Advanced Methods of Information Retrieval
- Information Extraction -

Dr. Elena Demidova
SS 2018
Recap: Knowledge Graphs and RDF
Data on the Web: knowledge graphs

IR search result

Brunswick stew Recipe | Jamie Deen | Food Network
https://www.foodnetwork.com/recipes/.../brunswick-stew-recipe-3383244
Rating: 4 - 15 reviews - 2 hrs 10 mins
Watch how to make this recipe. Melt the butter in a large Dutch oven over medium-high heat. Once melted and foamy, add the garlic and onions and sauté until soft, about 4 minutes. Stir in the tomatoes, chicken stock, barbecue sauce, Worcestershire, brown sugar, cayenne, smoked pork, lime beans, corn and a big pinch of ...

Brunswick stew - Wikipedia
https://en.wikipedia.org/wiki/Brunswick_stew
Brunswick stew is a traditional dish, popular in the American South, generally involving a tomato base, local beans, vegetables, and originally small game meat, though today often chicken. The origin of the dish is uncertain, but it is believed to have been invented in the early 19th century. The states of Virginia and Georgia ...

Chicken-and-Brisket Brunswick Stew Recipe | MyRecipes
https://www.myrecipes.com/recipe/chicken-brisket-brunswick-stew
Rating: 5 - 8 reviews - 2 hrs 40 mins
Stir together beef soup base and 2 cups water, and add to Dutch oven. Stir in chicken and...

KG result

Brunswick stew

Brunswick stew is a traditional dish, popular in the American South, generally involving a tomato base, local beans, vegetables, and originally small game meat, though today often chicken. The Place of origin: United States of America

People also search for
A knowledge graph

A KG is an entity-centric knowledge repository that:

- describes real-world entities and their interrelations
- defines possible classes and relations of entities
- allows for interrelating arbitrary entities
- covers various topical domains

(Paulheim, 2016)
Knowledge representation in RDF

- RDF is a node-and-arc-labeled directed graph.
- A description of a resource is represented as a number of triples.
- A triple (also: RDF statement / fact): subject, predicate, object.
- Subject: the URI identifying the described resource.
- Object: either a literal value (e.g. a string, number, or date); or
  - the URI of another resource related to the subject.
- Predicate: URI that indicates relation between subject and object.

https://www.w3.org/RDF/
RDF example

Example: as an RDF graph with two triples
Query a knowledge graph: SPARQL query language

Which maize dishes are popular in the United States?

**SELECT** ?dish ?name **WHERE** {
}

https://dbpedia.org/sparql
Search in SPARQL literals

```
SELECT ?subject ?name
WHERE {
FILTER regex(?name, "^maize", "i") }
```
Challenges in query & search over knowledge graphs

- Large / missing / unknown schema
  - But precise graph patterns in SPARQL
- Too many interpretations for pure literal search
  - E.g. "^maize" in DBpedia: plants, locations, schools, etc.

Search "Maize" in DBpedia (an excerpt)

- http://dbpedia.org/resource/Beet_Maize_&_Corn
- http://dbpedia.org/resource/La_Villafranca-Bolleneве-et-la-Maize
- http://dbpedia.org/resource/Maize,_Waterburg,_Ge
- http://dbpedia.org/resource/Colliery_Maizery
- http://dbpedia.org/resource/2014_Canberra_United_W-League_season_Melissa_Maizels_1
- http://dbpedia.org/resource/Maize,_Kansas
- http://dbpedia.org/resource/Maizet
- http://dbpedia.org/resource/Kelley_Maize
- http://dbpedia.org/resource/Fort_de_Maizere
- http://dbpedia.org/resource/Maize_rural
- http://dbpedia.org/resource/Michael_Maize
- http://dbpedia.org/resource/International_Maize_and_Wheat_Improvement_Center
- http://dbpedia.org/resource/Maize_South_High_School
- http://dbpedia.org/resource/Pierres_Maizeaux
- http://dbpedia.org/resource/Maize_chlorotic_dwarf_virus

https://dbpedia.org/sparql
Overview of popular knowledge graphs

Instances: # of instances defined in the graph
Facts: # of statements about those instances
Entity types: # of different types defined in the schema
Relation types: # of different relations defined in the schema

<table>
<thead>
<tr>
<th>Name</th>
<th>Instances</th>
<th>Facts</th>
<th>Types</th>
<th>Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBpedia (English)</td>
<td>4,806,150</td>
<td>176,043,129</td>
<td>735</td>
<td>2,813</td>
</tr>
<tr>
<td>YAGO</td>
<td>4,595,906</td>
<td>25,946,870</td>
<td>488,469</td>
<td>77</td>
</tr>
<tr>
<td>Freebase</td>
<td>49,947,845</td>
<td>3,041,722,635</td>
<td>26,507</td>
<td>37,781</td>
</tr>
<tr>
<td>Wikidata</td>
<td>15,602,060</td>
<td>65,993,797</td>
<td>23,157</td>
<td>1,673</td>
</tr>
<tr>
<td>NELL</td>
<td>2,006,896</td>
<td>432,845</td>
<td>285</td>
<td>425</td>
</tr>
<tr>
<td>OpenCyc</td>
<td>118,499</td>
<td>2,413,894</td>
<td>45,153</td>
<td>18,526</td>
</tr>
<tr>
<td>Google’s Knowledge Graph</td>
<td>570,000,000</td>
<td>18,000,000,000</td>
<td>1,500</td>
<td>35,000</td>
</tr>
<tr>
<td>Google’s Knowledge Vault</td>
<td>45,000,000</td>
<td>271,000,000</td>
<td>1,100</td>
<td>4,469</td>
</tr>
<tr>
<td>Yahoo! Knowledge Graph</td>
<td>3,443,743</td>
<td>1,391,054,990</td>
<td>250</td>
<td>800</td>
</tr>
</tbody>
</table>

