Software project

Vivi Nastase

Summer semester 2012
ICL, University of Heidelberg

What is the purpose of this course?

- consolidation and acquisition of new programming skills:
  - modularization
  - interfacing
  - testing
  - documenting
  - ...
- development of team working and planning skills
- implementing software for solving a computational linguistics research question

Organization

- Prerequisits:
  - Programming exam (Congratulations!)
  - Participation in the Introduction to Resources course
- Duration: 1 semester, including the semester breaks

Course plan and duties

- Credits: 6LP + 4LP
- Presentations:
  - Work plan presentation (max. 45 minutes)
  - Final presentation (max. 45 minutes) + Demo (if appropriate)
  - Poster for a postersession at the beginning of the fall semester 2012 (probably mid-October)
- Documentation:
  - Documentation and archiving of the project
  - Documentation of the source code
  - README file
- If the project is outstanding – licencing and publishing

Grading

Grades are awarded to teams![1]

- Work plan and final presentations
- Code and documentation
- Team work
- Poster
- Attendance of joint events

"Bonus points" (a.k.a ego boosters): particularly good evaluations; graphical representation of the data and/or results

[1] In special situations individual grades can be awarded.

Content

In the software project course, you must, autonomously and collaboratively,

- plan
- program
- test
- document
- present

an assigned computational linguistics project

Events

- Time frame: 14-18 Tuesdays
- Group sessions – mandatory:
  - May 8^{th} 2012
  - May 15^{th} 2012 – distribution of projects
  - Work plan presentations – to be announced
  - Final project presentations – to be announced
- Office hours for advice on planning, implementing, ...:
  - 14-18 on Tuesdays when there is no group session, room 121/INF325

Timetable

Date	Event
24.04.2012	I am absent
01.05.2012	Holiday
08.05.2012	Organization and project presentations and assignments
15.05.2012	Office hours
22.05.2012	Office hours
29.05.2012	Work plan presentations
05.06.2012	Office hours
	...
23.07.2012	Final project presentations and Demos
30.07.2012	Final project submission
?.10.2012	Poster and demo session

Group structure

- 2-4 members per group
- groups can be formed before projects are assigned
- each student will be assigned to a project
- we'll try to settle conflicts of interest

Projects

- Multi-language semantic relatedness using Wikipedia
- Analysis of sensitivity of part-of-speech analysis to context
- Detecting and removing meta-language sequences from german data

Multi-language semantic relatedness using Wikipedia

Multi-language semantic relatedness using Wikipedia

Motivation NLP tasks such as segmentation, textual entailment, summarization, rely on computations of semantic relatedness between pairs of words or text fragments:

- *(car, vehicle); (car, highway)*
- *(apple pie recipe, cooking); (Apple iPad, touchscreen interface)*

Goal Develop an open-source, end-to-end system that implements the Explicit Semantic Analysis (ESA) for any language that has a Wikipedia version.

Multi-language semantic relatedness using Wikipedia

Idea words that appear in the same Wikipedia articles are somehow related:

car – transport, road, engine, Karl Benz, pistons, ...
highway – transport, road, European route, ...

apple – tree, fruit, cooking apple, iPad, Macworld Expo ...
cooking – cooking apple, fruit, vegetable, ...

Reading Gabrilovich, E. and Markovitch, S. (2007). "Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis", Proceedings of The 20th International Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India, January 2007

Multi-language semantic relatedness using Wikipedia

Work plan (based on Gabrilovich and Markovich):

- download Wikipedia dump
- build a Word x Article co-occurrence matrix
- recompute the entries in this matrix (normalization, filtering based on tfidf, ...)
- use this representation to compute relatedness between words or text fragments

Analysis of sensitivity of part-of-speech analysis to context

Analysis of sensitivity of part-of-speech analysis to context

Motivation Google has recently made available a very large amount of data in form of n-grams (n=1 to 5). By n-gram we mean here sequences of n words. For many applications it would be very helpful to have at least part-of-speech (POS) annotations of such word sequences.

Goal Determine the minimum n that leads to reasonable POS tags with an automatic POS tagger.

Analysis of sensitivity of part-of-speech analysis to context

Work plan:

- find manually POS annotated data in English, German (possibly other languages)
- split the given data into n-grams, n = 3 to MaxN (MaxN to be determined)
- strip the n-grams of the provided part of speech (POS) annotations, and keep them for evaluation.
- process the clean n-grams using existing POS-taggers.
- evaluate separately and compare the performance of the taggers on the different n (from 3 to MaxN), to determine the minimum length of a sequence that would provide high quality POS tags.
- if the determined n is greater than 5, develop algorithms for expanding n-grams

Detecting and removing meta-language sequences from german data

Detecting and removing meta-language sequences from german data

Motivation Discussions on the Internet contain both text and meta-language sequences – such as URLs, file paths and names, system log fragments, etc. For the purpose of understanding the message, such inserts are considered noise.

Goal Identify and remove such meta-language sequences from texts in a corpus of German Internet discussions

Detecting and removing meta-language sequences from german data

Paradigms to try:

- Language identification – most logs/URL/... will be in English, contrasting with the German text)
- Supervised classification, trained on annotated training data
- Regular-expression based meta-language detection

Detecting and removing meta-language sequences from german data

Work plan:

- split the corpus into training and test set
- annotate samples in the training and test set
- identify and annotate subcategories (URL paths, logs, ...)
- develop models on the training set
- evaluation on the test set