CS 585: Natural Language Processing

Fall 2004
Final Project

One-page proposal due: November 4
Progress report due: November 18
Source code due: December 2
Final report and class presentation due: December 7
Project guidelines based on those of Chris Manning

1 Introduction

The final project is an opportunity for you to work on a larger NLP system on a topic of your choice.

The projects will be judged on clarity in defining the problem to be investigated, the methods used, thoroughness in considering and justifying your design decisions, and quality of your write-up, including your testing of the system and reporting of results. You will not be penalized if your system performs poorly, providing your initial design decisions weren't obviously unjustifiable, and you have made reasonable attempts to analyze why it failed, and to examine how the system might be improved.

The final project can be a group project. Indeed, by working as a group, you can attempt something larger and more interesting. However, the amount of work should be appropriately scaled to the size of the group, and you should include a brief statement on the responsibilities of different members of the team. Team members will normally get the same grade, but I reserve the right to differentiate in egregious cases. In general we would like group sizes of 2; if you are considering a bigger group, you need to talk to me. Solo projects are, of course, allowed.

You are free (and, where appropriate, encouraged) to make use of existing code and systems as part of your project, but you should make sure their use is properly acknowledged, and make clear what additional value your project is adding.

The first deadline (November 4th, midnight) is to submit a project proposal. This will be graded for thoughtfulness, and its strength in addressing the following six questions:

1. What is the problem or task that you propose to solve?
2. What is interesting about this problem from an NLP perspective?
3. What technical method or approach will you use?
4. On what data will you run your system?
5. How will you evaluate the performance of your system?
6. What NLP-related difficulties and challenges do you anticipate?

This is to encourage you to get organized, and also a chance for further dialog between you and the instructor. I can give you extra references, and also information on whether we think the scope of the project is too small or two big.

2 Data

A quite large amount of natural language data of various sorts is available at UMass. This includes collections from major publishers such as the Linguistic Data Consortium (http://www.ldc.upenn.edu/), and some smaller collections, such as text categorization and information extraction training and test sets. The biggest amount of this data is in English, but there is also some in major foreign languages.

In particular, you might want to consider the following data sources. The last three bullets are lists of additional sources, and may be especially interesting.

- Penn Treebank (parse trees and POS)
- Brown Corpus (parse trees and POS)
- NetTalk (text to speech)
- ACE (named entity IE from newswire, and relations)
- CoNLL (named entity IE from newswire)
- Stanford pointers: http://nlp.stanford.edu/links/statnlp.html

- CMU: http://www.cs.cmu.edu/ TextLearning
- ISI IE archive: http://www.isi.edu/ muslea/RISE

Size and Grading

A size recommendation is difficult to define, but roughly you should be aiming for each member of the team to do as much work as on two of the homeworks. You should aim to do something that is interesting, not just an exercise in programming. This may only be an extension of a previous homework assignemnt, and implementation of an existing method (with some of your own extensions, would be nice). There should be a clear focus in terms of what you hope to achieve, or hope to show.

You will be graded on the deliverables for all four due dates above: proposal, progress report, source code, oral presentation and report, with more emphasize on the later deliverables. In the first two deliverables, I'm looking for evidence of clear thinking, and good facility with the concepts we've learned in class, in your answers to the six questions named above.

Your project write-up should be adequate, but doesn't need to scale linearly in size. One person might want to write 5-6 pages. A three person project may well find that a 10 page write-up is quite sufficient. Think of the write-up as something like a small conference paper, focussed on NLP research questions and achievements, though you may want to include a bit more detail on methods used, examples, etc. The quality of your write-up is important.

It's hard to define exactly what the write-up should cover, because it depends on the project, but generally, I'm looking again for answers to the six questions above, also including

- a technical description of the method you used,
- discussion on the linguistic assumptions of the model and their validity
- a clear presentation of the data, experimental setup, and the experimental results,
- your analysis of those results,
- discussion of alternatives or things you tried to improve performance, and how they fared.

I'm happy to help give you direction in each of these aspects, so please actively communicate with me about your progress.

You should make all submissions by email to Gary and myself.

In your in-class presentation (both for the proposal and the final project) you should help make your points using slide transparencies—either the plastic or Powerpoint variety.

3 Project Ideas

Some of you are still having difficulty settling on a project topic. You are encouraged to think up your own, but I am realizing that some of you may need more concrete suggestions.

A good source of ideas could be recent NLP conferences. You can find many NLP conference papers available online at http://acl.ldc.upenn.edu.

- Write a system for some task in natural language clustering, such as finding related web pages, or learning part of speech categories from raw data, for which you might look at (Schutze 1993, Schutze 1995). Or, one could attempt to use clusters to improve the quality of a language model, or predicting what objects a verb takes.