(Paulheim, 2016)
Information Extraction
Motivation

Unstructured data, i.e. text, is written for humans, not machines.

Information Extraction enables machines to automatically identify information nuggets such as named entities, time expressions, relations and events in text and interlink these information nuggets with structured background knowledge.

Extracted information can then be used e.g. to categorize and cluster text, enable faceted exploration, extract semantics, populate knowledge bases, correlate extracted data with other sources, e.g. across languages etc.

“Learn to read better than yesterday.” NELL project

http://rtw.ml.cmu.edu/rtw/overview
Aims of the session

• Provide an overview of the methods of Information Extraction, in particular for:
  • Named Entity Extraction
  • Named Entity Linking
  • Temporal Extraction
  • Relation Extraction

• Understand how different methods of Information Extraction work
  • Rule-based approaches
  • Machine learning approaches
  • Different supervision models for machine learning
(Semi-)structured Information on the Web

David Farley
Sunday 6 December 2015 14.00 GMT

On my first day in Dubrovnik, the stunning walled city on the southern Dalmatian coast, I sat down at an outdoor café on the Stradun, the main limestone-clad pedestrian street in the old town, and ordered a beer. It hit the spot, the crisp pilsner washing away the memories of a long flight. But then I got the bill: £5. This wouldn’t have been outrageous if I’d been in, say, Oslo, but here in Croatia, it seemed particularly expensive.

Source: https://www.theguardian.com/travel/2015/dec/06/bar-tour-dubrovnik-croatia-holiday
Data “Hidden” in the Text

Overall 2015 was another record breaking year for Dubrovnik tourism. Last year the city saw 932,621 tourist arrivals, which when added to the number of cruise ship passengers brings the number of tourists in Dubrovnik in 2015 close to 2 million. The number of tourists in Dubrovnik rose by 8 percent in 2015 compared to 2014 and the city achieved 3.3 million overnight stays, an increase of 6 percent on 2014. Once again tourists from Great Britain were the most numerous, followed by guests from the US with German tourists in third place. Breaking down the tourism statistics for Dubrovnik for 2015 even further it is clear that the city is a hit with middle-aged travellers. The majority of tourists fell into the age group of between 41 and 60, whilst in second place were tourists over 60 years old.

Source: http://dubrovacki.hr/clanak/81000/2015-tourism-figures-for-dubrovnik
Named Entities & Temporal Expressions

Overall 2015 was another record breaking year for Dubrovnik tourism. Last year the city saw 932,621 tourist arrivals, which when added to the number of cruise ship passengers brings the number of tourists in Dubrovnik in 2015 close to 2 million. The number of tourists in Dubrovnik rose by 8 percent in 2015 compared to 2014 and the city achieved 3.3 million overnight stays, an increase of 6 percent on 2014. Once again tourists from Great Britain were the most numerous, followed by guests from the US with German tourists in third place. Breaking down the tourism statistics for Dubrovnik for 2015 even further it is clear that the city is a hit with middle-aged travellers. The majority of tourists fell into the age group of between 41 and 60, whilst in second place were tourists over 60 years old.

Source: http://dubrovacki.hr/clanak/81000/2015-tourism-figures-for-dubrovnik
Entity Linking

Overall 2015 was another record breaking year for Dubrovnik tourism. Last year the city saw 932,621 tourist arrivals, which when the number of cruise ship passengers brings the number of tourists in Dubrovnik close to 2 million. The number of tourists in Dubrovnik rose compared to 2014 and the city achieved 3.3 million arrivals, an increase of 6 percent on 2014. Once again tourists from Germany were the most numerous, followed by guests from the US with German tourists in third place. Breaking down the tourism statistics for Dubrovnik for 2015 even further it is clear that the city is a hit with middle-aged travellers. The majority of tourists fell into the age group of between 41 and 60, whilst in second place were tourists over 60 years old.

Information Extraction (IE)

• IE is the task of identification of structured information in text. IE includes:
  ■ Named Entity extraction and disambiguation
    ▪ Dubrovnik is a city. Dubrovnik -> http://dbpedia.org/resource/Dubrovnik
  ■ Extraction of temporal expressions
  ■ Extraction of relations between Named Entities
    ▪ Dubrovnik is located in the region of Dalmatia.
  ■ Event extraction
    ▪ One of Croatia’s most famous events, the Dubrovnik Summer Festival, took place from 10th July to 25th August 2016.
Named Entity Extraction: Terminology

**Named Entities**: Proper nouns or phrases, which refer to real-world objects (entities).

**Named Entity Extraction (Recognition, Identification)**: Detecting boundaries of named entities (NEs) in unstructured text.

