Information Extraction

MPRI 2.26.2: Web Data Management

Antoine Amarilli

Friday, January 11th

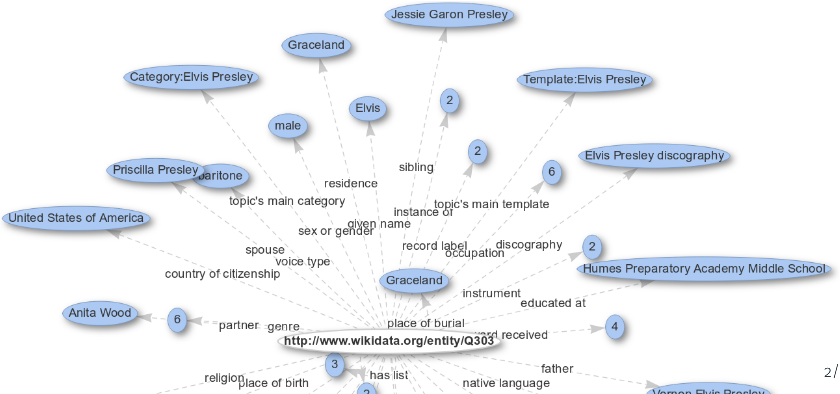
Idea of Information Extraction (IE)

- Going from unstructured text...

Elvis Presley

Elvis Aaron Presley[a] (January 8, 1935 – August 16, 1977) was an American singer and actor. Regarded as one of the most significant cultural icons of the 20th century, he is often referred to as the "**King of Rock and Roll**" or simply "**the King**".

- ... to structured data



Detour: Natural Language Processing (NLP)

- Part-of-speech (POS) tagging: annotate each word with its grammatical nature

  `The/DET text/NN is/V annotated/ADJ.`

Detour: Natural Language Processing (NLP)

- **Part-of-speech (POS) tagging:** annotate each word with its grammatical **nature**

  `The/DET text/NN is/V annotated/ADJ.`

- **Word-sense disambiguation (WSD):** choose the right **meaning**:

  `bass/1: a type of fish`

  `bass/2: a music instrument`

Detour: Natural Language Processing (NLP)

- **Part-of-speech (POS) tagging:** annotate each word with its grammatical **nature**

  `The/DET text/NN is/V annotated/ADJ.`

- **Word-sense disambiguation (WSD):** choose the right **meaning**:

  `bass/1: a type of fish`

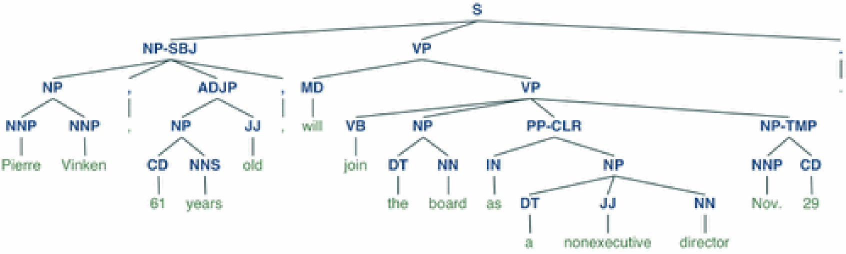
  `bass/2: a music instrument`

- **Coreference resolution:**

  `Trump told Macron that \rd{he} was not a spying t`

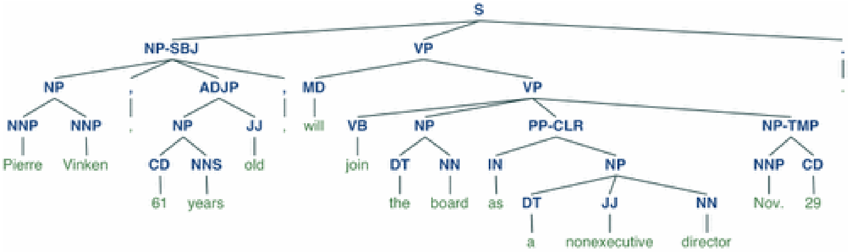
Detour: Natural Language Processing (NLP), cont'd

