

Exercise #9

Introduction to Knowledge Management, WS2016

Camilo Thorne

(based on slides by A. Melo)

Mannheim, 09.05.2016

Named Entity Recognition (NER)

In the following text find all **Named Entities** and annotate them. Write down the complete Named Entity, the line where you found it and its type (**Person**, **Location** or **Organization**):

1| Justin Drew Bieber (born March 1, 1994) is a Canadian pop musician,
2| actor, and singer-songwriter. Bieber was discovered in 2008 by American
3| talent manager Scooter Braun, who came across Bieber's videos on YouTube
4| and later became his manager. Braun arranged for him to meet with
5| entertainer Usher Raymond in Atlanta, Georgia, and Bieber was signed to
6| Raymond Braun Media Group (RBMG), and then to an Island Records
7| recording contract offered by record executive L.A. Reid.

Named Entity Recognition (NER)

Persons

Named Entity Recognition (NER)

- ▶ Line 1: Justin Drew Bieber
- ▶ Line 2: Bieber
- ▶ Line 3: Scooter Braun
- ▶ Line 3: Bieber
- ▶ Line 4: Braun
- ▶ Line 5: Usher Raymond
- ▶ Line 5: Bieber
- ▶ Line 7: L.A. Reid

Named Entity Recognition (NER)

Locations

Named Entity Recognition (NER)

- ▶ Line 5: Atlanta
- ▶ Line 5: Georgia

Named Entity Recognition (NER)

Organizations

Named Entity Recognition (NER)

- ▶ Line 3: Youtube
- ▶ Line 6: Raymond Braun Media Group (RBMG)
- ▶ Line 7: Island Records

Automated NER

Download the documents attached to this exercise from ILIAS, open the document **justin.txt** as a corpus and run the ANNIE pipeline on it.

Compare the resulting annotations of Named Entities with the ones created by you:

- ▶ What are the differences in the annotations?
- ▶ Which types of errors occur in the automatic annotation?

Automated NER

Differences/errors:

- ▶ Line 4: Braun not recognized
- ▶ Line 5: Bieber not recognized as First Name
- ▶ ...

Information Extraction (IE)

Phase: Birthdate

Input: Person Organization Unknown Token

Options: control = appelt

Rule: Birthdate

```
(
  ({Person}):person
  {Token.string == "("}
  ({Token.string == "born"})
  ({Token}):month
  ({Token.kind == "number"}):day
  ({Token.kind == "punctuation"}) //the comma after the day
  ({Token.kind == "number"}):year
  {Token.string == ")"})
-->
:person.Birthdate = {rule = "Birthdate",
  birthyear = :year.Token.string,
  birthmonth = :month.Token.string,
  birthday = :day.Token.string}
```

Information Extraction (IE)

1. Which files contain the annotation of type **Birthday**?
2. For one of the files report the complete information of the annotation (type, start, end, and features).¹

¹Available from the *Annotation List*-view

Information Extraction (IE)

| | |
|---------------|--------------------------------|
| grohl.txt | Birthdate annotation available |
| justin.txt | Birthdate annotation available |
| madonna.txt | Birthdate annotation available |
| metallica.txt | No birthdate annotation |
| nirvana.txt | No birthdate annotation |

| Filename | Type | Start | End | Id | Features |
|-------------|-----------|-------|-----|-----|--|
| justin.txt | Birthdate | 0 | 18 | 484 | birthday=1, birthmonth=March, birthyear=1994, rule=Birthdate |
| madonna.txt | Birthdate | 0 | 22 | 997 | birthday=16, birthmonth=August, birthyear=1956, rule=Birthdate |

Adding Information with the `IsA` Relation

```
01| Phase: IsA
02| Input: Person Organization Unknown Token
03| Options: control = appelt
04|
05| Rule: IsA
06| (
07| ({Person} | {Organization} | {Unknown}):entity
08| ({!Person, !Organization, !Unknown})*
09| ({Token.string == "is"})
10| ({Token.string == "a"} | {Token.string == "an"})
11| (({Token.category == "JJ"})*({Token.category == "NN"})+):isa
12| )
13| -->
14| :entity.IsA = {rule = "IsA", IsA = :isa@string}
```

Adding Information with the `IsA` Relation

1. What does the operator `|` (“bar”) do in line 7? Give two examples of a match for the pattern in line 7.
2. What do the operators `!` and `*` (“kleene star”) do in line 8? Describe in your own words what is matched by the complete pattern in this line.
3. What does the operator `+` do in comparison to `*`?
4. Which kind information is provided by `Token.category`? Give two examples of a match for the pattern in line 11.
5. Which information is accessed by the meta property `@string`? What happens to the `isA` annotations if you change line 14 to the following:

```
:entity.IsA = {rule = "IsA", IsA = :isa.Token.string}
```

6. Give a short description what is matched by the complete `IsA` rule.
7. How could Watson take advantage of this information?

