Relation Extraction

Matthias Huck and Alexander Fraser

Center for Information and Language Processing
LMU Munich

14 December 2016

Introduction

Reminder:

- Information extraction (IE) $\approx$ extracting structured information from unstructured documents
- Acquire knowledge from natural language text, and store it in a machine-readable format

We already learned how to do (cf. previous IE lectures):

- Named entity recognition (NER)

Next: With named entities already annotated, how can we gather facts from text documents?

- Structured information that may be used to populate a knowledge base
- Such as (typed) relations between named entities

Relation extraction is an enabling technology for:

- Question answering (QA), spoken dialogue systems, recommender systems, text summarization, . . .

Outline

Relation Extraction: Motivation

Relations between Entities: Motivation

`Shortly before Christmas 1966, more than seven years since they first met, Presley proposed to Priscilla Beaulieu. They were married on May 1, 1967, in a brief ceremony in their suite at the Aladdin Hotel in Las Vegas.`

[`https://en.wikipedia.org/wiki/Elvis_Presley`, 6 Dec. 2016]

- NER: PERSON / DATE / LOCATION
- Relation extraction: (We may also utilize a coreference resolution system to resolve `They` / `their`.)
  - *Has_married*(Elvis Presley, Priscilla Beaulieu)
  - *Married_in*(Elvis Presley, Las Vegas)
  - *Married_on*(Elvis Presley, May 1, 1967)
- Application in QA:
  "Where did Priscilla Beaulieu get married?"
  - Analyze question and issue database query
  - *Married_in*(Priscilla Beaulieu, *x*) not in knowledge base.
    (We could have added it, though.)
  - But: *Has_married*(*y*, Priscilla Beaulieu) and *Married_in*(*y*, *x*)

Relations between Entities: Motivation

Shortly before Christmas 1966, more than seven years since they first met, Presley proposed to Priscilla Beaulieu. They were married on May 1, 1967, in a brief ceremony in their suite at the Aladdin Hotel in Las Vegas.

[https://en.wikipedia.org/wiki/Elvis_Presley, 6 Dec. 2016]

- NER: PERSON / DATE / LOCATION
- Relation extraction: (We may also utilize a coreference resolution system to resolve They / their.)
  - *Has_married*(Elvis Presley, Priscilla Beaulieu)
  - *Married_in*(Elvis Presley, Las Vegas)
  - *Married_on*(Elvis Presley, May 1, 1967)
- Application in QA:
  "Where did Priscilla Beaulieu get married?"
  - Analyze question and issue database query
  - *Married_in*(Priscilla Beaulieu, x) not in knowledge base.
    (We could have added it, though.)
  - But: *Has_married*(y, Priscilla Beaulieu) and *Married_in*(y, x)

Relations between Entities: Motivation

`Shortly before Christmas 1966, more than seven years since they first met, Presley proposed to Priscilla Beaulieu. They were married on May 1, 1967, in a brief ceremony in their suite at the Aladdin Hotel in Las Vegas.`

[`https://en.wikipedia.org/wiki/Elvis_Presley`, 6 Dec. 2016]

- NER: PERSON / DATE / LOCATION
- Relation extraction: (We may also utilize a coreference resolution system to resolve `They` / `their`.)
  - *Has_married*(Elvis Presley, Priscilla Beaulieu)
  - *Married_in*(Elvis Presley, Las Vegas)
  - *Married_on*(Elvis Presley, May 1, 1967)
- Application in QA:
  "Where did Priscilla Beaulieu get married?"
  - Analyze question and issue database query
  - *Married_in*(Priscilla Beaulieu, x) not in knowledge base.
    (We could have added it, though.)
  - But: *Has_married*(y, Priscilla Beaulieu) and *Married_in*(y, x)

