CSE6390 3.0 Special Topics in AI & Interactive Systems II
Introduction to Computational Linguistics
Insructor: Nick Cercone – 3050 CSEB – nick@cse.yorku.ca
Tuesdays,Thursdays 10:00-11:30 – South Ross 104
Fall Semester, 2010

Project Suggestions

The following lists of project suggestions are appropriate for course projects in CSE6390. All of these projects have been chosen with care to be well defined for a one semester senior undergraduate or introductory level graduate course project which is nonetheless somewhat open-ended. I will be more than happy to supply additional details concerning each project and serve as a source of reference material.

1. News stories from different sources often contain contradictory information regarding a particular event such as the number of people killed in an earthquake. Build a numerical expression recognizer and revolver that can identify equality and contradiction between numerical expression such as: "5 adults" ≠ "3 children and 2 adults", but "5 people" = "3 children and 2 adults".
2. Write a program to generate stories, which, given a set of characters, some goals, and a reasonable set of actions that these characters might follow in attempting to reach their goals, will generate a coherent story.
3. Design a Question-Answering System, perhaps, for example, to successfully handle family genealogy and statements such as "The mother of the father of Bill's wife is the wife of Sally's husband"`.
4. Write a program to extrapolate sequences like the following: AAAAAAAAAA...; ABACADAEAF..., ABDGKP...; AABABBABBB...; 1 2 4 8 16 ...; 1 4 9 16 ...; etc. This program should perform induction on certain general kinds of data in a manner superior to the majority of people.
5. Write a small theorem proving system capable of extracting answers to questions from a small database. Start with the predicate calculus form as input and translate the premises and rules of inference into clause form first.
6. Write a program to extend the Unification algorithm such that two terms might be unified if they are of the same type but not necessarily alphabetic variants of each other. Be careful since consistency in type hierarchies must be maintained.
7. Develop two prepositional phrase attachment classifiers, one should a corpus for training and testing, the other one should be non statistically based.
8. Build a GUI-based grammar development environment that will help users identify and fix bugs in their grammars.
9. Write a program to generate referring expressions: assume a collection of entities having attributes for shape, color, size, etc, then generate a noun phrase that mentions enough attributes in order to uniquely identify the intended entity (e.g. "the small green book")
10. Develop a morphological analyzer for a language of your choice.
11. Implement a rule-based language-independent syllabifier.
12. Develop a non-trivial grammar fragment and a parser for it.
13. Build a shallow discourse parser, which takes chunked or parsed sentences as input and yields a discourse structure as output.
14. Develop a text classification system which efficiently classifies documents in two or three closely related languages. Consider the discriminating features between languages despite their apparent similarity. Implementation should be evaluated using unseen data.
15. Build a natural language interface to search engine.
16. A character N-gram and word N-gram approach to classification of literature by literary period.
17. Rule-based acronym extraction and expansion
18. e-English normalization: converting SMS-Text to correct English
19. Blog generation using a dialog agent

20. Text format segmentation using HMM
21. Experiments in character N-gram based Information Retrieval
22. Character N-gram based approach to classification of movie reviews
23. N-grams and Spam: using N-gram analysis to detect spam email messages
24. A practical method for extracting prefixes and suffixes of Biological terms
25. Email authorship attribution using N-Grams
26. Source text disambiguation for improved machine translation
27. An Unsupervised Approach to morphological analysis for a language of your choice.
28. Improving automatic term extraction using shallow parsing
29. Using natural language queries for email retrieval
30. Context-dependent spelling correction in languages with no word boundaries
31. A comparative study of text categorization using a Naive Bayes classifier with different feature space and dimensionality reduction methodologies
32. Improving Naive Bayes classification using natural language processing
33. Information retrieval performance using morphology, part of speech tagging, and semantic expansion
34. Document clustering with automatic term extraction
35. An approach to evaluating the readability of texts
36. Proper noun detection for search engines