**Named Entity Classification**: Automatically assigning pre-defined classes to NEs, such as PERSON, LOCATION, ORGANISATION, etc.

**Named Entity Linking / Disambiguation**: Linking NEs to entries in a knowledge graph (e.g. DBpedia, Wikidata, etc.).

Dubrovnik -> [http://dbpedia.org/resource/Dubrovnik](http://dbpedia.org/resource/Dubrovnik)

-> [https://www.wikidata.org/wiki/Q1722](https://www.wikidata.org/wiki/Q1722)
Named Entity Extraction: Examples

Dubrovnik is a Croatian city on the Adriatic Sea, in the region of Dalmatia founded in the 7th century. The Imperial Fortress was built in 1806 by Marshal Marmont in honor of emperor Napoleon. The HBO series Game of Thrones used Dubrovnik as a filming location, representing the cities of King's Landing and Qarth.

Extraction by Stanford Named Entity Tagger

Any issues?

NE Classification

LOCATION
ORGANIZATION
DATE
MONEY
PERSON
PERCENT
TIME
Named Entity Extraction: Examples

Dubrovnik is a Croatian city on the Adriatic Sea, in the region of Dalmatia, founded in the 7th century. The Imperial Fortress was built in 1806 by Marshal Marmont in honor of emperor Napoleon. The HBO series Game of Thrones used Dubrovnik as a filming location, representing the cities of King's Landing and Qarth.

Problems:
- Unknown entities
- Unknown entity types
- Ambiguities / wrong types

NE Classification
- LOCATION
- ORGANIZATION
- DATE
- MONEY
- PERSON
- PERCENT
- TIME
Named Entity Extraction: Methods

- **Rule-based approaches**: Using hand-coded extraction rules
- **Machine learning based approaches**
  - Supervised learning (domain specific): Manually annotate the text, train a model
  - Unsupervised learning (Web-scale NER): Extract language patterns, cluster similar ones
  - Semi-supervised learning: Start with a small number of language patterns, iteratively learn more (bootstrapping)
- **Methods based on using existing resources**
  - Gazetteer-based method: Use existing list of named entities
  - Using Web resources, KGs / knowledge bases: Wikipedia, DBpedia, Web n-gram corpora, etc.
- **Combinations of the methods above**
NERC: Choice of Machine Learning Algorithms

NE Extraction Pipeline

Pre-processing of text
- Text extraction (mark up removal), sentence splitting, tokenization (identification of individual terms)

Linguistic pre-processing of tokens
- Lemmatisation (lexicon) or stemming (algorithms):
  - reduce inflectional forms of a word to a common base form
- Part of speech (POS) tagging

Chunking (shallow parsing), parsing (parse tree)
- Noun phrases, grammatical relations

Semantic and discourse analysis, anaphora resolution (co-references)
- What actions are being described? What roles entities play in this actions? How does they relate to other entities and actions in other sentences?
NE Extraction Pipeline

- Sentence splitting: *Dubrovnik is located in the region of Dalmatia.*
NE Extraction Pipeline

- Sentence splitting: *Dubrovnik is located in the region of Dalmatia.*
- Tokenization: "Dubrovnik is located in the region of Dalmatia."
NE Extraction Pipeline

- Sentence splitting: *Dubrovnik is located in the region of Dalmatia.*
- Tokenization: *Dubrovnik is located in the region of Dalmatia.*

Lemmatisation or stemming (reduce inflectional forms of a word to a common base form):
- E.g. “located” -> “locate”
NE Extraction Pipeline

- Sentence splitting: *Dubrovnik is located in the region of Dalmatia.*
- Tokenization: Dubrovnik is located in the region of Dalmatia.

Lemmatisation or stemming:
- E.g. “located” -> “locate”