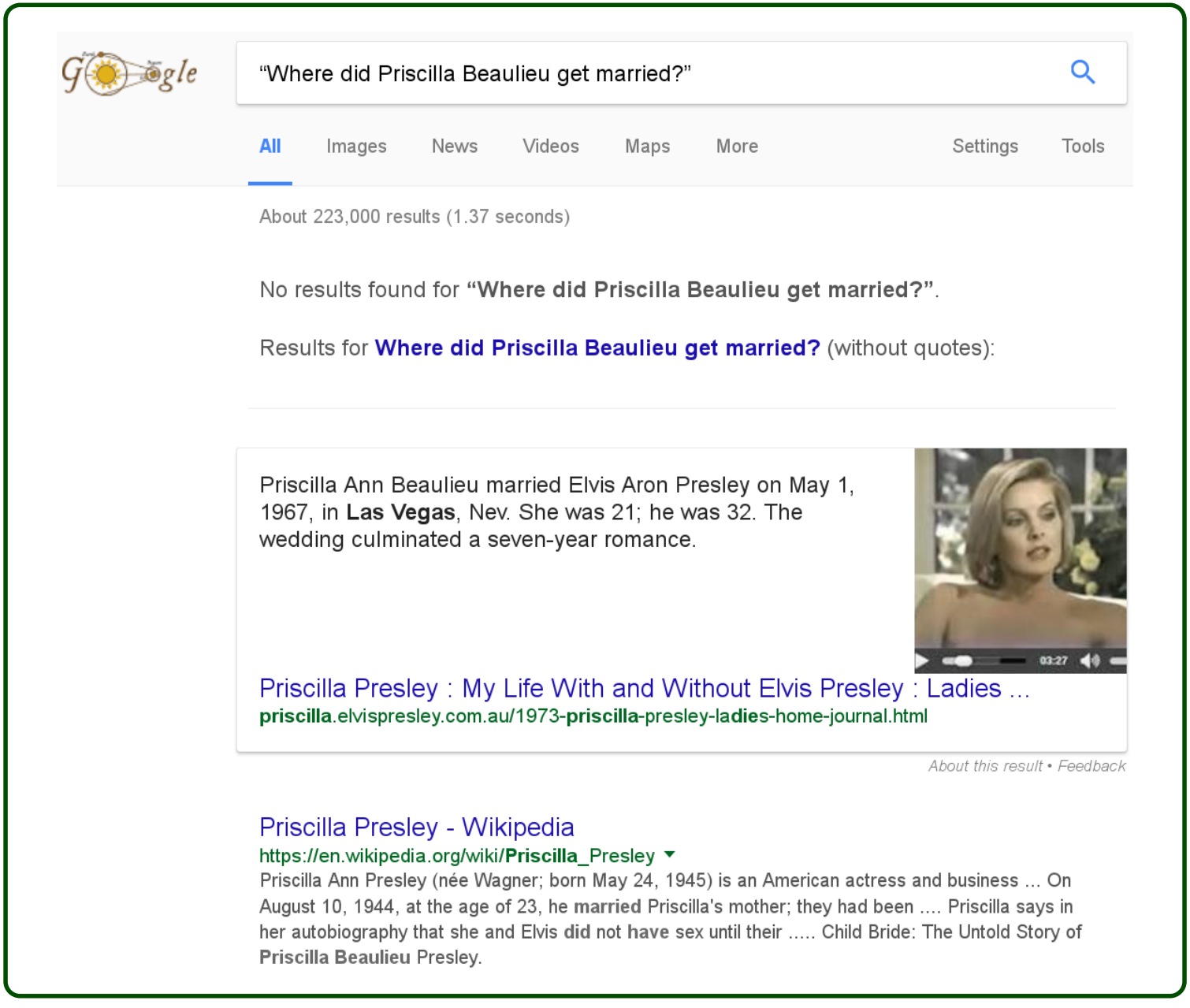
Relations between Entities: Motivation

`Shortly before Christmas 1966, more than seven years since they first met, Presley proposed to Priscilla Beaulieu. They were married on May 1, 1967, in a brief ceremony in their suite at the Aladdin Hotel in Las Vegas.`

[`https://en.wikipedia.org/wiki/Elvis_Presley`, 6 Dec. 2016]

- NER: PERSON / DATE / LOCATION
- Relation extraction: (We may also utilize a coreference resolution system to resolve `They` / `their`.)
  - *Has_married*(Elvis Presley, Priscilla Beaulieu)
  - *Married_in*(Elvis Presley, Las Vegas)
  - *Married_on*(Elvis Presley, May 1, 1967)
- Application in QA:
  "Where did Priscilla Beaulieu get married?"
  - Analyze question and issue database query
  - *Married_in*(Priscilla Beaulieu, x) not in knowledge base.
    (We could have added it, though.)
  - But: *Has_married*(y, Priscilla Beaulieu) and *Married_in*(y, x)

Relations between Entities: Motivation



[Google.co.uk query, 7 Dec. 2016]

Relations between Entities: Motivation

Automatically constructing knowledge bases
(or further populating existing ones)

Freebase

yago
select knowledge

NELL: Never-Ending Language Learning

DBpedia

OpenIE
(Reverb, OLLIE)

Facebook's
Entity Graph

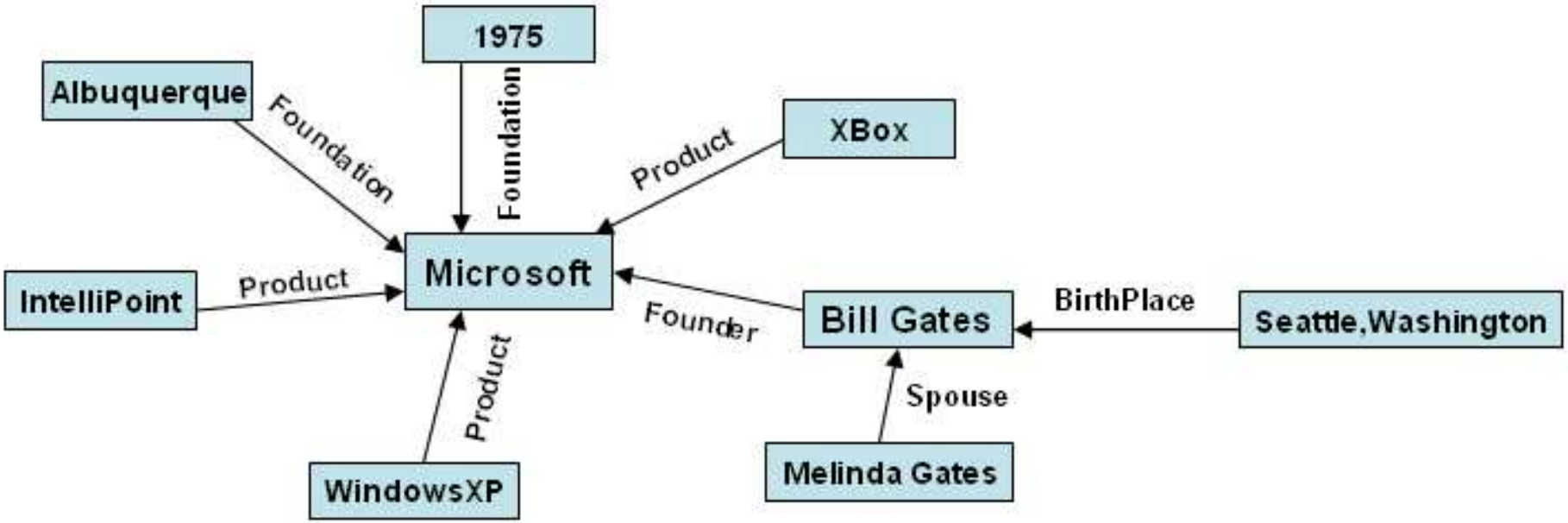
Microsoft's
Satori

Google's
Knowledge Graph

[Antoine Bordes and Evgeniy Gabrilovich. KDD 2014 Tutorial on Constructing and Mining Web-scale Knowledge Graphs, New York, August 24, 2014. Page 8.
`http://www.cs.technion.ac.il/~gabr/publications/papers/KDD14-T2-Bordes-Gabrilovich.pdf`]
[Linking Open Data cloud diagram 2014, by Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak. `http://lod-cloud.net/`. Available under a CC-BY-SA license.]

