Programming exercise: A machine-learning algorithm for co-reference resolution

Ina Rösiger, Arndt Riester

Universität Stuttgart
Institut für maschinelle Sprachverarbeitung

Preamble

- This programming exercise is based on the paper by Soon et al. (2001): A Machine Learning Approach to Coreference Resolution of Noun Phrases
- It is meant as a chance for you to better understand the different steps that are involved in co-reference resolution, not as a coding competition
- You can code in Python or Java (please comment a lot!)
- You can do the exercise alone or in a team of two. If you choose to do it in a team, please specify who is responsible for which sections. Please do not copy code written by other teams.
- Note that we are not working with the same corpus as in the original paper. Some details that are given in the paper (e.g. on markables, etc.) may not be applicable in our case.
- Solutions for step 1 (markable extraction) and step 2 (feature extraction) will have to be handed in by everyone and will not be graded
- Work on further steps only if you choose the programming exercise for your final grade
- For those doing the whole programming exercise:

- Please document the steps you implemented and the results you achieved in a separate, short text document, giving details on challenges and problems you came across as well as possible solutions (2-3 pages)
- Not every step in the implementation has to be perfect
  if you encounter difficulties, just describe them
  the mark you'll get will be based on both implementation and presentation of results
- You might have to present your ideas/difficulties/solutions in a very short talk (depending on how much time we've got left)
- Steps marked with *bonus* are optional (you should do them after your pipeline is finished when there is time left ...)
- When you hand in the exercise: please, create a README that gives instructions on how to use your script(s) –commands, arguments, etc. Please create a folder *output* that contains the output of every step (the required outputs for every step are described in the different sections). We should be able to use your scripts on IMS linux machines in order to check and compare results.
- **Deadline: to be specified**

1 Determination of Markables

Aim: Implement a module that extracts markables
Output: List of markables for every document

- Download the corpus from ILIAS: the corpus contains coreference clusters (in the last column) as well as gold annotations (parses, POS tags, named entities, etc.)
- Have a look at the corpus format and familiarise yourself with the annotation levels
- Markable extraction: have a look at the corpus to see which entities have been marked as coreferent. Please document which entities you extract from which annotation level. Typically this involves:
  - Parses: to extract noun phrases
  - POS tags: to extract personal and possessive pronouns (only if not already marked as NP in the parse: you should check this)

  - Named entity recognition: to extract organisation, person, location, date, time, money, and percent entities

- Nested noun phrase extraction: check if needed at all, depends on the parser and the gold annotations.
  In Soon et al. 2001 they added these nested noun phrases:
  his long-range strategy and **Eastern's** parent,
  wage reductions, **Union** representatives added as markables

- Think about which information you want to add to your list of markables: you will have to extract further information on your markables in the following steps, so you should be able to find your markable again in the CoNLL document

- *Bonus*: Evaluation of markables: can you compute recall and precision for your markable extraction module?