POS tagging: Nouns, adjectives and verbs
**Morphology: Penn Treebank POS tags**

<table>
<thead>
<tr>
<th>Number</th>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>CC</td>
<td>Coordinating conjunction</td>
</tr>
<tr>
<td>2.</td>
<td>CD</td>
<td>Cardinal number</td>
</tr>
<tr>
<td>3.</td>
<td>DT</td>
<td>Determiner</td>
</tr>
<tr>
<td>4.</td>
<td>EX</td>
<td>Existential <em>there</em></td>
</tr>
<tr>
<td>5.</td>
<td>FW</td>
<td>Foreign word</td>
</tr>
<tr>
<td>6.</td>
<td>IN</td>
<td>Preposition or subordinating conjunction</td>
</tr>
<tr>
<td>7.</td>
<td>JJ</td>
<td>Adjective</td>
</tr>
<tr>
<td>8.</td>
<td>JJR</td>
<td>Adjective, comparative</td>
</tr>
<tr>
<td>9.</td>
<td>JJS</td>
<td>Adjective, superlative</td>
</tr>
<tr>
<td>10.</td>
<td>LS</td>
<td>List item marker</td>
</tr>
<tr>
<td>11.</td>
<td>MD</td>
<td>Modal</td>
</tr>
<tr>
<td>12.</td>
<td>NN</td>
<td>Noun, singular or mass</td>
</tr>
<tr>
<td>13.</td>
<td>NNS</td>
<td>Noun, plural</td>
</tr>
<tr>
<td>14.</td>
<td>NNP</td>
<td>Proper noun, singular</td>
</tr>
<tr>
<td>15.</td>
<td>NNPS</td>
<td>Proper noun, plural</td>
</tr>
<tr>
<td>16.</td>
<td>PDT</td>
<td>Predeterminer</td>
</tr>
<tr>
<td>17.</td>
<td>POS</td>
<td>Possessive ending</td>
</tr>
<tr>
<td>18.</td>
<td>PRP</td>
<td>Personal pronoun</td>
</tr>
<tr>
<td>19.</td>
<td>PRPS</td>
<td>Possessive pronoun</td>
</tr>
<tr>
<td>20.</td>
<td>RB</td>
<td>Adverb</td>
</tr>
<tr>
<td>21.</td>
<td>RBR</td>
<td>Adverb, comparative</td>
</tr>
<tr>
<td>22.</td>
<td>RBS</td>
<td>Adverb, superlative</td>
</tr>
<tr>
<td>23.</td>
<td>RP</td>
<td>Particle</td>
</tr>
<tr>
<td>24.</td>
<td>SYM</td>
<td>Symbol</td>
</tr>
<tr>
<td>25.</td>
<td>TO</td>
<td><em>to</em></td>
</tr>
<tr>
<td>26.</td>
<td>UH</td>
<td>Interjection</td>
</tr>
<tr>
<td>27.</td>
<td>VB</td>
<td>Verb, base form</td>
</tr>
<tr>
<td>28.</td>
<td>VBD</td>
<td>Verb, past tense</td>
</tr>
<tr>
<td>29.</td>
<td>VBG</td>
<td>Verb, gerund or present participle</td>
</tr>
<tr>
<td>30.</td>
<td>VBN</td>
<td>Verb, past participle</td>
</tr>
<tr>
<td>31.</td>
<td>VBP</td>
<td>Verb, non-3rd person singular present</td>
</tr>
<tr>
<td>32.</td>
<td>VBZ</td>
<td>Verb, 3rd person singular present</td>
</tr>
<tr>
<td>33.</td>
<td>WDT</td>
<td></td>
</tr>
<tr>
<td>34.</td>
<td>WP</td>
<td></td>
</tr>
<tr>
<td>35.</td>
<td>WPS</td>
<td></td>
</tr>
<tr>
<td>36.</td>
<td>WRB</td>
<td></td>
</tr>
</tbody>
</table>

**Verbs (all start with V)**

**Adjectives (all start with J)**

**Nouns (all start with N)**

Source: Isabelle Augenstein, Information Extraction with Linked Data Tutorial, ESWC Summer School 2015

**Dubrovnik is located in the region of Dalmatia.**
NE Extraction Pipeline

- Sentence splitting: *Dubrovnik is located in the region of Dalmatia.*
- Tokenization: Dubrovnik is located in the region of Dalmatia.

Lemmatisation or stemming:
- E.g. “located” -> “locate”

POS tagging: Nouns, adjectives and verbs

Chunking, parsing:
- “Dubrovnik is located”, “region of Dalmatia”

Noun phrases, grammatical relations

nsubj: nominal subject - a noun phrase, the syntactic subject of a clause.
NE Extraction Pipeline

- Sentence splitting: *Dubrovnik is located in the region of Dalmatia.*

- Tokenization: Dubrovnik is located in the region of Dalmatia.

Lemmatisation or stemming:

- E.g. “located” -> “locate”

POS tagging: Nouns, adjectives and verbs

Chunking, parsing:

- “Dubrovnik is located”, “region of Dalmatia”

Co-reference resolution:

- “Dubrovnik” and “This city”.

(Examples: Stanford NER)
NE Extraction Pipeline

- Sentence splitting: *Dubrovnik is located in the region of Dalmatia.*

- Tokenization: **Dubrovnik** is located in the region of **Dalmatia.**

Lemmatisation or stemming:
  - E.g. “located” -> “locate”

POS tagging: Nouns, adjectives and verbs

Chunking, parsing:
  - “Dubrovnik is located”, “region of Dalmatia”

Co-reference resolution:
  - “Dubrovnik” and “This city”.