Relations between Entities: Motivation

Knowledge representation as a directed graph:
entities = nodes, relations = edges



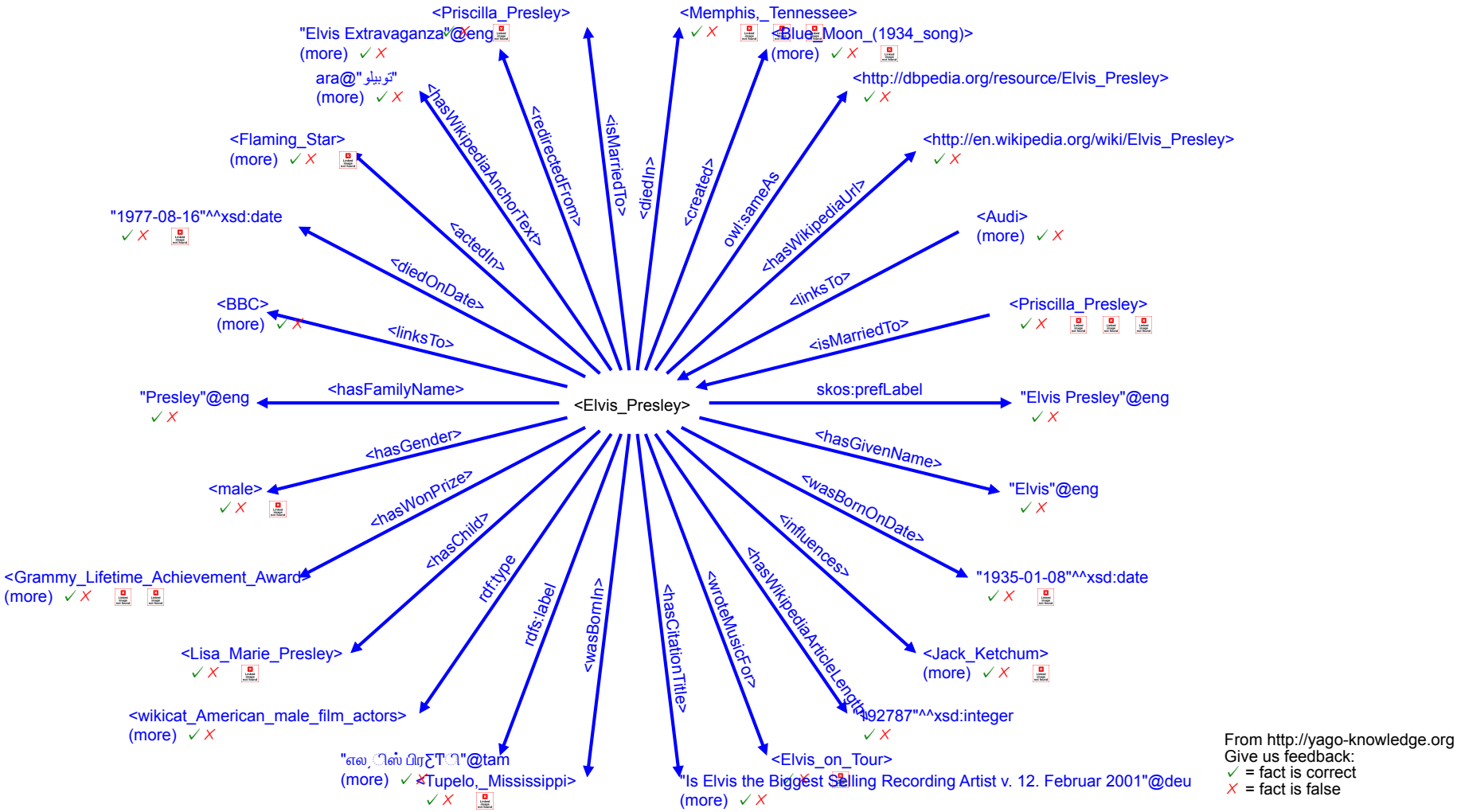
[Dat P.T Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. Exploiting Syntactic and Semantic Information for Relation Extraction from Wikipedia, IJCAI Workshop on Text-Mining & Link-Analysis, Hyderabad, India, 2007. Page 11.]

Relations between Entities: Motivation

Graph browsing the YAGO knowledge base

- $>$ 10 million entities (like persons, organizations, cities, etc.)
- $>$ 120 million facts about these entities

[http://yago-knowledge.org, 8 Dec. 2016]



[https://gate.d5.mpi-inf.mpg.de/webyago3spotlx/SvgBrowser, 8 Dec. 2016]

Relation Extraction: Example Tasks

Example 1: Geographical Location

"Which German federal state is Bogenhausen located in?"

Bogenhausen

From Wikipedia, the free encyclopedia

This article **does not cite any sources**. Please help improve this article by adding citations to relia
Unsourced material may be challenged and removed. *(March 2010) (Learn how and when to remove this t*

Bogenhausen (Central Bavarian: *Bognhausn*) is the 13th borough of Munich, Germany. It is the geographically largest borough

[https://en.wikipedia.org/wiki/Bogenhausen, 6 Dec. 2016]

Munich

From Wikipedia, the free encyclopedia

For other uses of "Munich" or "München", see Munich (disambiguation).

Munich (/ˈmjuːnɪk/; also /ˈmjuːnɪx/ in British English; German: *München*, pronounced ['mʏnçn̩] (listen),[2] Bavarian: *Minga* ['mɪŋ(ː)ɐ]) is the capital and largest city of the German state of Bavaria, on the banks of River Isar north of the Bavarian Alps. Munich is the third largest city in Germany, after Berlin and Hamburg, and the 12th biggest city of

[https://en.wikipedia.org/wiki/Munich, 6 Dec. 2016]

Borough_of(Bogenhausen, Munich) *Capital_of*(Munich, Bavaria)
State_of(Bavaria, Germany)

Example 1: Geographical Location

"Which German federal state is Bogenhausen located in?"

