

# 1<sup>st</sup> International PhD School in Language and Speech Technologies

2006

2<sup>nd</sup> TERM

## PROGRAMMES

### **LINGUISTIC CORPORA AS RESOURCES FOR LANGUAGE ENGINEERING**

Udo Hahn, *University of Jena*

[udo.hahn@uni-jena.de](mailto:udo.hahn@uni-jena.de)

To be determined.

### **ONTOLOGY ENGINEERING: FROM COGNITIVE SCIENCE TO THE SEMANTIC WEB**

Maria Teresa Pazienza, *University of Rome Tor Vergata*

[pazienza@info.uniroma2.it](mailto:pazienza@info.uniroma2.it)

To be determined.

### **INFORMATION EXTRACTION**

Guy Lapalme, *University of Montréal*

[lapalme@iro.umontreal.ca](mailto:lapalme@iro.umontreal.ca)

1. Context of NLP applications
2. Initial work on understanding
3. Information extraction. Named entities. Superficial grammars
4. Question answering. Architecture. Evaluation

### References

Baeza-Yates, R. & B. Ribeiro-Neto (1999), *Modern Information Retrieval*. Addison-Wesley, Reading, MA.

Grishman, R. (2003), Information extraction, ch. 30 in R. Mitkov, ed., *Oxford Handbook of Computational Linguistics*. Oxford University Press, Oxford.

Harabagiu, S. & D. Moldovan (2003), Question answering, ch. 31 in R. Mitkov, ed., *Oxford Handbook of Computational Linguistics*. Oxford University Press, Oxford.

Hearst, M. (2003), Text data mining, ch. 34 in R. Mitkov, ed., *Oxford Handbook of Computational Linguistics*. Oxford University Press, Oxford.

Jacquemin, C. (2001), *Spotting and Discovering Terms through Natural Language Processing*. MIT Press, Cambridge, MA.

Maybury, M.T. (2004), *New Directions in Question Answering*. MIT Press, Cambridge, MA.

Minel, J.-L. (2002), *Filtrage Sémantique (du Résumé Automatique à la Fouille de Texte)*. Hermès, Paris.

Moens, M.-F. (2006), *Information Extraction: Algorithms and Prospects in a Retrieval Context*. Springer, Berlin.

Vorhees, E.M. & D.K. Harman (2005), *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, Cambridge, MA, ch. 10.

## **STATISTICAL MACHINE TRANSLATION**

Reinhard Rapp, *Rovira i Virgili University*

[reinhard.rapp@urv.cat](mailto:reinhard.rapp@urv.cat)

To be determined.

## **LANGUAGE PROCESSING FOR HUMAN-MACHINE DIALOGUE MODELLING**

Yorick Wilks, *University of Sheffield*

[yorick@dcs.shef.ac.uk](mailto:yorick@dcs.shef.ac.uk)

1. History and context of machine dialogue systems
2. Some philosophy, and the scope of the dialogue project
3. The main paradigms now in play in the field: do they really differ?
4. Architectures for dialogue systems
5. The past and future roles of machine learning in dialogue systems
6. Companions: emotion and persona in dialogue companions

### References

To be determined.

## **MACHINE LEARNING APPROACHES TO DEVELOPING LANGUAGE PROCESSING MODULES**

Walter Daelemans, *University of Antwerpen*

[walter.daelemans@ua.ac.be](mailto:walter.daelemans@ua.ac.be)

This course addresses the use of machine learning methods in language technology. Applied computational linguistics has put a lot of effort in the definition of reusable modules like part of speech taggers, chunkers, named entity recognizers, word sense disambiguation, etc., each of which can be defined as a classification problem and is therefore amenable to machine learning solutions. In this course, we address theoretical and methodological issues in a machine learning approach to developing these modules, and train practical skills in developing accurate, efficient, and robust classifiers for language processing.

1. Introduction. Crash course on machine learning. Learning as search, bias, supervised and unsupervised learning. Basic methodology for machine learning experiments. Types of machine learning algorithms. Natural language processing modules and applications. Formulating NLP problems as classification tasks (windowing, feature extraction)
2. The eager - lazy dimension in machine learning. Polymorphism and small disjuncts in language processing problems. Introduction to eager (rule induction, decision tree learning) and lazy learning (memory-based learning). Introduction to Ripper
3. Memory-based language processing. The "forgetting exceptions is harmful" hypothesis. Empirical evidence and theoretical motivation. Introduction to memory-based language processing using TiMBL

4. Developing NLP modules with Ripper, TiMBL and WEKA. Applications in computational morphology, tagging, chunking, NER, and WSD
5. Problems in comparative methodology. Issues in the comparison of different machine learning algorithms and different sets of information sources using the same algorithm. Search methods for classifier optimization
6. Ensemble methods in machine learning for NLP tasks. Types of ensemble methods (voting, stacking and arbiter approaches). Methodological issues
7. Hands-on exercises

## References

- Daelemans, W. & A. van den Bosch (2005), *Memory-Based Language Processing*. Cambridge University Press, Cambridge. See website: <http://ilk.uvt.nl/mblp>
- Daelemans, W., V. Hoste, F. De Meulder & B. Naudts (2003), Combined optimization of feature selection and algorithm parameter interaction in machine learning of language, in *Proceedings of the 14th European Conference on Machine Learning (ECML-2003)*: 84-95, 2003. <http://www.cnts.ua.ac.be/Publications/2003/DHDN03/>
- Halteren, H. van, J. Zavrel & W. Daelemans (2001), Improving accuracy in word class tagging through combination of machine learning systems, *Computational Linguistics* 27(2): 99-230, 2001. <http://www.cnts.ua.ac.be/Publications/2001/HZD01/>
- Jurafsky, D. & J. Martin (2000), *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall, Englewood Cliffs, NJ. See website: <http://www.cs.colorado.edu/~martin/slp.html>
- Mitchell, T.M. (1997), *Machine Learning*. McGraw-Hill, New York. See website: <http://www.cs.cmu.edu/~tom/mlbook.html>
- Ripper: <http://www.cs.cmu.edu/~wcohen/>
- WEKA: <http://www.cs.waikato.ac.nz/ml/weka/>