- Parsing: figuring out the structure of the sentence:



Detour: Natural Language Processing (NLP), cont’d

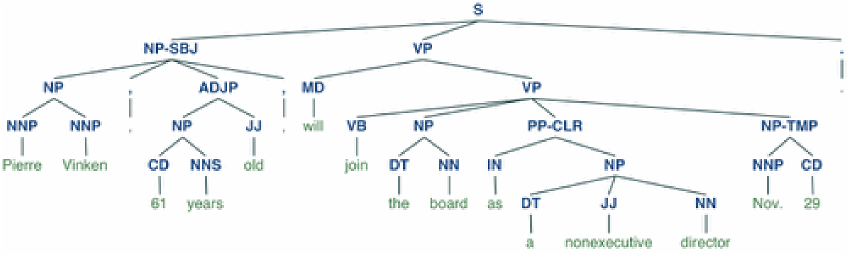
- **Parsing:** figuring out the **structure** of the sentence:



- **Preprocessing, e.g.:**
  - **Stop-word removal:** “the”, “is”, “at”
  - **Stemming:** “suffixed” $\rightarrow$ “suffix”

Detour: Natural Language Processing (NLP), cont'd

- **Parsing:** figuring out the **structure** of the sentence:



- **Preprocessing, e.g.:**
  - **Stop-word removal:** "the", "is", "at"
  - **Stemming:** "suffixed" → "suffix"

→ All these tasks are **related** to information extraction

→ But we will often try to do IE **without** solving these problems

Named Entity Recognition (NER)

- Identifying named entities in a document

  The first MPRI class for 2019 takes place at Sophie Germain.

Named Entity Recognition (NER)

- Identifying **named entities** in a document

  *The first **MPRI** class for **2019** takes place at **Sophie Germain**.*

- Possibly classify names in a simple **type hierarchy**: person, address, date, organization, etc.

Named Entity Recognition (NER)

- Identifying **named entities** in a document

  *The first **MPRI** class for **2019** takes place at **Sophie Germain**.*

- Possibly classify names in a simple **type hierarchy**: person, address, date, organization, etc.

- Difficulties:
  - **Nested entities**: *"Bank of America", "Carnegie Hall"*
  - **Boundaries**: *"All England Lawn Tennis and Croquet Club"*

Approaches for NER

- If the set is finite, use a dictionary
  - For efficient implementation, represent the dictionary as a trie and run the Aho-Corasick algorithm

Approaches for NER

- If the set is finite, use a dictionary
  - For efficient implementation, represent the dictionary as a trie and run the Aho-Corasick algorithm
- For fixed format identifiers (e.g., ISBNs, DOIs, emails, GTINs), write a regexp with captures and run an automaton

Approaches for NER

- If the set is finite, use a dictionary
  - For efficient implementation, represent the dictionary as a trie and run the Aho-Corasick algorithm
- For fixed format identifiers (e.g., ISBNs, DOIs, emails, GTINs), write a regexp with captures and run an automaton
- Otherwise, statistical approaches
  - May use various features: context, morphology, case, punctuation, part-of-speech tags, previously extracted named entities…
  - Use a pre-trained model or train it on your data

Approaches for NER

- If the set is finite, use a dictionary
  - For efficient implementation, represent the dictionary as a trie and run the Aho-Corasick algorithm
- For fixed format identifiers (e.g., ISBNs, DOIs, emails, GTINs), write a regexp with captures and run an automaton
- Otherwise, statistical approaches
  - May use various features: context, morphology, case, punctuation, part-of-speech tags, previously extracted named entities...
  - Use a pre-trained model or train it on your data
- Implemented, e.g., in Spacy, NLTK, OpenNLP, or Stanford NER
  `http://nlp.stanford.edu:8080/ner/process`