Bogenhausen

From Wikipedia, the free encyclopedia

This article **does not cite any sources**. Please help improve this article by adding citations to relia
Unsourced material may be challenged and removed. *(March 2010) (Learn how and when to remove this t*

Bogenhausen (Central Bavarian: *Bognhausn*) is the 13th borough of Munich, Germany. It is the geographically largest borough

[https://en.wikipedia.org/wiki/Bogenhausen, 6 Dec. 2016]

Munich

From Wikipedia, the free encyclopedia

For other uses of "Munich" or "München", see Munich (disambiguation).

Munich (/ˈmjuːnɪk/; also /ˈmjuːnɪx/ in British English; German: *München*, pronounced ['mʏnçn̩] (listen),[2] Bavarian: *Minga* ['mɪŋ(ː)ɐ]) is the capital and largest city of the German state of Bavaria, on the banks of River Isar north of the Bavarian Alps. Munich is the third largest city in Germany, after Berlin and Hamburg, and the 12th biggest city of

[https://en.wikipedia.org/wiki/Munich, 6 Dec. 2016]

Borough_of(Bogenhausen, Munich) *Capital_of*(Munich, Bavaria)
State_of(Bavaria, Germany)

Example 1: Geographical Location

Some basic structured information is sometimes provided along with unstructured text, e.g. Wikipedia Infoboxes. How to exploit it?

Flag

Coat of arms

München
Munich

Coordinates: 48°08'N 11°34'E

Country	Germany
State	Bavaria
Admin. region	Upper Bavaria
District	Urban district
First mentioned	1158
Government	
• **Lord Mayor**	Dieter Reiter (SPD)
• **Governing parties**	SPD / CSU
Area	
• **City**	310.43 km² (119.86 sq mi)
Elevation	520 m (1,710 ft)
Population (2015-12-31)[1]	
• **City**	1,450,381
• **Density**	4,700/km² (12,000/sq mi)
• **Urban**	2,606,021
Time zone	CET/CEST (UTC+1/+2)
Postal codes	80331–81929
Dialling codes	089
Vehicle registration	M
Website	www.muenchen.de

[https://en.wikipedia.org/wiki/Munich, 6 Dec. 2016]

```
{{Infobox German location
  |imagesize              = 300px
  |image_caption          = From left to right:<br />The [[Munich Frauenkirch
  |Bürgermeistertitel     = Oberbürgermeister
  |Gemeindeschlüssel      = 09 1 62 000
  |Stand                  = 2015-10-31
  |pop_urban              = 2606021
  |pop_ref                = http://www.muenchen.de/sehenswuerdigkeiten/muench
  |name                   = München
  Munich
  |German_name            =
  |image_photo            = Munchen collage.jpg
  |type                   = City
  |image_coa              = Muenchen Kleines Stadtwappen.svg
  |image_flag             = Flag of Munich (striped).svg|120px
  |lat_deg                = 48
  |lat_min= 08
  |lon_deg                = 11
  |lon_min= 34
  |Höhe                   = 520
  |state                  = Bavaria
  |region                 = Upper Bavaria
  |district               = urban
  |elevation              = 519
  |area                   = 310.43
  |population             = 1520408
  |postal_code            = 80331-81929
  |PLZ-alt                = 8000
  |area_code              = 089
  |licence                = M
  |LOCODE                 = DE MUC
  |divisions              = [[Boroughs of Munich|25 boroughs]]
```

Example 2: Food Domain

`I usually have mashed potatoes with my fish fingers.`

Entity types:

- FOOD-ITEM
- EVENT
- DISH
- DISPOSITION

Relations:

- *Suits-to*(FOOD-ITEM, EVENT)
- *Can-be-Served-with*(FOOD-ITEM, FOOD-ITEM)
- *Can-be-Substituted-by*(FOOD-ITEM, FOOD-ITEM)
- *Ingredient-of*(FOOD-ITEM, DISH)
- *Recommended-for-People-with*(FOOD-ITEM, DISPOSITION)
- *Not-Recommended-for-People-with*(FOOD-ITEM, DISPOSITION)

Example 2: Food Domain

Suits-to(??, picnic)
sandwiches, wraps, noodle salad, potato salad, fruit salad, meat balls, filet of pork, vegetables, apples, melons, strawberries, muffins, biscuits, antipasti, . . .

Can-be-Served-with(??, falafel)
lettuce, coleslaw, sauce, yoghurt, tomato salad, olives onions, sesame paste, pita, cucumbers, radish, fries, carrots

Can-be-Substituted-by(??, porridge)
millet gruel, muesli, semolina pudding, cornflakes, grits, oat meal, . . .

Ingredient-of(??, apple pie)
apples, flour, eggs, sugar, cinnamon, yeast, baking powder, butter, milk, margarine, honey, almonds, almond paste, baking soda, sour cream, . . .

Recommended-for-People-with(??, diabetes)
dietary fibre, fish, vegetables, lettuce, fruits, potatoes, magnesium, low-fat yoghurt, low-fat cheese, mineral water, unsweetened tea, muesli, . . .

Not-Recommended-for-People-with(??, diabetes)
alcohol, pastries, butter, soft drinks, sugar, convenience products, fat, sweets, honey, rice pudding, fructose, lactose, fries, sweetened bread spread, . . .