(Examples: Stanford NER)
Named Entity Extraction: Features

- **Words:**
  - Words in window before and after mention
  - Sequences (n-grams), frequencies
  - Bags of words, word2vec

- **Morphology:**
  - Capitalization: is upper case (*China*), all upper case (*IBM*), mixed case (*eBay*)
  - Symbols: contains $, £, €, roman symbols (*IV*), ..
  - Contained special characters: period (*google.com*), apostrophe (*Mandy’s*), hyphen (*speed-o-meter*), ampersand (*Fisher & Sons*)
  - Stem or lemma (*cats->cat*), prefix (*disadvantages->dis*), suffix (*cats->s*), interfix (*speed-o-meter->o*)
Named Entity Extraction: Features

- POS tags, POS tag patterns
  - NN and NNS singular and plural nouns
  - NNP proper nouns
  - JJ adjectives
- Near n-Gram punctuation
- Other n-gram related features

Top 6 most frequent part-of-speech tag patterns of the SIGIR collection.

(Prokofyev 2014)
Named Entity Extraction: Features

- Gazetteers
  - Using regular expressions patterns and search engines (e.g. “Popular artists such as * ”)
  - Retrieved from knowledge graphs
    - General: Wikipedia, DBpedia, Wikidata, (Freebase)
    - Domain-specific: DBLP, Physics Concepts, etc.
  - Retrieved from the Web tables and lists

Sources: https://en.wikipedia.org/wiki/List_of_German_Green_Party_politicians
http://www.brainyquote.com/quotes/keywords/artist.html

Whether looking at pop music, hip-hop or R&B, it’s rare to find an artist who hasn’t been touched or affected by the power and soul of gospel music. In fact, many of today’s popular artists such as Whitney Houston, John Legend, and Katy Perry started their careers in the church choir.

Marvin Sapp
Named Entity Extraction: Evaluation Measures

• Precision
  – Proportion of correctly extracted NEs among all extracted NEs.

• Recall
  – Proportion of NEs found by the algorithm to all NEs in the collection.

• F-Measure
  – The weighted harmonic mean of precision and recall.
Open Problems in NER

Extraction does not work equally well in all domains

- Specialised technical texts
- Other languages / multilingual text collections

Newly emerging / unknown entities (e.g. in the context of news events)

- Edward Snowden before the NSA scandal
- Regional entities (e.g. not widely known politicians)
- Annotating named entities in local news papers

Entity evolution (entity name or attribute changes over time)

- St. Petersburg vs. Leningrad and Petrograd
- Pope Francis vs. Jorge Mario Bergoglio

“…Bombay, also known as Mumbai…”

- January 09, 2000 - Arts – Article NYTimes
Entity Linking

Entity Linking (EL): detecting entities and linking them to the entries of a Knowledge Graph

Dubrovnik is located in the region of Dalmatia.

-> http://dbpedia.org/resource/Dubrovnik
-> https://www.wikidata.org/wiki/Q1722
Entity Linking: Motivation

Provide additional information / facts about the entities in the text
Uncover relations between entities

**Dubrovnik**

-> [http://dbpedia.org/resource/Dubrovnik](http://dbpedia.org/resource/Dubrovnik)

-> [https://www.wikidata.org/wiki/Q1722](https://www.wikidata.org/wiki/Q1722)

The Republic of Ragusa was a maritime republic centered on the city of Dubrovnik. It was conquered by Napoleon's French Empire in 1808.
Entity Linking: Related Problems

- Knowledge Graph population
  - Populate a KG with named entities identified in text
    - -> lifting unstructured data into a pre-defined structure

- Interlinking records across databases
  - Determine records represent the same entity to be merged (referred to as object identification, data de-duplication, entity resolution, entity disambiguation and record linkage)
    - -> matching structured data instances

- Co-reference resolution or entity resolution
  - Clustering entity mentions either within the same document or across multiple documents together, where each cluster corresponds to a single real-world entity
    - -> matching instances extracted from unstructured data
Entity Linking: Pipeline

Spotting
- Detecting all non-overlapping strings in a text that could mention an entity
- Methods: Named Entity Recognition, detecting multi-word entities, finding sequences of capitalized words, surface form dictionary

Candidate generation
- Finding all possible candidate entities in KG that may be referred to the spotted string
- Methods: query expansion and matching

Candidate disambiguation
- Selection of the most likely candidate in KG (if any)
- Methods: ranking, classification
Entity Linking: Disambiguation Challenges

Name variation / evolution
- The same entity can be referred to by different names
  - Pope Francis, Franciscus, Jorge Mario Bergoglio
- Methods: Dictionary

String ambiguity
- The same name string can refer to more than one entity
- Methods: use of context
  - “Eclipse, is a 2010 American romantic fantasy film”
  - “Eclipse is famous for its Java IDE”
  - “On August 21, 2017, North America was treated to an eclipse of the sun.“

Absence / KG incompleteness
- Many mentioned entities may not appear in a KG (NIL)
- Methods: classification, thresholds
Entity Linking: Disambiguation Features

Statistics
- TF-IDF (frequency and selectivity of candidates)

Entity context similarity
- Context in the observed phrase and in the textual description in the KG
- Dependencies among entities in text and KG

Entity type information
- Restrictions to specific types (e.g. PERSON, LOC, ORG or domain-specific)

Link-based measures
- Popularity: hyperlinks between entities in KG (normalised inlink count or PageRank)
- Link context: anchor text (e.g. in Wikipedia)