Evaluating NER systems

- We must often **evaluate** systems to compare their performance
- We do so against a **gold standard** of correct results

Evaluating NER systems

- We must often **evaluate** systems to compare their performance
- We do so against a **gold standard** of correct results
- Performance is measured along two independent **dimensions**:
  - **Precision**, the percentage of extracted matches that are correct
  - **Recall**, the percentage of correct matches that are extracted

$\rightarrow$ Extract **everything** gives 100% recall (and very bad precision)

$\rightarrow$ Extract **nothing** gives undefined precision and 0% recall

Evaluating NER systems

- We must often **evaluate** systems to compare their performance
- We do so against a **gold standard** of correct results
- Performance is measured along two independent **dimensions**:
  - **Precision**, the percentage of extracted matches that are correct
  - **Recall**, the percentage of correct matches that are extracted

  → Extract **everything** gives 100% recall (and very bad precision)

  → Extract **nothing** gives undefined precision and 0% recall
- Combining these two scores: **F1 measure**, which is the harmonic mean of precision and recall
- Or **precision-recall curve** to show the tradeoff

Evaluating NER systems

- We must often **evaluate** systems to compare their performance
- We do so against a **gold standard** of correct results
- Performance is measured along two independent **dimensions**:
  - **Precision**, the percentage of extracted matches that are correct
  - **Recall**, the percentage of correct matches that are extracted
  - → Extract **everything** gives 100% recall (and very bad precision)
  - → Extract **nothing** gives undefined precision and 0% recall
- Combining these two scores: **F1 measure**, which is the harmonic mean of precision and recall
- Or **precision-recall curve** to show the tradeoff
- To avoid **overfitting**, evaluate the system on a **validation dataset** different from the one on which the system was designed/trained

Entity Disambiguation

- Disambiguate which entity is being used
  - → *The place and function of Venus in Ovid*
  - → *Computed backscattering function of Venus and the moon*
- Usually means choosing one of several entities with that name

Entity Disambiguation

- Disambiguate **which entity** is being used
  - → *The place and function of Venus in Ovid*
  - → *Computed backscattering function of Venus and the moon*
- Usually means **choosing** one of several **entities** with that name
- Several **signals**:
  - **Prior**:
    - How **well-known** the entity is
    - How well the name **fits** the entity
    - → *She went to Paris.*
  - Similarity between the **context** of the word in the text and that of the entity in the knowledge base
  - **Consistency** with other disambiguated entities

`https://gate.d5.mpi-inf.mpg.de/webaida/`

Instance extraction

- Extracting a taxonomy with is-A relations
  - *"Pluto is a dog"*
  - *"a dog is an animal"*

Instance extraction

- Extracting a taxonomy with is-A relations
  - *"Pluto is a dog"*
  - *"a dog is an animal"*
- Hearst patterns:
  - *"Many scientists, including Einstein, believed..."*
  - *"France, Germany and other countries have been plagued with..."*
  - *"Other forms of government such as constitutional monarchy..."*

Instance extraction

- Extracting a taxonomy with is-A relations
  - *"Pluto is a dog"*
  - *"a dog is an animal"*
- Hearst patterns:
  - *"Many scientists, including Einstein, believed..."*
  - *"France, Germany and other countries have been plagued with..."*
  - *"Other forms of government such as constitutional monarchy..."*
- Set expansion:
  - Start with a set of entities of the same type (e.g., countries),
  - Find a list or table column containing several such entities
  - Add the other entities (assume that they have the same type)

Instance extraction

- Extracting a taxonomy with is-A relations
  - *"Pluto is a dog"*
  - *"a dog is an animal"*
- Hearst patterns:
  - *"Many **scientists, including Einstein**, believed..."*
  - *"**France, Germany and other countries** have been plagued with..."*
  - *"Other **forms of government such as constitutional monarchy**..."*
- Set expansion:
  - Start with a set of entities of the same type (e.g., countries),
  - Find a list or table column containing several such entities
  - Add the other entities (assume that they have the same type)
- Problems
  - False positives: *"the classification of such cities as urban"*
  - Boundaries: *"some scientists, such as computer scientists"*
  - Disambiguation, and semantic drift