Example 3: Biological Domain

SeeDev task at BioNLP-ST 2016:

- Event extraction of genetic and molecular mechanisms involved in plant seed development

Entity types:

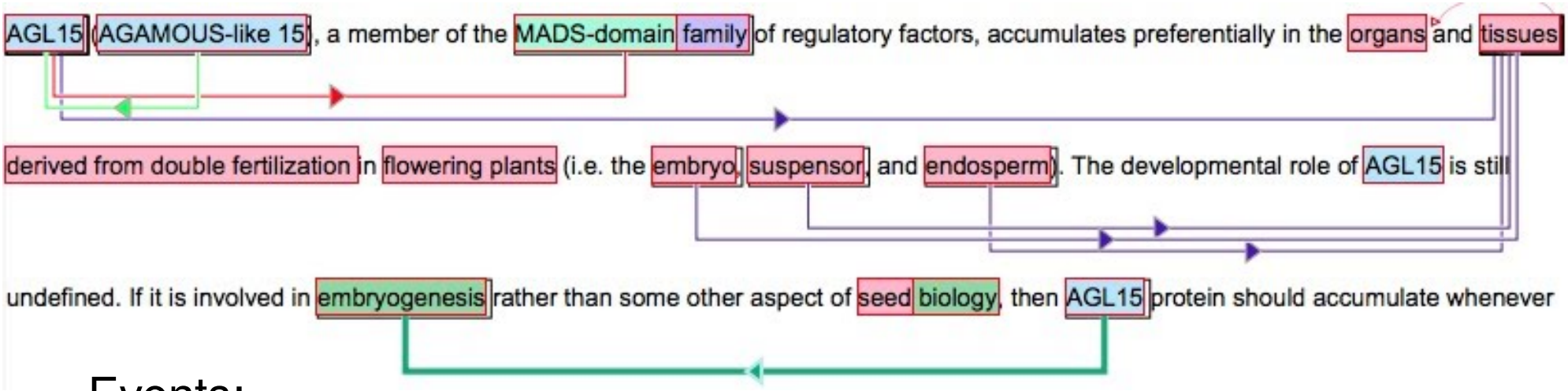
- Gene Gene_Family Box Promoter RNA Protein Protein_Family Protein_Complex Protein_Domain Hormone Regulatory_Network Pathway Genotype Tissue Development_Phase Environmental_Factor

Relations:

- *Binds_To*(Functional_Molecule: Functional_Molecule, Molecule: Molecule)
- *Composes_Primary_Structure*(DNA_Part: Box | Promoter, DNA:DNA)
- *Composes_Protein_Complex*(Amino_Acid_Sequence: Protein | Protein_Family | Protein_Complex | Protein_Domain, Protein_Complex: Protein_Complex)
- *Exists_At_Stage*(Functional_Molecule: Functional_Molecule, Development: Development_Phase)
- ...

[http://2016.bionlp-st.org/tasks/seedev/seedev-data-representation, 6 Dec. 2016]

Example 3: Biological Domain



Events:

Where and When	**Regulation**	**Composition and Membership**
• Presence_In_Genotype	• Regulation_Of_Accumulation	• Primary_Structure_Composition
• Occurrence_In_Genotype	• Regulation_Of_Development_Phase	• Protein_Complex_Composition
• Presence_At_Stage	• Regulation_Of_Expression	• Protein_Domain_Composition
• Occurrence_During	• Regulation_Of_Molecule_Activity	• Family_Membership
• Localization	• Regulation_Of_Process	• Sequence_Identity
Function	• Regulation_Of_Tissue_Development	**Interaction**
• Involvement_In_Process		• Interaction
• Transcription_Or_Translation		• Binding
• Functional_Equivalence		

[http://2016.bionlp-st.org/tasks/seedev, 6 Dec. 2016]

Example 4: NIST Automatic Content Extraction

> *"The objective of the Automatic Content Extraction (ACE) series of evaluations has been to develop human language understanding technologies that provide automatic detection and recognition of key information about real-world entities, relations, and events in source language text, and to convert that information into a structured form, which can be used by follow-on processes, such as classification, filtering and selection, database update, relationship display, and many others."*

ACE08 Evaluation Plan v1.2d, 7 April 2008
[http://www.itl.nist.gov/iad/mig/tests/ace/2008/doc/ace08-evalplan.v1.2d.pdf]

Data sources: broadcast conversations, broadcast news, meetings, newswire, telephone, usenet, weblogs

Example 4: NIST Automatic Content Extraction

Entities (ACE08):

Type	**Subtypes**
FAC (Facility)	Airport, Building-Grounds, Path, Plant, Subarea-Facility
GPE (Geo-Political Entity)	Continent, County-or-District, GPE-Cluster, Nation, Population-Center, Special, State-or-Province
LOC (Location)	Address, Boundary, Celestial, Land-Region-Natural, Region-General, Region-International, Water-Body
ORG (Organization)	Commercial, Educational, Entertainment, Government, Media, Medical-Science, Non-Governmental, Religious, Sports
PER (Person)	Group, Indeterminate, Individual

Example 4: NIST Automatic Content Extraction

Relations (ACE08):

Type	**Subtypes**
ART (artifact)	User-Owner-Inventor-Manufacturer
GEN-AFF (General affiliation)	Citizen-Resident-Religion-Ethnicity, Org-Location
METONYMY	None
ORG-AFF (Org-affiliation)	Employment, Founder, Ownership, Student-Alum, Sports-Affiliation, Investor-Shareholder, Membership
PART-WHOLE (part-to-whole)	Artifact, Geographical, Subsidiary
PER-SOC (person-social)	Business, Family, Lasting-Personal
PHYS (physical)	Located, Near

Example 4: NIST Automatic Content Extraction

Annotated corpus:

Sample of the Datasets for Generic Relation Extraction (LDC2011T08) in Edinburgh Regularized ACE (reACE) mark-up

[`http://benhachey.info/data/gre/examples/ace.xml`, 7 Dec. 2016]

```
- <doc id="15">
  - <text>
    - <p>
      - <s id="s17">
          <w id="w292">American</w>
          <w id="w293">saxophonist</w>
          <w id="w294">David</w>
          <w id="w295">Murray</w>
          <w id="w296">recruited</w>
          <w id="w297">Amidu</w>
          <w id="w298">Berry</w>
          <w id="w299">and</w>
          <w id="w300">DJ</w>
          <w id="w301">Awadi</w>
          <w id="w302">.</w>
        </s>
      </p>
    </text>
```