"Eclipse is famous for its Java IDE"
Entity Linking: Selected Tools

**DBpedia Spotlight (hands on)**

**AIDA**
- Maps mentions of ambiguous names onto canonical entities (e.g. individual people or places) registered in the YAGO2 knowledge base [https://github.com/yago-naga/aida](https://github.com/yago-naga/aida)

**Illinois Wikifier**
- Disambiguating concepts and entities in a context sensitive way in Wikipedia [http://cogcomp.org/page/software_view/Wikifier](http://cogcomp.org/page/software_view/Wikifier)

**Babelfy (hands on)**
- Babelfy is a unified, multilingual, graph-based approach to Entity Linking and Word Sense Disambiguation. [http://babelfy.org/about](http://babelfy.org/about)
Temporal Extraction

Temporal extraction is the extraction and normalization of temporal expressions.

- **TimeML**: specification language for temporal expressions (Pustejovsky et al. 2005)

**Types of temporal expressions in TimeML**

- Date: “August 23, 2017”, “tomorrow”.
- Time: “11 a.m.”, “3 in the afternoon”.
- Duration (length of an interval): “three years”, “since yesterday”.
- Set (periodical aspect of an event): “twice a month”.
Temporal Extraction: Pipeline

**Pre-processing**
- Linguistic pre-processing documents with sentence, token, and POS tagging.
- Identification of the publication date (e.g. for news articles).

**Pattern extraction**
- Goal: decide whether a token is part of a temporal expression.
- Methods: rule-based or machine learning (as a classification problem).
- Features, patterns: terms frequently used to form temporal expressions; names of months and weekdays or numbers that may refer to a day or year; POS, context information.

**Normalization**
- Goal: assign temporal expression a value in a standard format.
- Methods: rule-based. E.g. “January” -> “01”.
- Output: type (date, time, duration, set) and value.
Temporal Extraction

**Explicit expressions**
- Fully specified and can thus be normalized without any further knowledge
  - August 22, 2017

**Implicit expressions**
- Names of days and events that can directly be associated with a point or interval in time
  - Summer Semester 2018, Christmas 2017, World War II

**Relative expressions**
- Require context to normalize
  - “Today”, “the following year”
Temporal Extraction: Examples

Dubrovnik in Yugoslavia and Croatia

After World War I, Dubrovnik became part of Croatia which itself was part of the Kingdom of Serbs, Croats and Slovenes which became Yugoslavia after World War II.

Dubrovnik was subjected to considerable shelling by Serbs during the war in 1991/2 in a siege that lasted seven months. The Old Town suffered considered damage, but was quickly restored to its former beauty.

Annotated Text
(tagged using sTime)

After World War I, Dubrovnik became part of Croatia which itself was part of the Kingdom of Serbs, Croats and Slovenes which became Yugoslavia after World War II. Dubrovnik was subjected to considerable shelling by Serbs during the war in 1991/2 in a siege that lasted seven months. The Old Town suffered considered damage, but was quickly restored to its former beauty.

Any issues? [Compute precision and recall of the extraction results – hands on.]

Source: http://www.visit-croatia.co.uk/index.php/croatia-destinations/dubrovnik/history-dubrovnik/
Temporal Extraction: Tools

**HeidelTime** (Strötgen 2015)
- Online demo: [http://heideltime.ifi.uni-heidelberg.de/heideltime/](http://heideltime.ifi.uni-heidelberg.de/heideltime/)

**SUTime** (Angel 2012)
Relation Extraction

- **Relations** between two or more entities, which relate to one another in real life.
  - A relation is defined in the form of a tuple \( t = (e_1, e_2, ..., e_n) \), where \( e_i \) are entities in a predefined relation \( R \) within document \( D \).

- **Relation extraction:**
  - is a task of detecting relations between entities and assigning relation types to them.

- **Binary relations**: a relation between two entities.
  - located-in(Dubrovnik, Croatia), married-to(Angelina Jolie, Brad Pitt).

- **Higher-order relations**:
  - A 4-ary biomedical point mutation relation: a type of variation, its location, and the corresponding state change from an initial-state to an altered-state.
  - “At codons 12, the occurrence of point mutations from G to T were observed”
  - point mutation(codon, 12, G, T).
Relation Extraction: Features

Syntactic features

- the entities
- the types of the entities
- word sequence between the entities
- path length (e.g. the number of words between the entities) [3]

Semantic features

- the path between the two entities in the dependency parse tree

Example:

Located-in(Dubrovnik, the region of Dalmatia)
Relation Extraction: Methods

Supervised learning

- E.g. as binary classification

Unsupervised

Semi-supervised and bootstrapping approaches
Relation Extraction: Supervised Methods

- Relation extraction as a binary classification problem
  - Given a set of features extracted from the sentence $S$, decide if entities in $S$ are connected using given relation $R$.