Adding Information with the `isA` Relation

1. What does the operator `|` (“bar”) do in line 7? Give two examples of a match for the pattern in line 7.

Answer

- ▶ The vertical bar `|` is used to denote alternatives. Means it would match either a token whose category is **Person** or one whose category is **Organisation** or **Unknown**

| Filename | Type | Start | End | Id | Features |
|--------------|------|-------|-----|-----|---|
| metalica.txt | isA | 0 | 9 | 866 | isA=American heavy metal band, rule=isA |
| grohl.txt | isA | 18 | 23 | 822 | isA=American rock musician, rule=isA |

Adding Information with the ISA Relation

2. What do the operators `!` and `*` (“kleene star”) do in line 8? Describe in your own words what is matched by the complete pattern in this line:

```
08 | ({!Person, !Organization, !Unknown})*
```

Answer

- ▶ The `!` operator is a negative constraint. It is equivalent to the `!` in Java and to the `NOT` statement in SQL.
- ▶ The `*` operator means that there can be zero or more matches

Adding Information with the `ISA` Relation

3. What does the operator `+` do in comparison to `*`?

Answer

- ▶ the `+` operator matches 1 or more occurrences.
- ▶ the `*` operator matches 0 or more occurrences.
- ▶ In the example `(Token.category == "NN")+` the `+` indicates that one or more matches are necessary for the rule

Adding Information with the `ISA` Relation

4. Which kind information is provided by `Token.category`? Give two examples of a match for the pattern in line 11.

```
11 | ((Token.category == "JJ") * (Token.category ==  
"NN")) : isa
```

Answer

- ▶ `Token.category` provides the POS (= Part-Of-Speech) tag
- ▶ One example of match is American heavy metal band from **Metallica.txt**
- ▶ Another example is American rock musician in **Grohl.txt**

Adding Information with the `IsA` Relation

5. Which information is accessed by the meta property `@string`? What happens to the `isA` annotations if you change line 14 to the following:

```
:entity.IsA = {rule = "IsA", IsA = :isa.Token.string}
```

Answer

- ▶ is a meta-property which returns the whole string spanned by the annotation
- ▶ **Change to:**

```
:entity.IsA = {rule = "IsA", IsA = :isa.Token.string}
isA=American heavy metal band, {rule = "IsA", isA = "American heavy", rule=isA}
isA=American rock musician, {rule = "IsA", isA = "American rock", rule=isA}
```

Adding Information with the `isA` Relation

6. Give a short description what is matched by the complete `isA` rule.

Answer

- ▶ The complete `isA` rule matches a **Person**, an **Organization** or **Unknown** annotation with a

$(\textit{Adjective})^*(\textit{Noun})^+$

“pattern” if there is a combination of “is a” or “is an” between the person and the object

- ▶ there is no other information between the P/O/U annotation and the “is a” or “is an” tokens

Adding Information with the ISA Relation

7. How could Watson take advantage of this information?

Discussion

- ▶ e.g. answer questions for bands singers
- ▶ Ferrucci (2010) shows that **group**, **singer**, **actor**, **actress**, **musician** are frequent answer types
- ▶ ...

Debugging

1. Why can't the rule find an `IsA` relation for Nirvana? Fix the rule in order to find the relation.
2. Why can't the rule find the relation `IsA - singer-songwriter` for Madonna?² Why would it not be a good idea to change the **rule** in order to find the relation for Madonna. How should this error be fixed instead?
3. Why is there no birthday annotation for "David Eric "Dave" Grohl"?

²HINT: Have a look at the `Token` annotations in the text.

Debugging

1. Why can't the rule find an `IS_A` relation for Nirvana? Fix the rule in order to find the relation.

Answer

- ▶ For Nirvana nothing could be found because it uses “was an” between the person annotation and the category. Fix could look like that:

```
(Token.string == "is" | Token.string == "was")  
(Token.string == "a" | Token.string == "an")
```


Debugging

- .2 Why can't the rule find the relation `IsA - singer-songwriter` for Madonna? Why would it not be a good idea to change the **rule** in order to find the relation for Madonna. How should this error be fixed instead?

