms
Mrs [REDACTED] was seen today and appears brighter in *mood* and to be *feeling* *positive*. She's continuing with *Sertraline* and self help.
Bio-YODIE disambiguation scoring strategies

- PageRank compiled across the UMLS co-occurrence table (compiled from MeSH co-occurrences in PubMed articles) gives a prior probability
- Prior probabilities of entities compiled from training data gives a valuable score
- Prior probabilities of an entity for a particular term is also valuable
- Contextual scores are slow and haven't been found effective in our work so far
- Scores are simply combined, as there is currently insufficient data for a machine learning approach to score combination
Comparison with other Biomedical NEL systems

<table>
<thead>
<tr>
<th>System</th>
<th>Time</th>
<th>Prec.</th>
<th>Recall</th>
<th>F1</th>
<th>Accuracy</th>
<th>Kappa</th>
</tr>
</thead>
<tbody>
<tr>
<td>MetaMap</td>
<td>2994</td>
<td>0.57</td>
<td>0.56</td>
<td>0.57</td>
<td>0.86</td>
<td>0.86</td>
</tr>
<tr>
<td>MetaMapLite</td>
<td>149</td>
<td>0.65</td>
<td>0.55</td>
<td>0.59</td>
<td>0.88</td>
<td>0.88</td>
</tr>
<tr>
<td>Bio-YODIE</td>
<td>263</td>
<td>0.62</td>
<td>0.61</td>
<td>0.62</td>
<td>0.89</td>
<td>0.89</td>
</tr>
</tbody>
</table>

Results obtained on MIMIC 2014 test corpus
Mimir index statistics

Using GCP, we (USFD and KCL) have indexed ...

- 15 million documents
- 4 million Trip documents

Proportion of types found on 50K test documents (patient records)
GATE Components for YODIE

- StringAnnotation plugin: ExtendedGazetteer
  https://github.com/johann-petrak/gateplugin-StringAnnotation

- JdbcLookup plugin
  https://github.com/johann-petrak/gateplugin-JdbcLookup

- ModularPipelines plugin
  https://github.com/johann-petrak/gateplugin-ModularPipelines
  - modularization
  - configuration

- LearningFramework plugin
  https://github.com/GateNLP/gateplugin-LearningFramework

- GATE ConText Wrapper (Harkema et.al. 2009)
  https://github.com/GateNLP/gateplugin-ConText

  (Negation, Experiencer, Temporal Status)
GATE Components for YODIE

• Java “Scripting” plugin
  https://github.com/johann-petrak/gateplugin-Java
  Fast within-GUI Java prototyping / programming

• AnnotationGraph plugin
  https://github.com/johann-petrak/gateplugin-AnnotationGraphs
  Represent relations, sequences, candidate lists as Annotations

• Evaluation plugin
  https://github.com/johann-petrak/gateplugin-Evaluation

• [several others ....]

• GATE GCP: process large corpora

• GATE Cloud: deploy service