## **SEARCH METHODS IN NATURAL LANGUAGE PROCESSING**

Helmut Horacek, *University of Saarbrücken*  
[horacek@ags.uni-sb.de](mailto:horacek@ags.uni-sb.de)

This course puts a specific perspective on a variety of natural language processing procedures - efficiency through elaborate search techniques. In the context of this course, search is in two complementary ways: 1. search in the sense of managing a specific, widely homogeneous task effectively, 2. search in the sense of building a system architecture to orchestrate several, typically heterogeneous subtasks.

Reflecting the amount of research devoted to each of these issues, the course mostly deals with the first interpretation of search. Specifically, competing methods in a number of natural language processing tasks are compared in terms of the relation between effort and quality. The tasks addressed include some well-known ones, such as parsing and machine translation, but also several less common ones, prominently in the area of

natural language generation, such as the generation of referring expressions. The second interpretation of search is discussed along with architectures for the generation of text and for dialog systems. The course aims at illustrating the functionality and application of search techniques in various tasks. Specifically, students should learn about the benefit and drawbacks of competing methods, developing the ability to apply search techniques in similar tasks effectively.

1. Introduction and motivation
2. Search methods for natural language processing
3. Syntactic parsing
4. Syntactic generation
5. Constraint-based techniques in semantic processing
6. Search in machine translation
7. Rhetorical parsing
8. Generating referring expressions
9. Text-to-text generation
10. Architectural issues for generation
11. Architectural issues for dialog systems
12. Lessons learned

## References

- Beale, S. (1997), Hunter-Gatherer: Applying Constraint Satisfaction, Branch-and-Bound and Solution Synthesis to Computational Semantics, Ph. dissertation, School of Computer Science, Carnegie-Mellon University.
- Bohnet, B. & R. Dale (2005), Viewing referring expression generation as search, in *Proceedings of IJCAI-05*, Edinburgh, Scotland: 1004-1009.
- Carroll, J., A. Copestake, D. Flickinger & V. Poznanski (1999), An efficient generator for (semi-)lexicalist grammars, in *Proceedings of the 7th European Workshop on Natural Language Generation*, Toulouse, France: 86-95.
- Dale, R. & E. Reiter (1995), Computational interpretations of the Gricean maxims in the generation of referring expressions, *Cognitive Science* 18: 233-263.
- Germann, U., M. Jahr, K. Knight, D. Marcu & K. Yamada (2001), Fast decoding and optimal decoding for machine translation, in *Proceedings of the 39<sup>th</sup> Annual Meeting of the Association for Computational Linguistics*: 228-235, Toulouse.
- Horacek, H. (2006), Handling dependencies in reorganizing content specifications: a case study of case analysis, *Research on Language and Computation* 4(1): 111-139.
- Kay, M. (1996), Chart generation, in *Proceedings of ACL-96*: 200-204, Santa Cruz.
- Kiefer, B., H.-U. Krieger, J. Carroll & R. Malouf (1999), A bag of useful techniques for efficient and robust parsing, in *Proceedings of ACL-99*.
- Lemon, O., A. Gruenstein, A. Battle & S. Peters (2002), Multi-tasking and collaborative activities in dialogue systems, in *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue*: 113-124.
- Marcu, D. (2000), The rhetorical parsing of unrestricted texts: a surface-based approach, *Computational Linguistics* 26(3): 395-448.

Meteer, M. (1992), *Expressibility and the Problem of Efficient Text Planning*. Pinter, London.

Robin, J. & K. Mc Keown (1996), Empirically designing and evaluating a new revision-based model for summary generation, *Artificial Intelligence* 85 (special issue on empirical methods).

Shieber, S., F. Pereira, G. van Noord & R. Moore (1990), Semantic-head-driven generation, *Computational Linguistics* 16: 30-42.

White, M. (2006), Efficient realization of coordinate structures in combinatory categorial grammar, *Research on Language and Computation* 4(1): 39-75.

## **COMPUTATIONAL MORPHOLOGY**

Harald Trost, *Medical University of Vienna*

[harald.trost@meduniwien.ac.at](mailto:harald.trost@meduniwien.ac.at)

The course aims to give an introduction both from a linguistic and a computer science perspective to this very active area of computational linguistics.

1. Introduction to morphology. Provides a largely pre-theoretical overview of morphology (basic terminology; morphotactics, morphophonology, morphosyntax; morphological phenomena across languages)
2. Computational morphology: applications and techniques. Applications of computational morphology. Analysis and generation. Morphology and the lexicon
3. Finite-state morphology. From two-level morphology to Xerox's finite-state-tools. Gives a short overview of the development of finite-state morphology
4. Introduction into finite-state tools with practical examples. Shows how to describe morphological phenomena of varying complexity and from different languages with the use of finite-state morphological tools. Will be using the tools provided together with Beesley & Karttunen (2003)

### References

Beesley, K.R. & L. Karttunen (2003), *Finite State Morphology*. CSLI, Stanford, CA.

Sproat, R.W. (1992), *Morphology and Computation*. MIT Press, Cambridge, MA.

Trost H. (2003), Morphology, in R. Mitkov, ed., *Oxford Handbook of Computational Linguistics*. Oxford University Press, Oxford.