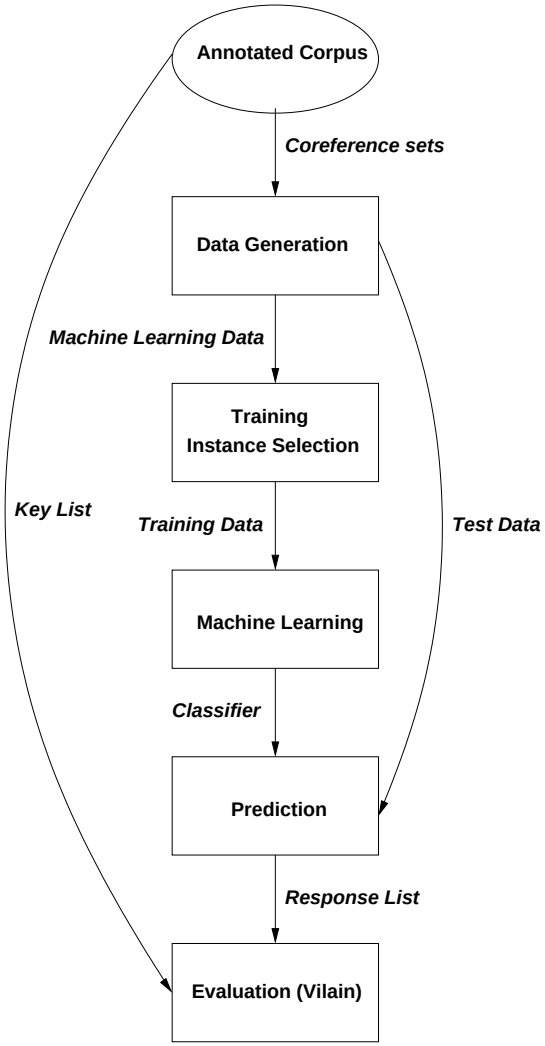
What are the annotated data good for?

- research/development system
  - data generation
  - machine learning (or any other kind of algorithm)
  - evaluation
- production/real world system

Experimental Setup for Coreference Resolution



Data Generation

- formulate your research question in such a way that it is amenable to a machine learning algorithm
- transform your hypotheses (or the concepts you are used to) into features
- do some post-processing on the annotations (e.g. for coreference: look up WordNet for world knowledge, computation of distance between anaphor and antecedent, computation of string match values, edit distance, . . .)
- convert annotations into format which the machine learning algorithm of your choice may be able to understand (in most cases some kind of comma-separated list, could be XML in the future)

Features for Coreference Resolution

- NP-level features describing properties of a particular NP;
- coreference-level features describing properties of the relation between potential anaphor and potential antecedent.

NP-level Features

1.	ante_gram_func	grammatical function of antecedent
2.	ante_npform	form of antecedent
3.	ante_agree	person, gender, number
4.	ante_case	grammatical case of antecedent
5.	ante_s_depth	the level of embedding in a sentence
6.	ana_gram_func	grammatical function of anaphor
7.	ana_npform	form of anaphor
8.	ana_agree	person, gender, number
9.	ana_case	grammatical case of anaphor
10.	ana_s_depth	the level of embedding in a sentence

Coreference-level Features

11.	agree_comp	compatibility in agreement between anaphor and antecedent
12.	npform_comp	compatibilty in NP form between anaphor and antecedent
13.	wdist	distance between anaphor and antecedent in words
14.	mdist	distance between anaphor and antecedent in markables
15.	sdist	distance between anaphor and antecedent in sentences
16.	syn_par	anaphor and antecedent have the same grammatical function

Transformation

- transform problem of finding the antecedent of an anaphor into a classification task
- often done as binary classification

Comma-separated List of Features

- in bad cases you may end up with hundreds of Mb of these data
- 250 short German texts about Heidelberg with about 36000 words generated about 100Mb of training and testing data for a coreference resolution classifier

Machine Learning

- use the ML-algorithm of your choice (some may be suited better to your task)
- we have good experience with the WEKA machine learning library (www.cs.waikato.ac.nz/~ml) with Java reimplementations of several standard ML-algorithms
- we used the statistical software package R (www.r-project.org) which proved faster than WEKA by retaining flexibility
- we also used the original implementations of C4.5 (Quinlan, 1993) and C5.0 which are less flexible but very fast

Training vs. Test Data

- there is no need for this distinction for descriptive analyses
- even at recent ACL conferences there are many papers which distinguish not between training and test data (ML and symbolic approaches)
- e.g. Callaway (ACL 2003) used three texts from the NYT for developing his algorithm and for evaluating it
- e.g. Strube & Müller (ACL 2003) did 10-fold cross-validation without testing on holdout data, i.e. we were essentially tweaking on training data (and we should know better)
- if you do not distinguish between training and test data you do not know whether your findings generalize

Intrinsic vs. extrinsic evaluation

intrinsic evaluation: how well does a certain component perform ? (e.g. how many pronouns are resolved correctly?)

extrinsic evaluation: how much does a certain component contribute to the overall system performance? (e.g. how much does anaphora resolution contribute to a summarization system?)

Standard evaluation measures

Evaluation measures for particular components:

precision: done correctly/done overall

recall: done correctly/done in key

F-measure $F = 2PR/(P + R)$

Kappa: take the system as one annotation and compare it with the key

Evaluation measures for dialogue systems

This is not settled yet. A starting point would be

- PARADISE by Walker et al. (1997)

Production/Real World System

If your system/component/algorithm performs well you may consider to put in a real system:

- take the model generated by the ML algorithm
- put in an environment with unseen test data
- automatic annotation

Let’s have a dream . . .

- Evaluation by customer satisfaction.
- Evaluation in the market place.