Example 4: NIST Automatic Content Extraction

```
- <markup>
  - <nes>
    - <ne id="e62" gid="E20" fr="w292" to="w292" t="GPE" st="Nation">
        <textspan type="extent">American</textspan>
        <textspan type="head">American</textspan>
        <exattr n="CLASS" v="SPC"/>
        <exattr n="LDCTYPE" v="PRE"/>
      </ne>
    - <ne id="e61" gid="E18" fr="w292" to="w293" hfr="w293" hto="w293" t="PER">
        <textspan type="extent">American saxophonist</textspan>
        <textspan type="head">saxophonist</textspan>
        <exattr n="CLASS" v="SPC"/>
        <exattr n="LDCTYPE" v="PRE"/>
      </ne>
    - <ne id="e60" gid="E18" fr="w292" to="w295" hfr="w294" hto="w295" t="PER">
        <textspan type="extent">American saxophonist David Murray</textspan>
        <textspan type="head">David Murray</textspan>
        <exattr n="CLASS" v="SPC"/>
        <exattr n="LDCTYPE" v="NAM"/>
      </ne>
    - <ne id="e4" gid="E38" fr="w297" to="w298" t="PER">
        <textspan type="extent">Amidu Berry</textspan>
        <textspan type="head">Amidu Berry</textspan>
        <exattr n="CLASS" v="SPC"/>
        <exattr n="LDCTYPE" v="NAM"/>
      </ne>
    - <ne id="e5" gid="E1" fr="w300" to="w301" hfr="w301" hto="w301" t="PER">
        <textspan type="extent">DJ Awadi</textspan>
        <textspan type="extent">Awadi</textspan>
        <exattr n="CLASS" v="SPC"/>
        <exattr n="LDCTYPE" v="NAM"/>
      </ne>
    </nes>
```

Example 4: NIST Automatic Content Extraction

```
  – <rels>
      <rel id="11-1" gid="11" e1="e61" e2="e62" t="GPE-AFF" st="Citizen-or-Resident"/>
      <rel id="2-1" gid="2" e1="e60" e2="e4" t="PER-SOC" st="Business"/>
      <rel id="3-1" gid="3" e1="e60" e2="e5" t="PER-SOC" st="Business"/>
    </rels>
  </markup>
</doc>
```

Hand-crafted Rules for Relation Extraction

Relation Extraction via Pattern Matching

- Manually identify a set of lexico-syntactic patterns
- Write rules to recognize the patterns in text

Example: hyponym relation *Is_a*(*x*,*y*)

`The bow lute, such as the Bambara ndang, is plucked and has an individual curved neck for each string.`

Pattern: $ENTITY_0$ `such as` { $ENTITY_1$, $ENTITY_2$ ..., (`and` | `or`)} $ENTITY_n$

implies: $\forall ENTITY_i, 1 \leq i \leq n$, *Is_a*($ENTITY_i$, $ENTITY_0$)

i.e.: *Is_a*(`Bambara ndang`, `bow lute`)

[Marti A. Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora, Fourteenth International Conference on Computational Linguistics, Nantes, France, 1992.]

Hearst's Patterns for Hyponym Relations

Hyponym relation *Is_a*(*x*,*y*)

Pattern	**Example occurrence**
y such as *x*	The bow lute, such as the Bambara ndang, ...
such *y* as *x*	... works by such authors as Herrick, Goldsmith, and Shakespeare.
x or other *y*	Bruises, wounds, broken bones or other injuries ...
x and other *y*	... temples, treasuries, and other important civic buildings.
y including *x*	All common-law countries, including Canada and England ...
y , especially *x*	... most European countries, especially France, England, and Spain.

[Marti A. Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora, Fourteenth International Conference on Computational Linguistics, Nantes, France, 1992.]

Harnessing Named Entity Tags

Intuition: relations often hold between specific entities

- *Located_in* (ORGANIZATION, LOCATION)
- *Founded* (PERSON, ORGANIZATION)
- *Cures* (DRUG, DISEASE)
- *Serves_as* (PERSON, POSITION)

Start with named entity tags to help extract relation.

Pattern	**Example occurrence**
`PERSON, POSITION of ORG`	`George Marshall, Secretary of State of the United States`
`PERSON (named|appointed|...) PERSON POSITION`	`Truman appointed Marshall Secretary of State`
`PERSON [be]? (named|appointed|...) ORG POSITION`	`George Marshall was named US Secretary of State`

Slide adapted from Dan Jurafsky [`http://www.stanford.edu/class/cs124/lec/rel.pdf`, 8 Dec. 2016. Pages 19 & 22.]

Hand-crafted Rules: Pros and Cons

Pros:

- Human patterns tend to be high-precision
- Can be tailored to specific domains

Cons:

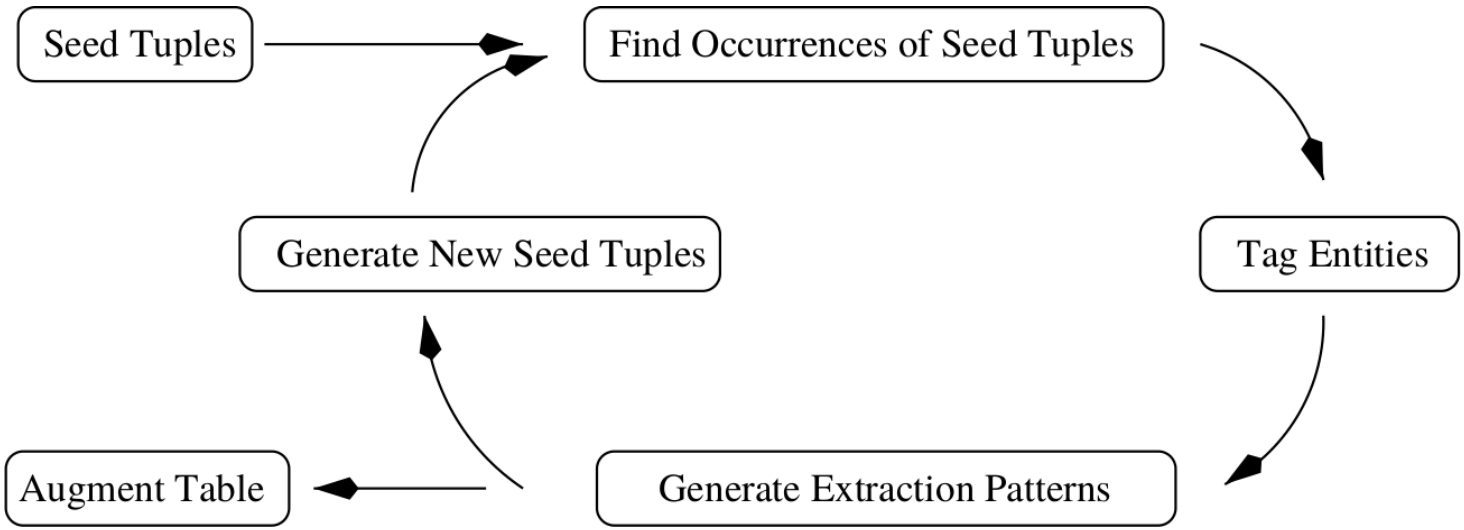
- Human patterns are often low-recall
- A lot of work to think of all possible patterns
- Don’t want to have to do this for every relation type
- We’d like better accuracy

Slide adapted from Dan Jurafsky [`http://www.stanford.edu/class/cs124/lec/rel.pdf`, 8 Dec. 2016. Pages 23.]

Rule Learning for Relation Extraction

Learning New Patterns (Semi-Supervised)

Initialize with hand-crafted rules, iteratively find new ones (*bootstrapping*)



[Eugene Agichtein and Luis Gravano. Snowball: Extracting Relations from Large Plain-Text Collections, Fifth ACM Conference on Digital Libraries. San Antonio, TX, USA, 2000. Page 3.]

Learning New Patterns (Semi-Supervised)

1. Hand-crafted pattern for *Located_in* (ORGANIZATION, LOCATION):
   `ORGANIZATION [be]? located in LOCATION`
2. Apply existing patterns to data:
   `LMU is located in Germany.`
   `The University of Edinburgh is located in Scotland.`
3. Extract seed tuples: (`LMU`, `Germany`) (`University of Edinburgh`, `Scotland`)
4. Find occurrences of known tuples in data:
   `LMU is among Germany's oldest universities.`
   `The University of Edinburgh is one of Scotland's ancient universities.`
5. Generate new patterns for relation *Located_in*:
   `ORGANIZATION [be]? among LOCATION's`
   `ORGANIZATION [be]? one of LOCATION's`
6. Apply updated set of patterns to data:
   `RWTH is among Germany's Top Universities.`
   `Dundee University is one of Europe's most innovative universities.`
7. Generate new tuples: (`RWTH`, `Germany`) (`Dundee University`, `Europe`)

Rule Learning: Pros and Cons

Pros:

- More relations can be discovered
- Less human effort (when using a small amount of hand-crafted, high-quality seed patterns for bootstrapping)
- (It might even work without any hand-crafted patterns if instead some seed relations are know from an external source)

Cons:

- The set of patterns becomes more error-prone with each iteration
- Need to find best practices for generalizing the context around tuple occurrences when generating new patterns
- Extending to new relation types requires additional seed patterns, to be written manually

Supervised Machine Learning for Relation Extraction

Supervised Machine Learning

How to build a statistical classifier for relation extraction:

1. Manually annotate a corpus with relations between named entities (typically relations within individual sentences or paragraphs)
2. Divide corpus into training, development, and test sets
3. Train statistical classifier on the training set
   - The overall task can be split into subtasks with separate classifiers for each:
     (a) detecting related entity pairs, and
     (b) deciding on the relation type of a related entity pair
4. Evaluate with precision/recall/F_1

Statistical Classification of Relations

How to apply a statistical classifier for relation extraction:

1. Preprocess raw document
2. Run NER
3. Run any other auxiliary tools, such as coreference resolution, or POS tagging, dependency parsing
4. For all pairs of entities (within each sentence or paragraph): decide whether they are related or not (binary classification)
5. For related entity pairs: classify the relation type

Statistical Classification of Relations: Example

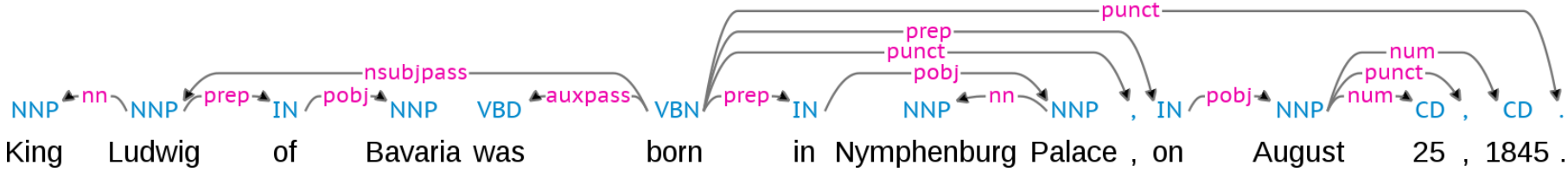
1. Preprocess

   `King Ludwig of Bavaria was born in Nymphenburg Palace , on August 25 , 1845 .`

2. Run NER (here: PERSON / DATE / LOCATION)

   `King Ludwig of Bavaria was born in Nymphenburg Palace , on August 25 , 1845 .`

3. Run auxiliary tools



4. For all pairs of entities: decide whether they are related

   (`King Ludwig of Bavaria`, `Nymphenburg Palace`) ? Related.
   (`King Ludwig of Bavaria`, `August 25 , 1845`) ? Related.
   (`Nymphenburg Palace`, `August 25 , 1845`) ? Unrelated.