Instance extraction

- Extracting a **taxonomy** with **is-A** relations
  - *"Pluto is a dog"*
  - *"a dog is an animal"*
- **Hearst patterns**:
  - *"Many **scientists, including Einstein**, believed..."*
  - *"**France, Germany and other countries** have been plagued with..."*
  - *"Other **forms of government such as constitutional monarchy**..."*
- **Set expansion**:
  - Start with a set of entities of the same type (e.g., countries),
  - Find a **list** or **table column** containing several such entities
  - **Add** the other entities (assume that they have the same type)
- **Problems**
  - **False positives:** *"the classification of such cities as urban"*
  - **Boundaries:** *"some scientists, such as computer scientists"*
  - **Disambiguation**, and **semantic drift**
- **Taxonomy induction:** cleaning up the resulting taxonomy

Example: **NELL** `http://rtw.ml.cmu.edu/rtw/kbbrowser/pred:bird`

Fact extraction

- Generalization of Hearst patterns, e.g.: "X was born in Y"

Fact extraction

- Generalization of Hearst patterns, e.g.: “X was born in Y”
- DIPRE: Dual Iterative Pattern Relation Expansion:
  - Apply the patterns to generate more facts
  - Use the new facts to learn more patterns

  → Problems: semantic drift; sometimes multiple relations match…

Fact extraction

- Generalization of Hearst patterns, e.g.: “X was born in Y”
- DIPRE: Dual Iterative Pattern Relation Expansion:
  - Apply the patterns to generate more facts
  - Use the new facts to learn more patterns

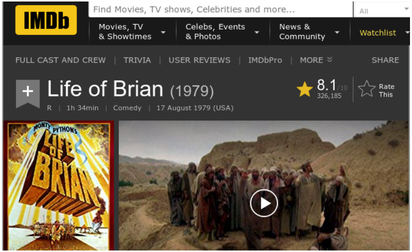
  $\rightarrow$ Problems: semantic drift; sometimes multiple relations match…
- Learning from structured Web content, e.g., lists and tables (cf Web Data Commons), Wikipedia infoboxes…

Fact extraction

- Generalization of **Hearst patterns**, e.g.: "X was born in Y"
- DIPRE: Dual Iterative Pattern Relation Expansion:
    - Apply the **patterns** to generate more **facts**
    - Use the new **facts** to learn more **patterns**

  $\rightarrow$ **Problems:** semantic drift; sometimes multiple relations match...
- Learning from **structured Web content**, e.g., lists and tables (cf Web Data Commons), Wikipedia infoboxes...
- General technique: **Wrapper induction** (see next slide)

Wrapper induction



Wrapper induction

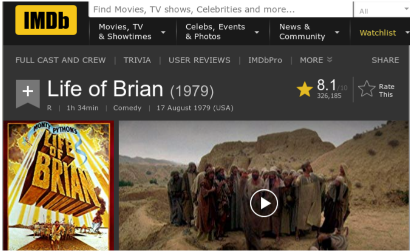

```
        <div class="title_wrapper">
<h1 class="">Monty Python and the Holy Grail <span id='
>1975</a>)</span>            </h1>
            <div class="subtext">
                PG
    <span class="ghost">|</span>                    <time da
                    1h 31min
                </time>
    <span class="ghost">|</span>
<a href="/search/title?genres=adventure&explore=title_type,(
>Adventure</a>,
<a href="/search/title?genres=comedy&explore=title_type,genr
>Comedy</a>,
<a href="/search/title?genres=fantasy&explore=title_type,ger
>Fantasy</a>
    <span class="ghost">|</span>
<a href="/title/tt0071853/releaseinfo?ref_=tt_ov_inf"
title="See more release dates" >3 December 1975 (France)
```


```
        <div class="title_wrapper">
<h1 class="">Life of Brian <span id="titleYear">(<a hre
>1979</a>)</span>            </h1>
            <div class="subtext">
                R
    <span class="ghost">|</span>                    <time da
                    1h 34min
                </time>
    <span class="ghost">|</span>
<a href="/search/title?genres=comedy&explore=title_type,genr
>Comedy</a>

    <span class="ghost">|</span>
<a href="/title/tt0079470/releaseinfo?ref_=tt_ov_inf"
title="See more release dates" >17 August 1979 (USA)
</a>            </div>
        </div>
    </div>
```