- Disadvantages of supervised methods
  - Need for labelled data. Difficult to extend to new relation types.
  - Extensions to higher order entity relations are difficult as well.
  - Errors in the pre-processing (feature extraction, e.g. parse tree) affect the performance.
  - Pre-defined relations only.
Relation Extraction: Semi-supervised Methods

- Semi-supervised and bootstrapping approaches (e.g. KnowItAll (Etzioni et al., 2005) and TextRunner (Banko et al., 2007))

  - “Weak supervision”: Require a small set of tagged seed instances or a few hand-crafted extraction patterns per relation to launch the training process.
  - Use the output of the weak learners as training data for the next iteration.

  - **Step1**: Use the seed examples to label some data.
  - **Step2**: Induce patterns from the labelled examples.
  - **Step3**: Apply the patterns to data, to get a new set of pairs.
  - **Return to Step2**, and **iterate** until convergence criteria is reached.
Relation Extraction: Semi-Supervised Methods

- **Relation to be extracted** *(author, book)*

- **Step 1**: Use the seed examples to label data.
  - Start with one seed (Arthur Conan Doyle, The Adventures of Sherlock Holmes).
  - Pattern [order, author, book, *prefix*, *suffix*, *middle*].
  - Order = 1 if the author string occurs before the book string and 0 otherwise.
  - *Prefix* and *suffix* are strings of 10 characters to the left/right of the match.
  - *Middle* is the string occurring between the author and book.

Examples from DIPRE (Brin, 1998)
Relation Extraction: Semi-Supervised Methods

- **Step1 (continued):** Use the seed examples to label data.

- [order, author, book, prefix, suffix, middle].

- (Arthur Conan Doyle, The Adventures of Sherlock Holmes)

- $S_1$="know that Sir Arthur Conan Doyle wrote The Adventures of Sherlock Holmes, in 1892"
Relation Extraction: Semi-Supervised Methods

- **Step1 (continued):** Use the seed examples to label data.

- [order, author, book, *prefix*, *suffix*, *middle*].

- (Arthur Conan Doyle, The Adventures of Sherlock Holmes)

- $S1$="*know that Sir* Arthur Conan Doyle *wrote* The Adventures of Sherlock Holmes, *in 1892"

Relation Extraction: Semi-Supervised Methods

- **Step1 (continued):** Use the seed examples to label data.
  - [order, author, book, prefix, suffix, middle].
  - (Arthur Conan Doyle, The Adventures of Sherlock Holmes)
  - S1="know that Sir Arthur Conan Doyle wrote The Adventures of Sherlock Holmes, in 1892"
  - [1, Arthur Conan Doyle, The Adventures of Sherlock Holmes, know that Sir, in 1892, wrote]
  - S2="When Sir Arthur Conan Doyle wrote the adventures of Sherlock Holmes in 1892 he was high ..."
  - [1, Arthur Conan Doyle, The Adventures of Sherlock Holmes, When Sir, in 1892 he, wrote]
Relation Extraction: Semi-supervised Methods

- **Output Step1:** [order, author, book, prefix, suffix, middle].

  [1, Arthur Conan Doyle, The Adventures of Sherlock Holmes, *now that Sir*, in 1892, wrote]

- **Step2:** Induce patterns from the labelled examples.
  - Exact match: generalize the pattern: *[Sir, .*?, wrote, .*?, in 1892]*.
  - Approximate match: use similarity metrics for patterns. (Agichtein & Gravano, 2000)

- **Step3:** Apply the patterns to data, to get a new set of pairs.

- **Return to Step2**, and **iterate** until convergence criteria is reached.
Open Information Extraction (Open IE)

Open IE extracts tuples consisting of argument phrases and a relation between the arguments

- (arg₁; rel; arg₂).

- For example: $S3$=“Trump (arg₁) was elected (pred) President (arg₂).”

- (Trump; was elected; President)

Different to relation extraction

- No pre-specified sets of relations
- No domain-specific knowledge engineering

Example applications

- A news reader who wishes to keep abreast of important events
- An analyst who recently acquired a terrorist’s laptop
Open IE: TextRunner

Relation is a tuple: \( t = (e_1, r, e_2) \)
- \( e_1 \) and \( e_2 \) are surface forms of entities or noun phrases.
- \( r \) denotes a relationship between \( e_1 \) and \( e_2 \).

TextRunner components:
- Self-supervised Learner: automatic labelling of training data.
- Single-pass Extractor: generates candidate relations from each sentence, runs a classifier and retains the ones labelled as trustworthy relations.
- Redundancy-based Assessor: assigns a probability to each retained tuple based on a probabilistic model of redundancy in text.

(Banko et al., 2007)
7-Step training for each sentence:

**Step 1:** a noun phrase chunker.

**Step 2:** the relation candidator.

**Steps 3-5:** a syntactic parser and dependency parser are run. The relation filter uses parse trees, dependency trees, and set of constraints to label trustworthy relations.

**Step 6:** map each relation to a feature vector representation.

**Step 7:** train a binary classifier using labelled trustworthy and untrustworthy relations.

---

Figure 2: Self-supervised training of Learner module in TextRunner.

(Banko et al., 2007)  
image: (Bach 2013)
Open IE: REVERB and Ollie

REVERB (Fader 2011) and Ollie (Mausam 2012) extract binary relationships from English sentences.

Designed for Web-scale information extraction, where the target relations cannot be specified in advance and speed is important.

