Spring 2010
CS 544 Final Assignment

This final assignment is worth 10% of your final grade. But if you choose to do the coding project, you can earn an additional bonus of up to 15%.

Paper
Pease write a discussion about one of the questions below. The answer should NOT be longer than 4 pages. **Please email your paper** (in ASCII, Word, postscript, or pdf) to *hovy@isi.edu* **on or before Friday May 7**.

The papers will be marked according to their creativity, strength of argument, and quality of writing. You do not need to provide answers that are consistent with what you've learned in the class *as long as the answers are credible and well-thought out*. The more computationally inspired detail you provide, of course, the better—vague claims and intuitions count for nothing, and code is too much, but some references to concrete results, papers, notes from class—these show that you have internalized the material of the course and can use it to think creatively about problems in language and NLP.

Please do not write just any nonsense. The assignment is short so that you need write only concrete, clear thinking, results, and conclusions. Base your arguments on things you learned in the class, with specific reference to algorithms, existing systems, evaluations, modules, and theories. Your own thinking is interesting, but only insofar it is solidly based on facts.

Please clearly identify which question you are addressing!

1. How would you apply statistical / machine learning techniques in a system that derives the *meaning* of natural language sentences? For example, how would you augment Mini-TACITUS and/or learn more abduction rules? Define relevant (aspects of) meaning, explain how you will create a training corpus and describe the overall system.
2. Given the material you've learned in the course, what killer natural language application would you like to implement? How would this application work? How would you evaluate it? Who would use it?
3. Design and describe a text summarization system that uses information extraction frames/templates for the main information and adds to this unexpected but important information that appears in the text. How would you build/learn a large (over 1000) collection of frames? How would the system recognize important additional information (and know that it's not already contained in the frame?). How would you integrate the two kinds of information into a single *coherent* summary? How would you evaluate the output so as to determine how well each part works? Include a flowchart of the system architecture, showing whatever automated learning you do.

Coding a summarization system: earn additional points! (up to 15%)

Implement a simple text summarization engine. The input should be
- a query of *M* words ($0 =< M < 5$) indicating the reader's interests
- a document of *N* sentences (one sentence per line)

and the output should be a summary of *0.3N* sentences.

Points will be awarded as follows:
- Required unless you do the information extraction frame method: At least 2 simple (word-based) topic scoring methods. Up to 5 points each for each method (word frequency counting of various forms, position, title, and cue phrase methods, query overlap method) — max 4 methods
- Optional: Up to 10 points each for each complex (cross-sentence) topic scoring method (cross-sentence overlaps of various kinds, lexical chains, discourse structure, etc.) — max 2 methods
- Optional (hard): Up to 20 points for an implementation of the information extraction frame/template method that includes at least 5 frames with at least 4 slots per frame
- Optional: Up to 5 points for redundancy checking, pronoun replacement, and sentence order manipulations that improve the coherence of the summary
- Required: Up to 5 points for method of integrating the scores of the various modules (the more advanced the method, the more points. Simply adding the various modules' scores gives 1 point; weighting them in some way is better, but requires an analysis showing how and why the particular weighting factors were determined)

Points will only be awarded if ALL the following is mailed to hovy@isi.edu **by Friday May 7**:
- The **code of each topic scoring module**
- A **trace-through** of at least 2 texts for *each* topic scoring module (can be the same texts) showing the output scores assigned to each unit
- The **code of the score integration module**, which a discussion/description of how the weightings (if any) were determined
- The **input and final output** of at least 2 texts, including the scores of the output sentences