Wrapper induction


```
      <div class="title_wrapper">
<h1 class="">Monty Python and the Holy Grail <span id="
>1975</a>)</span>             </h1>
        <div class="subtext">
             PG
    <span class="ghost">|</span>                    <time da
                    1h 31min
                </time>
    <span class="ghost">|</span>
<a href="/search/title?genres=adventure&explore=title_type,g
>Adventure</a>,
<a href="/search/title?genres=comedy&explore=title_type,genr
>Comedy</a>,
<a href="/search/title?genres=fantasy&explore=title_type,ger
>Fantasy</a>
    <span class="ghost">|</span>
<a href="/title/tt0071853/releaseinfo?ref_=tt_ov_inf"
title="See more release dates" >3 December 1975 (France)
```

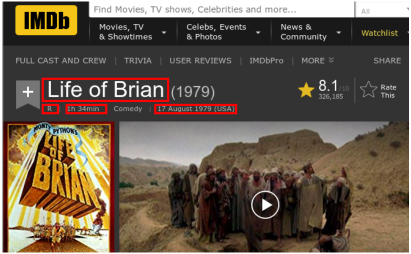



```
      <div class="title_wrapper">
<h1 class="">Life of Brian <span id="titleYear">(<a hre
>1979</a>)</span>             </h1>
        <div class="subtext">
             R
    <span class="ghost">|</span>                    <time da
                    1h 34min
                </time>
    <span class="ghost">|</span>
<a href="/search/title?genres=comedy&explore=title_type,genr
>Comedy</a>

    <span class="ghost">|</span>
<a href="/title/tt0079470/releaseinfo?ref_=tt_ov_inf"
title="See more release dates" >17 August 1979 (USA)
</a>              </div>
      </div>
    </div>
```

Wrapper induction


```
        <div class="title_wrapper">
<h1 class="">Monty Python and the Holy Grail <span id='
>1975</a>)</span>            </h1>
            <div class="subtext">
                PG
    <span class="ghost">|</span>                    <time da
                    1h 31min
                </time>
    <span class="ghost">|</span>
<a href="/search/title?genres=adventure&explore=title_type,g
>Adventure</a>,
<a href="/search/title?genres=comedy&explore=title_type,gen
>Comedy</a>,
<a href="/search/title?genres=fantasy&explore=title_type,ger
>Fantasy</a>
    <span class="ghost">|</span>
<a href="/title/tt0071853/releaseinfo?ref_=tt_ov_inf"
title="See more release dates" >3 December 1975 (France)
```




```
        <div class="title_wrapper">
<h1 class="">Life of Brian <span id="titleYear">(<a hre
>1979</a>)</span>            </h1>
            <div class="subtext">
                R
    <span class="ghost">|</span>                    <time da
                    1h 34min
                </time>
    <span class="ghost">|</span>
<a href="/search/title?genres=comedy&explore=title_type,gen
>Comedy</a>

    <span class="ghost">|</span>
<a href="/title/tt0079470/releaseinfo?ref_=tt_ov_inf"
title="See more release dates" >17 August 1979 (USA)
</a>            </div>
        </div>
    </div>
```

Slide credits

- Course structure inspired by the class by Fabian Suchanek `https://suchanek.name/work/teaching/inf344-2018/index.html`
- Slide 4: `https://www.nltk.org/_images/tree.gif`