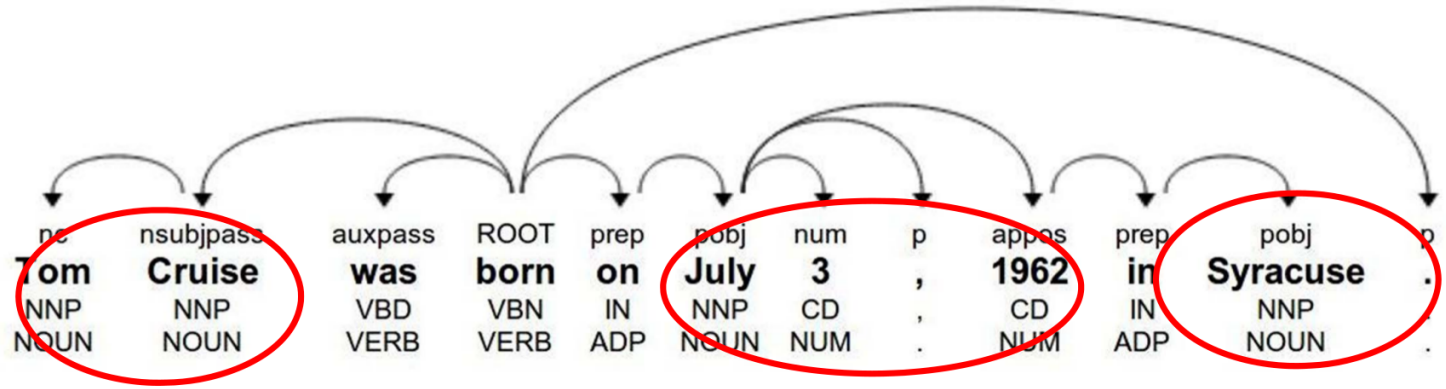
5. For related entity pairs: classify the relation type

   Born_in(`King Ludwig of Bavaria`, `Nymphenburg Palace`)
   Born_on(`King Ludwig of Bavaria`, `August 25 , 1845`)

Supervised Machine Learning: Features

Typical features for the statistical classifier:

- context words + POS
- dependency path between entities
- named entity tags
- token/parse-path/entity distance



X was born on DDDD in Y

- DEP: X <*nsubjpass* / born *prep*> on *pobj*> DATE *prep*> in *pobj*> Y
- NER: X = PER, Y = LOC
- POS: X = NOUN, NNP; Y = NOUN, NNP
- Context: born, on, in , "born on"

[Antoine Bordes and Evgeniy Gabrilovich. KDD 2014 Tutorial on Constructing and Mining Web-scale Knowledge Graphs, New York, August 24, 2014. Pages 83–84.
`http://www.cs.technion.ac.il/~gabr/publications/papers/KDD14-T2-Bordes-Gabrilovich.pdf`]

Supervised Machine Learning: Pros and Cons

Pros:

- Can get high accuracies with enough hand-labeled training data, if test similar enough to training

Cons:

- Annotating a large corpus is expensive
- Supervised models are brittle, don't generalize well to different domains (topics and genres)

Slide adapted from Dan Jurafsky [`http://www.stanford.edu/class/cs124/lec/rel.pdf`, 8 Dec. 2016. Pages 37.]

Machine Learning with Distant Supervision for Relation Extraction

Machine Learning with Distant Supervision

If you want to build a statistical classifier but there is no manually annotated training corpus:

- Relations from an existing (hand-crafted) knowledge base can be employed for corpus annotation
- Automatically create corpus annotation by labeling all cooccurrences of entity pairs that are related according to the knowledge base
  - assuming that sentences that contain a related pair are expressing the type(s) of relationship(s) that these entities have in the knowledge base
- Train classifier
- Ideally, evaluate on a small amount of gold-standard data

Distant Supervision: Pros and Cons

Pros:

- Less manual effort
- Scalability: can use large amounts of unlabeled data

Cons:

- Noisy automatic annotation of the training corpus: sentences with entity cooccurrences might not express the seed relation
- No explicit negative samples for classifier training

Conclusion and Outlook

Summary

- Relation extraction: find relations of entities in unstructured text
  - Relation types such as *Is_a*(entity_x, entity_y), *Instance_of*(entity_x, entity_y), *Has*(entity_x, entity_y), *Happened_on*(entity_x, entity_y), . . .
- Relation extraction techniques:
  pattern matching vs. statistical classifiers
  - Hand-crafted rules
  - Rule learning (relation bootstrapping)
  - Supervised machine learning
    (annotated training data + feature engineering)
- Relations can be stored in a database/knowledge graph, which can be queried in use cases such as question answering, etc.

Beyond This Presentation

Here: (mostly) limited domains, closed sets of entities and relations, relation detection within individual documents.

- *Open IE*: Can new (emerging) types of entities and relations be discovered automatically?
- *Commonsense knowledge* often not explicitly expressed in text: Include external knowledge sources.
- Cross-document references
- Temporal aspect of acquired knowledge:
  Is_Married_to(Priscilla Presley, *x*)
  Elvis? Divorced in 1973. (And yes, Elvis is dead.)

The End! Questions?

Thank you for your attention

Matthias Huck

mhuck@cis.lmu.de

Additional References

Blessing, A. and Schütze, H. (2010).

Fine-Grained Geographical Relation Extraction from Wikipedia.

In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009).

Distant supervision for relation extraction without labeled data.

In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Wiegand, M., Roth, B., Lasarcyk, E., Köser, S., and Klakow, D. (2012).

A Gold Standard for Relation Extraction in the Food Domain.

In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).