REVERB extracts relations mediated by verbs, does not consider the context:
  - shallow syntactic processing to identify relation phrases that begin with a verb and occur between the argument phrases.

Ollie extracts relations mediated by nouns, adjectives, and more.

Ollie includes contextual information from the sentence in the extractions.

http://reverb.cs.washington.edu/
https://knowitall.github.io/ollie/
Open IE: Ollie System Architecture

- Use a set tuples from REVERB to bootstrap a large training set.
- Learn open pattern templates over this training set.
- Apply pattern templates at extraction time.
- Analyse the context around the tuple to add information (attribution, clausal modifiers) and a confidence function.
**Open IE: Ollie Bootstrapping**

The goal is to automatically create a large training set, which encapsulates the multitudes of ways in which information is expressed in text.

Almost every relation can also be expressed via a REVERB-style verb-based expression.

Retrieve all sentences in a Web corpus that contain all content words in the tuple.

Assumption: sentences express the relation of the original seed tuple.

Not always true:
- (Boyle; is born in; Ireland)
- “Felix G. Wharton was born in Donegal, in the northwest of Ireland, a county where the Boyles did their schooling.”
Open IE: Ollie Bootstrapping

Over 110,000 seed tuples – high confidence REVERB extractions from ClueWeb - a large Web corpus. Contain only proper nouns in the arguments.

- Seed: “Paul Annacone is the coach of Federer.” ->
- REVERB pattern: (Paul Annacone; is the coach of; Federer).
- Retrieved sentence: “Now coached by Annacone, Federer is winning more titles than ever.”

Enforce additional dependency restrictions on the sentences to reduce bootstrapping errors. Restrict linear path length between argument and relation in the dependency parse (max 4).
REVERB vs. Ollie

“Early astronomers believed that the earth is the center of the universe.”

- R: (the earth; be the center of; the universe)
- O: ((the earth; be the center of; the universe) AttributedTo believe; Early astronomers)

“If he wins five key states, Romney will be elected President.”

- R: (Romney; will be elected; President)
- O: ((Romney; will be elected; President) ClausalModifier if; he wins five key states)
Relation Extraction with Distant Supervision

The typical workflow:

1. Detect entity mentions
2. Perform entity linking
3. Include all KG types of the KG-mapped entity;
4. Include all KG relation types between the KG-mapped entities.

Use the automatically labeled training corpus to infer types of the unlinkable candidate mentions.

Problems:
- Domain restriction
- Error propagation
- Label noise

Image: (Xiang Ren, et al., WWW 2017)
Joint Extraction of typed Entities and Relations with Distant Supervision

Goal: jointly extract entities and relations of target types with minimal or no human supervision.

Approach:
1) Detect candidate entity mentions with distant supervision and POS tagging
2) Model the mutual constraints between the types of the relation mentions and the types of their entity arguments
3) Rank relevant relation types (as opposed to every candidate type considered relevant to the mention).

(Xiang Ren, et al., WWW 2017)
NLP & ML Software

Natural Language Processing:
- Stanford NLP (Java)
- GATE (general purpose architecture, includes other NLP and ML software as plugins)
- OpenNLP (Java)
- NLTK (Python)

Machine Learning:
- scikit-learn (Python)
- Mallet (Java)
- WEKA (Java)
- Alchemy (graphical models, Java)
- FACTORIE, Wolfe (graphical models, Scala)
- CRFSuite (efficient implementation of CRFs, Python)
- Apache Spark MLLib
- Apache Mahout
NLP & ML Software

Ready to use NERC software:
- ANNIE (rule-based, part of GATE)
- Wikifier (based on Wikipedia)
- FIGER (based on Wikipedia, fine-grained Freebase NE classes)

Almost ready to use NERC software:
- CRFSuite (already includes Python implementation for feature extraction, you just need to feed it with training data, which you can also download)

Ready to use RE software:
- ReVerb, Ollie (Open IE, extract patterns for any kind of relation)
- MultiR (Distant supervision, relation extractor trained on Freebase)

Web content extraction software:
- Boilerpipe (extract main text content from Web pages)
- Jsoup (traverse elements of Web pages individually, also allows to extract text)
- iCrawl (focused web crawling)
  http://icrawl.l3s.uni-hannover.de/
Summary

• In this session we provided an overview of the state-of-the-art Information Extraction methods for:
  • Named Entity Extraction
  • Named Entity Linking
  • Temporal Extraction
  • Relation Extraction

• We addressed
  • Rule-based approaches
  • Machine learning approaches
  • Different supervision models for machine learning
  • An overview of tools

• We will get familiar with selected extraction & linking tools in the hands-on session
Thank you!

Questions, Comments?

Dr. Elena Demidova

L3S Research Center
Leibniz University of Hannover

email: demidova@L3S.de
www: https://demidova.wordpress.com
References

Entity Extraction and Linking


Temporal extraction


• Jannik Strötgen, Michael Gertz. A Baseline Temporal Tagger for All Languages. EMNLP ’15.

References

• Relation extraction and Open IE

References

- **Relation extraction**