Projects

Alexander Fraser & Liane Guillou
{fraser,liane}@cis.uni-muenchen.de

CIS, Ludwig-Maximilians-Universität München

Computational Morphology and Electronic Dictionaries
SoSe 2016
2016-06-06

Outline

1. Course Requirements

2. How Projects Work

3. Project Topics

4. Forming Groups

Outline

1. Course Requirements
2. How Projects Work
3. Project Topics
4. Forming Groups

Course Requirements

- To pass this course ...
  - Exercises and assignments
  - Regular attendance
  - Course project: implementation of a small project including extensive documentation; presentation
    - Roughly last 5-6 weeks of semester
    - Programming and data analysis intensive
    - Short presentation

Outline

1. Course Requirements

2. How Projects Work

3. Project Topics

4. Forming Groups

Projects in Computational Morphology and Electronic Dictionaries

- Projects will be done in groups of about 3 people
- Procedure will be to send Fraser a ranking of possible projects and teams (we will come back to this later)
- Please send the email at 19:00 this evening; emails sent earlier (even 1 minute earlier) will be looked at last

Evaluation

- Project code/analysis
- Write a project abstract, which includes what was done and who did what
- Project presentation
- Questions to individual group members

Schedule

- Today: presentation of topics (and later, your ranking)
- Wednesday: Project topics/groups announced, work starts (in class!)
- Three exercises over the next weeks: report on work in progress, interaction with Fraser, Guillou and Berlanda

⇒ this is a chance to ask questions and indicate problems, but also to meet with your group (you'll need to meet outside as well)

⇒ will also allow us to adjust topics (particularly if too hard or too easy)

- More information on polishing abstract and on presentation in an exercise at the end of June
- Abstract due July 6th at 8pm
- Presentations/questions on July 11th and July 13th (both in class in L155)

Outline

1. Course Requirements

2. How Projects Work

3. Project Topics

4. Forming Groups

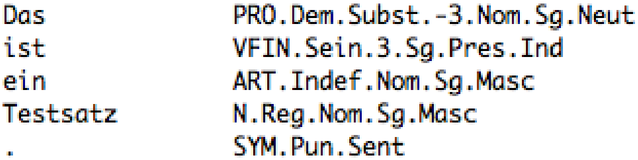
Introduction

Topics defined in terms of:

- Summary of what needs to be done
- Resources
- Programming Language (if applicable)
- Outcome
- Details of abstract (including whether German or English)
- What will be covered in the presentation

Problem: German Tagging and Lemmatization Difficult

- Summary: run German Marmot/Lemming (CIS Tagger/Lemmatizer) on two German corpora, provide a semi-automatic error analysis



```
Das           PRO.Dem.Subst.-3.Nom.Sg.Neut
ist           VFIN.Sein.3.Sg.Pres.Ind
ein           ART.Indef.Nom.Sg.Masc
Testsatz      N.Reg.Nom.Sg.Masc
.             SYM.Pun.Sent
```

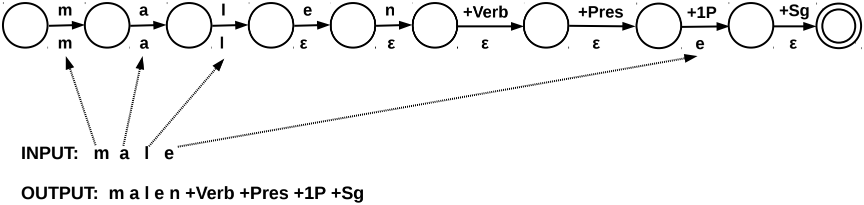
(example from RFTagger homepage, Schmid)

Project: Running German Tagger/Lemmatizer

- Summary: run German Marmot/Lemming (CIS Tagger/Lemmatizer) on two German corpora, provide a semi-automatic error analysis
- Resources: Two German corpora, SMOR (for manual disambiguation), Marmot/Lemming (see Thomas Mueller's web page)
- Programming Language: Perl (for the semi-automatic analysis)
- Outcome: Error analysis pointing to strengths and weaknesses of Marmot/Lemming in two domains, perl scripts for error analysis
- Abstract and Presentation: German or English, brief presentation on tagging/lemmatization, quantitative and qualitative discussion of results

Problem: German Verbs Have Complex Morphology

- Summary: Create SFST transducers which can be composed to analyze and generate German verbs (regular and irregular)



Project: German Verbs in SFST

- Summary: Create SFST transducers which can be composed to analyze and generate German verbs (regular and irregular)
- Resources: List of German verbs and their inflected forms, SFST
- Programming Language: SFST
- Outcome: Working transducers for analyzing and generating a large list of German verbs including both regulars and irregulars
- Abstract and Presentation: German or English, presentation of basic design of transducers including two examples (both regular and irregular verbs)

Problem: Rule-Based Machine Translation Highly Dependent on Morphology

- "Apertium is a shallow-transfer machine translation system, which uses finite state transducers for all of its lexical transformations, and hidden Markov models for part-of-speech tagging or word category disambiguation." (source: Apertium Project)
- Summary: look at extending the system, probably the morphologies in the English/German pair

Project: Apertium Rule-Based Machine Translation

- Summary: look at extending the rule-based transfer Apertium system (open source), probably the morphologies in the English/German pair
- Resources: open-source Apertium software, Apertium manual, possibly German/English parallel data provided later
- Programming Language: Perl (for checking coverage on corpus, possibly for error analysis, maybe for working with parallel data)
- Outcome: Extension of Apertium data in the English/German language pair
- Abstract and Presentation: English or German, basic presentation of how Apertium works, English and German morphologies, extensions carried out by the group

Outline

1. Course Requirements

2. How Projects Work

3. Project Topics

4. Forming Groups

Initial Group Discussions

- People discuss three times what to do in groups, grouped left-to-right and forwards-backwards and one move (front- row left, back-row right, forwards-backwards)
  - Please introduce yourselves, and then decide on a topic you could do together
- Email at 19:00 should contain TWO PARTS!:
  - PART ONE: Three teams (with team members!) and topics, in sorted order (preferred to least preferred)
  - PART TWO: Ranking of all 5 topics as an individual (preferred to least preferred)
- I reserve the right to completely ignore your preferences and just assign people however I want, sorry in advance

All Projects

- MT Error Analysis
- Compound Splitting
- Tagging/Lemmatization
- SFST German verbs
- Apertium English-German

Thank you for your attention.