Natural Language Processing: Final project

Reykjavik University – School of Computer Science

November 2009

1 Description

In this last programming project you can propose a project, but below are some ideas. Due to lack of time, the emphasis is on developing a **prototype** of a system, rather than building a fully developed system. It is assumed that your system uses some of the techniques that have been discussed in the course. It is preferred that you work on this project in a group of two students.

1.1 Grammar checking

Develop a system which reads a text in some language and points to grammatical errors in it. In the case of Icelandic, you might search for feature agreement errors between subjects and verbs, agreement errors in noun phrases, between prepositions and the following noun phrases, etc.

Use tools like *IceNLP* (or some equivalent tool for other languages than Icelandic) to perform tagging and parsing.

1.2 Machine translation

Develop a system which can translate texts from the source language S to the target language T, by using the so-called *shallow-transfer* method. Your program should perform shallow parsing on S and then translate each constituent, one by one.

Machine translation systems based on the transfer approach need to be able to map a word form in S to its lemma and then use the lemma for looking up the corresponding word in T (the target word then possibly needs to be inflected!). Note that if you want to develop a system for S=Icelandic then

IceNLP includes a lemmatiser, in addition to a PoS tagger and a shallow parser.

1.3 Named Entity Recognition (NER)

NER is a subtask of information extraction that seeks to locate and classify atomic elements in text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.

Develop a system based on hand-crafted rules that performs NER for a language of your choice. You can use basic units like a PoS tagger and a parser as an aid.

1.4 Combination of taggers

Use the three taggers *IceTagger*, *TriTagger* and *MXPOST*, to develop a combined tagger based on "simple voting". We will provide you with a training corpus, *01TM.txt* for *TriTagger* and *MXPOST*, and a test corpus *01PM.txt* to test all the taggers on.

Develop a user interface which allows the user to input text to be tagged with the combined tagger and displays the results. What is the relative gain in accuracy between the combined tagger and the best performing single tagger?

1.5 A statistical language model

Develop a program which can automatically generate word sequences (sentences). The program is based on a trigram model.

Use the words (not the POS tags) from some corpus (e.g. *eng.train* for English) for training. Then use the derived language model to generate sentences. The program offers the user to input the first two words but then it selects the next n words according the language model.

1.6 Verb subcategorisation frames

Develop a program which automatically collects information about the subcategorisation frames of verbs in a given language. The program reads a corpus (or texts from the web), performs tagging and shallow parsing, and extracts information from the parsed text about the verbs and their objects. The output should be a list of verbs along with the subcategorisation frames.

The frames show the number of slots or arguments attached to a verb, i.e. does a verb demand one object, two objects, a prepositional phrase, etc. For Icelandic, we also need the information about the case of the object governed by a given verb.

1.7 Intelligent Computer-Assisted Language Learning (ICALL)

ICALL is a relatively young field of interdisciplinary research exploring the integration of natural language processing in foreign language teaching.

Develop a system that helps students learn morphology/PoS tagging/shallow parsing in some language. The system might allow the user to input a sentence, analyse it and then give feedback based on an automatic analysis obtained using appropriate NLP components.

1.8 Intonation for Text-to-Speech

The goal is to get a speech synthesizer to sound more intelligent. You would develop a program that inserts special intonation markers into text to be spoken by a Text-to-Speech system (TTS). You should mark emphasis and phrasal tones. The assignment of intonation would be based on the annotation of Information Structure, which can be done using a Discourse Model. We may not be able to get our hands on a TTS for Icelandic, in which case it would be enough for the program to produce a printed record of the voice annotation, but if you choose to do this in English, you can use a TTS like Festival.

1.9 Question-Answering System

The goal is to get a system to give answers to questions about the contents of a text – all in natural language. For example, there could be a document describing the life and habitat of a certain animal. You would then process the text and store various facts that it describes ("x eats ants" or eat(x, ants)). A user can then type a question ("what does x eat?") and the system would answer ("x eats ants"). It is fine to assume that the text is a basic text (maybe from a children's book of knowledge).

1.10 Segmenting Icelandic Text

The goal is to automatically divide an Icelandic text into discourse segments and experiment with the Attentional Stack model for this purpose. There is not time to train a statistical segmenter, since there is no Icelandic corpus

that already contains segmentation tags, so this would be based on various rules such as finding Discourse Markers and possible shifts in place, time, voice and etc.

1.11 A Simple Embodied Dialog System

The goal is to construct a relatively simple dialog system, which can utilize a finite automata BUT it has to be augmented with a Discourse Model that informs the selection of nonverbal behavior, such as eyebrow and head movements while speaking (e.g. emphasizing new entities). It should also include some classing nonverbal behavior for interaction management (e.g. breaking eye contact when taking the turn). You should be able to use the CSLU toolkit for this, since it allows you to control a face from the conversation states.

1.12 Noun Phrase Information Status Tracking

The goal is to label all noun phrases with Information Status using the taxonomy given by Prince. You would develop a program that automatically annotates Information Status in an input text based on a dynamic Discourse Model. You would need to tackle the inference problem, in which case it is enough to solve some given kinds of inferences (like the part-whole relationship). This can be done in Icelandic or English.

2 Due date and hand-in

You will demonstrate your final project in a special demo session on Wednesday, December 2nd.

You need to hand in the following:

1. A 2-3 page report describing the functionality of your system, its scope and limitations, and how to run it.

2. All code, both source and executables.