List-based Named Entity Recognition and evaluation of NER algorithms

Matthias Jacob, Max Plauth

4. Dezember 2007
Agenda

- List-based Named Entity Recognition
  - what are the challenges
  - how does it work?
  - examples: the role of thesauri

- Extending list-based NER
  - Introduction of possible approaches
  - Part-Of-Speech tagging

- Performance measurement
  - how does performance measurement work
  - how well do we perform right now?
What is named entity recognition?

- extraction of information relevant to a certain purpose from unstructured textual sources
  - textual sources are available in natural language

- a named entity is a concrete characteristic of a concept

- 'Scarlett Johansson' is a textual representation of a named entity
  - but 'Scarlett Johansson' is not a named entity itself
  - 'Scarlett' or 'Johansson' are representations of the same Entity due to the concept used for ProminentPeople.info
Challenges coping with German texts

- not only names but all nouns are capitalized
  - homonyms become a major problem
  - „Mark“ as a common German name in contrast to
  - „Mark“ as the former German currency
  - large reliable lexical resources necessary for disambiguation

- sentence-structure with partially free word order
  - e.g. finite verbs may occur at three different positions:
    - „I liked the movie very much.“ in contrast to
    - „Mir hat der Film sehr gut gefallen.“
List-based NER (1/4)

- central component is a large list of words or phrases based on which we can identify named entities
  - as many morphological variants as possible are needed
  - an extremely large thesaurus is a prerequisite

- tokens matching the pattern of the desired characteristic are being identified as entities

- luckily enough, our domain is much more simple
  - we only have to look for names
  - therefore, our thesaurus can be much smaller
what if our thesaurus was incomplete?

identified entities:
- David Hasselhoff
- Franz Beckenbauer

Problem: entity „Frank Siering“ unrecognized!

<table>
<thead>
<tr>
<th>ID</th>
<th>FNAME</th>
<th>LNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>70776</td>
<td>Stefan</td>
<td>Beckenbauer</td>
</tr>
<tr>
<td>41689</td>
<td>Franz</td>
<td>Beckenbauer</td>
</tr>
<tr>
<td>50673</td>
<td>David Michael</td>
<td>Hasselhoff</td>
</tr>
</tbody>
</table>
what about the quality of textual sources and thesauri?

article implicates:
first name: 'Mario'
last name: 'Gomez'

thesaurus says:
first name: 'Mario Gomez'
last name: 'García'

we know:
first name: 'Mario'
last name: 'Gómez García'

<table>
<thead>
<tr>
<th>ID</th>
<th>FNAME</th>
<th>LNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>57296</td>
<td>Omar Gomez</td>
<td>Rey</td>
</tr>
<tr>
<td>84152</td>
<td>Ignacio Gomez</td>
<td>Novo</td>
</tr>
<tr>
<td>90043</td>
<td>Mario Gomez</td>
<td>García</td>
</tr>
</tbody>
</table>
List-based NER (4/4)

- **advantages** (*PPeople*)
  - enables good performance for our means
  - small thesaurus compared to many other applications

- **advantages** (*general purpose applications*)
  - negative lists deliver decent results with manageable effort

- **weaknesses**
  - insensitive to context, disambiguation of homonyms impossible
  - heavily depends on the quality of used data
  - enormous thesaurus (incl. misspellings, morphologies etc.)
How to eliminate such weaknesses?

- incompleteness of thesauri
  - introduce a logic that abstracts from the underlying text in order to find named entities not in the list

- context unregarded
  - inspect the environment of a word to explore its meaning and to discover formerly unrecognised entities

- misspellings and morphologies
  - work out an algorithm which generates normalised spellings
  - stem words to a root

- time-killing list look-ups
  - filter out words which can easily be identified as non-entities
Part-of-speech tagging (1/2)

- gathers information about the syntactical structure of a sentence
- does not decide whether a word belongs to an entity, but pre-processes the text

procedure:

- tokenisation (level: word boundaries)
- sentence segmentation (level: parts of sentences, sentence boundaries)
- tag each word with information about its part of speech (POS) depending on the grammatical understanding of the sentence

- POS tagging provides a rough set of potential named entities
- refine the choice by applying grammar rules
Part-of-speech tagging (2/2)

„Herr Beckenbauer ging nach Stuttgart.“

- **PER-NP** → ’Herr’ PROPER-NAME (PROPER-NAME)?
- **LOC-P** → DERIV(‘gehen’) LOC-PP
- **LOC-PP** → ’nach’ PROPER-NAME
- **LOC-PP** → ’von’ PROPER-NAME ’nach’ PROPER-NAME
Conclusion

- pure list-based NER might not seem too powerful at first sight
- on second sight, some little extensions can deliver us good results
- even for concepts with higher complexity, teamed up with some elaborate logic list-based NER can give us great results
- but there is one more thing:
  - how well do we perform exactly?
How well do we perform?

- We need indicators for performance measurement which are
  - common and comparable among projects
  - applicable to different NER domains
  - preferably easy to calculate and to understand

- There are three measurements:
  - precision
  - recall
  - F-measure
Precision & Recall

- **Precision**
  - number of correct positive predictions compared to the number of total positive predictions
  
  \[
  \text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}
  \]

- **Recall**
  - number of positive predictions compared to the actual number of named entities existing in the textual source

  \[
  \text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}
  \]
F-measure classes (1/2)

- F-measure combines precision and recall
- aim: a score to evaluate the overall quality of the NER method
- standard definition:
  - unweighted harmonic mean of precision and recall ($F_1$)
  
  $$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$
F-measure classes (2/2)

- more general formula (for non-negative, real $\alpha$):

$$F_\alpha = \frac{(1 + \alpha) \cdot \text{precision} \cdot \text{recall}}{\alpha \cdot \text{precision} + \text{recall}}$$

- allows weighting towards one of the two characteristics
  - $F_{0.5}$ weights precision twice much as recall
  - $F_2$ weights recall twice much as precision
F-measure: What is best for us?

- F-measure: weighting might fit best for our problem?
Our tool suite: EvaFill

- first of all, news need to be parsed 'by hand' using EvaFill

- it picks a random article
- you have to enter all natural persons
- schema: fname$fname;
- Scarlett$Johansson;
- Scar$.Johansson;
- don’t worry, it will be improved soon
Our tool suite: EvaP’s internals

- EvaFill fills a table (NID, TEXTPOSITION, FNAME, LNAME)

- EvaP determines the multisets TP, FP and NP as follows
  - TruePositives = EVAP_SOURCE ∩ GROUP.RESULTS
  - FalsePositives = GROUP.RESULTS \ EVAP_SOURCE
  - FalseNegatives = EVAP_SOURCE \ GROUP.RESULTS

- from there, it is easy to use the formulas we already discussed

- that’s it already, but it caused more work than it looks like ;(
Thank you for your attention!

- Literature used:
  - Text Mining. Predictive Methods for Analyzing Unstructured Information (Sholom M. Weiss, Nitin Indurkhya, T. Zhang)
  - Corpus-based Learning of Lexical Resources for German Named Entity Recognition (Marc Rössler et al)
  - Named Entity Recognition without Gazetteers (Mikheev et al)
  - The Difficulties of Taxonomic Name Extraction and a Solution (Guido Sautter and Klemens Böhm)
  - Die Transformation von Text in Vektoren (Julian Forster)
  - Named Entity Recognition (Joel Lang)