- An information extraction system. This could aim to extract named entities (such as person names, organizations, etc) from a certain type of text (newspaper reports, biology articles, etc). This may involve extentions to your HMM homework. You could try to extract information about seminar announcments from email messages, so that the event could be automatically added to a calendar program.

- Write an information extraction system that extracts some type of content out of web pages—perhaps lists of corporate officers and directors, perhaps products and prices, or products and an overall evaluation of how good they are in reviews. Such systems commonly make more use of HTML markup. Consider using machine learning methods as well as hand-built detectors and classifiers. Look also at some of the wrapper generation data sets that can be found off Ion Muslea's page (the ISI RISE page above) there are a number of interesting data sets there for restaurants, etc.

- Write a system for anaphora resolution: when mentions of a noun or a pronoun refer back to something introduced earlier (preferably doing that problem as well as him). There are more recent papers, but one good one to look at is (Ge et al. 1998). Some annotated data is available at http://clg.wlv.ac.uk/resources/corefann.php .

- Write a text summarization system. This could take press releases, or newswire reports, and summarize them down to a few sentences. A lot of work in text summarization just selects whole sentences that best summarize a document (Kupiec et al. 1995, Salton et al. 1994, Goldstein et al. 1999). Some interesting recent work (Jing and McKeown 1999) attempts to actually take parts of sentences in a sensible fashion. There is a bibliography of work at:
http://www.ics.mq.edu.au/ swan/summarization/bibliography.htm

- Write (part of) a text-to-speech system. Implement a method for converting text to phoneme sequences, which can be used as input to a speech synthesizer. A good free speech synthesis system is the Edinburgh Festival speech synthesis system:
http://www.cstr.ed.ac.uk/projects/festival/ Some simple data to work with is the NetTalk data set, which is available for free on the Internet.

- Build (part of) a simple statistical machine translation system, and evaluate its effectiveness. The starting point is doing word for word translation based on finding words that correspond in bilingual texts. See M&S chapter 13, or (Melamed 1997). A reasonable small target would be to simply build a model of word alignments (i.e., an automatically learned bilingual dictionary) rather than a complete translation system. There are data resources available of both parallel and already aligned texts.

- Construct a text classification systems, which categorizes texts into useful categories. See M&S exercise 16.3 or 16.5. This should in some way go beyond what we did for naive Bayes spam classification, e.g., perhaps by using a maximum entropy classifier, or doing hierarchical classification. One possible domain of application here is to write an email spam filtering system, with many overlapping features.

- Text classification can be directed at many purposes other than topic, and many of the alternatives are more interesting and give more opportunities for using linguistic features. One version that has been explored a bit in recent years is positive vs. negative opinion. But there still aren't system that work very well, and there are clear challenges to find features that work better. For starters, see (Turney 2002, Pang et al. 2002). The data for the latter system is available from Lillian Lee's web site.

- Construct a text segmentation system, which divides a text into pieces on certain topics. Look at section 15.5, and perhaps exercise 15.8.
- Build a word segmentation system for Chinese or some other language written without word boundaries.
- Build a statistical parser, augmenting your CYK homework assignment to implement PCFGs. If starting from scratch, you should almost certainly build one that just works with unlexicalized categories—which usefully reduces the scale of the task. There are still a number of useful hypotheses you could experiment with about how different kinds of conditioning information could be employed to improving parsing accuracy. You could look at exercise 12.6 or 12.9 or other recent work like (Collins 1996, Collins 1997, Charniak 1996, Charniak 1997a, Charniak 1997b, Manning and Carpenter 1997, Eisner 1996, Klein and Manning 2003).
- Design and implement a part-of-speech tagger.
- Investigate collocation finding (Chapter 5), perhaps focussing on discontinuous collocations, ones longer than two words, or issues in better ways of identifying good collocates.
- Develop a good sentence boundary detection system: see M&S section 4.2.4, or (Mikheev 2000).
- Try building a system that learns hyperonym hierarchies (ISA links) automatically from text corpora. Or PART-OF hierarchies. Eugene Charniak and students have some recent papers on this topic. See: http://www.cs.brown.edu/people/ec/ or http://www.cs.brown.edu/people/sc/ Or (Riloff and Jones 1999).
- Write a system to correctly choose the order among multiple adjectives modifying a noun (so it knows that an old green leather couch sounds okay, but ??a leather green old couch sounds funny.
- There are quite a lot of competitions for grown-ups in the NLP world, where groups compete to build systems that outperform other systems. Many of these are on reasonable size tasks that people could attempt. Here are some examples you could look at: (a) CoNLL Shared Tasks: named entity recognition, semantic predicate argument structure, phrasal chunking etc. See the list at: http://cnts.uia.ac.be/signll/shared.html

(b) Senseval word sense disambiguation http://www.senseval.org/ (c) Chinese word segmentation: http://www.sighan.org/bakeoff2003/