Text Mining for Historical Documents (WS 2009/10)

Named Entity Recognition

Based on:

David Nadeau, Satoshi Sekine: „A survey of named entity recognition and classification“

Kate Byrne: „Nested Named Entity Recognition in Historical Archive Text“

What is a Named Entity?

A **Named Entity** is an entitiy referred to by a **rigid designator**.
Rigid means that the designator always refers to the same entity.

For example:

Bill Gates, Microsoft, Saarbrücken, Penicillin are Named Entities.

This building, he, the tree over there are not, because they can stand for different entities, even within the same text.

What about Dates?

- usually temporal (and some numerical) expressions are included
- some are good examples for NEs, like *the year 2010*
- some are not really NEs, like *June* (could mean *next June, last June, June of the year 1900...*)
- Definition of NEs is often loosened to allow those cases

Types of NEs

In addition to recognizing them, it is also useful to label NEs with **types**.

The types most often used are *person*, *location* and *organization*. (These are also known as **enamex**.)

Other frequent types are sub-categories such as *city, state, country* (sub-categories to *location*) or *politician, entertainer* (sub-categories to *person*).

More special types can be used if necessary, for example *email-address, research area* or *protein*.

Usually only a small number of types is used in a system, but there are exceptions.

Nested NEs

NEs can be **nested** within one another.

Example:

Edinburgh University is an NE of the type *organization* and also contains an NE of the type *location*.

How is it done?

There are basically two different ways to find NEs:

Handcrafted rules and rules generated by **machine learning**.

Handcrafted rules were used in the first systems (around 1991), **machine learning** became more popular over time and is now used most often.

There are three types of **machine learning**:

supervised, **semi-supervised** and **unsupervised** learning

Supervised Learning

- most common method today
- needs annotated training corpus to derive rules from
- performance strongly influenced by choice of features used to create rules
- performance greatly decreases if a system is used on a different domain than it is trained on
- creating the training corpus needs human work, therefore expensive

Semi-supervised Learning

- doesn't need annotated corpus
- instead uses examples for NEs, called 'seeds'
- seeds are searched in a corpus, then contextual information is derived from them
- contextual information is then used to find different words in similar contexts, considered to be NEs of the same type
- the process is then repeated multiple times

Unsupervised Learning

- many different approaches
- might use external ressources such as WordNet
- might use simple heuristics, for example if a type is followed by the phrase „such as“, the next word will probably a NE of this type (*„countries such as Germany“*)
- might also use frequencies of words (NEs usually appear in „bursts“ in news articles)

Features

- characteristic attributes of words/phrases
- the more features two words share, the more likely they are to have the same type
- choice of features is important for a system's performance
- there are three types of features:
  - word-level features
  - list lookup features
  - document and corpus features

Word-Level Features

Word-Level features can be things like:

- Case (word starts with a capital letter, is all uppercased...)
- Punctuation (word has an internal period, apostrophe...)
- Digit (word contains digits)
- Morphology (prefixes, suffixes)
- Part-of-speech (verb, noun, foreign word...)

List Lookup Features

For each word it is checked if it appears on a given list or not.

The easiest form of a list lookup feature is a dictionary. If a word appears in a dictionary, it is most likely not a NE.

Other lists can also be used, for example lists with words that appear often in organization names, like *associates, inc, corp...*

Lookup Techniques

- **exact match:** easiest way (word is on the list or not), often too strict
- **stemming** or **lemmatizing:** words are stripped of affixes
- **edit-distance:** if a word is similar enough to one on the list, it counts
- **Soundex algorithm:** words are compared by how they sound rather than how they are spelled

Document and Corpus Features

- multiple occurrences (e.g. uppercased and lowercased)
- local syntax (position in sentence, paragraph, document...)
- meta information ()
- corpus frequency

Evaluation

- necessary to measure improvement
- many possible methods, some of them are:
  - Exact Match Evaluation
  - MUC Evaluation
  - ACE Evaluation

Exact-Match Evaluation

- only NEs whose type and boundaries are recognized correctly are counted
- systems are compared using the F-Score (or F-Measure)
- doesn't take into account that partially recognized NEs can be useful already, for a query in information retrieval for example it can be enough to find a NE in a sentence, its exact boundaries are not required

MUC Evaluation

- a systems performance is measured on two axes:
  - how many NEs get recognized with correct boundaries
  - how many NEs get recognized with correct type

- this allows better comparison between systems, considering different purposes

ACE Evaluation

- each entity type has its own worth, for example a correct NE of the type *person* might be worth as much as two NEs of the type *organization*
- this allows two compensate for frequency effects (rare types are harder to detect, giving them a high value rewards systems who can find them