Answer

"Madonna Louise Ciccone (born August 16, 1958) is an American singer-songwriter, actress, director, dancer, and entrepreneur"

- ▶ Singer-songwriter rule has `category='JJ'` = of wrong tag type
- ▶ Support “,” and “and” to extract all the nouns

Debugging

3. Why is there no **Birthdate** annotation for “David Eric ”Dave” Grohl”?

Answer

- ▶ There is no **Birthdate** annotation for David Eric ”Dave” Grohl because of the ”Dave” in the name
- ▶ If you delete the ”Dave” the annotation will be detected with the rule

Persons with a Nickname (Bonus)

The screenshot shows a web-based NER interface. At the top, the text "David Eric 'Dave' Grohl (born January 14, 1969) is a" is displayed. The word "David" is highlighted in green. Below the text is a toolbar with navigation arrows, a green checkmark, a red 'X', and a pin icon. A dropdown menu is open, showing the entity type "Person". Below this, there is a table for defining properties:

| | | | |
|---|----------|----------|---|
| C | nickname | Dave | X |
| C | rule | Nickname | X |
| C | | | X |

At the bottom of the interface, there is a button labeled "Open Search & Annotate tool".

Figure : **Nickname** annotation.

Write a new rule that is able to match “David Eric ”Dave” Grohl”, annotate the match with the annotation type `Person` and add the property `nickname` with the value “Dave”.

Persons with a Nickname (Bonus)

Phase: Nickname

Input: Person Organization Unknown Token

Options: control = appelt

Rule: Nickname

```
(
  {Person}
  {Token.string == "\""} //the \" is an escaped sign
  ({Token})+:nick
  {Token.string == "\""}
  {Unknown}

):person
-->
:person.Person = {rule="Nickname", nickname = :nick@string}
```

Watson

1. What is the definition of lexical answer types (**LAT**)? You may cite the paper or use your own words to answer this question.
2. Why is extracting the **LAT** of a question important to answer it?
3. How does NER relate to **LAT**?
4. What are or could be practical applications of a system like Watson?

Watson

1. What is the definition of lexical answer types (**LAT**)? You may cite the paper or use your own words to answer this question.

Answer

A lexical answer type (**LAT**) is a word or noun phrase in the question that specifies the type of the answer without any attempt to understand its semantics.

e.g.: “Who is the 42nd president of the United States?” **LAT** = “president”

Watson

2. Why is extracting the **LAT** of a question important to answer it?

Answer

- ▶ Determining whether or not a candidate answer can be considered an instance of the **LAT** is an important kind of scoring and a common source of critical errors
- ▶ Helps to answer the question if you know the domain where the actual answer can be found

Watson

3. How does NER relate to **LAT**?

Answer

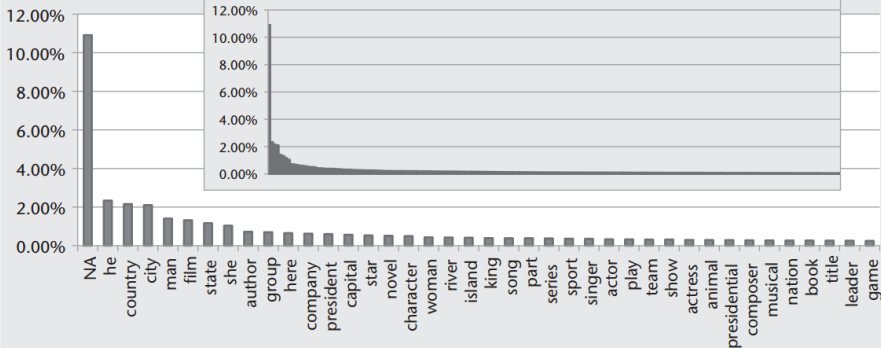
- ▶ Figure 1 on the paper from Ferucci (2010) shows the most frequent lexical answer types that have been found (**NA**, **he**, **country**, **city**, **man**, **film**, **state**,...)
- ▶ Many of the frequent lexical answer types can be related to the actual NER types
- ▶ e.g **country**, **state**, **city** = **Location**

Watson

3. How does NER relate to **LAT**?

Lexical Answer Type Frequency, Ferucci (2010)

40 Most Frequent LATs



Watson

4. What are or could be practical applications of a system like Watson?

Discussion

- ▶ Decision support systems
- ▶ Expert search
- ▶ ...
- ▶ Check this [BBC report](#) about it



Thank You!