Massachusetts Institute of Technology
6.863J/9.611J, Natural Language Processing, Spring, 2003
Department of Electrical Engineering and Computer Science
Department of Brain and Cognitive Sciences

Laboratory 1a: Two-level morphology Introduction

Handed out: February 10, 2003 **Due: February 19, 2003**

Goals: This laboratory will explore a computational approach to dictionary and morphological analysis — how to "parse" words, and assign them feature and part of speech labels for use in further natural language processing. Part 1 (this document) is designed for you to gain familiarity with the server computers and the PC-KIMMO system, in preparation for the second part of the laboratory. It is really meant to take you only a half-hour or so of computer time, and just a bit more of thinking. In the second part (handed out on Wednesday), you will design Kimmo automata and Kimmo lexicons to do a morphological analysis for a foreign language. (Well, perhaps foreign to a majority of you. . .)

- For this lab you will use the PC-KIMMO program. Importantly, we will be using a slightly older incarnation of this program (version 1.08) – if you go to the web site for PC-KIMMO,

  `http://www.sil.org/pckimmo/`

  you will see that the current version is 2.1.8. While this current version has many advantages, and we will be exploring some of these later on, the older version retains a certain clear division into rules and lexicon (dictionary), and does not have as steep a learning curve.

1 Part 1: Using PC-Kimmo

In Part 1, we will just crank up the PC-KIMMO machinery to make sure it works, and get you to think a bit about the logic of the two-level system. (Part 2 is the real laboratory.) To begin, login to a SUN Athena workstation. Then:

```
athena% attach 6.863
athena% cd /mit/6.863
athena% cd pckimmo-old
athena% pckimmo
PC-KIMMO TWO-LEVEL PROCESSOR
Version 1.0.8 (18 February 1992), Copyright 1992 SIL
Type ? for help
PC-KIMMO>
```

Now you have to load a set of rules (two-level automata) for a particular language (if you want to generate surface words) or rules and a lexicon for a particular language (if you want to recognize, i.e., parse) words. The rule files have the suffix `.rul` while the lexicon files have the appendix `.lex`. You can combine the loading of both in a `.tak` file. The rule and lexicon files would typically be in your own directory of course, and you would modify the `tak` file accordingly; in this first part, the files are in the directory where the program itself resides. You can now proceed as follows; the last 3 items are tests of the recognition and word generation machinery that are invoked in the `tak` file.

```
PC-KIMMO>load rules english
Rules being loaded from english.rul
PC-KIMMO>load lexicon english
Lexicon being loaded from english.lex
PC-KIMMO>generate ‘fox+s
  foxes

PC-KIMMO>recognize foxes
  ‘fox+s     [ N(fox)+PL ]
  ‘fox+s     [ V(fox)+3sg.PRES ]

PC-KIMMO>generate ‘spy+s
  spies

PC-KIMMO>recognize spies
  ‘spy+s     [ N(spy)+PL ]
  ‘spy+s     [ V(spy)+3sg.PRES ]

PC-KIMMO>
```

```
PC-KIMMO>recognize flies
‘fly+s     ‘fly+PL
‘fly+s     ‘fly+3SG
‘fly+s     ‘fly+PL
‘fly+s     ‘fly+3SG

PC-KIMMO>generate fly+s
flies
```

When you are ready to get out, type:

```
PC-KIMMO>quit
```

Of course, there is a facility for running entire files in and out via `log` files; see your PC-KIMMO documentation for this, or type `HELP` at the prompt. Be kind and make sure you log the file to your own directory.
Now, on to the simple warmup questions.

Question 1

Recognize the surface string `antibody`. What is the result? Does it make sense? Explain your answer in a few sentences.

Question 2

Generate from the surface string `refer+ing`. What is the result? What is going wrong? (Hint: take a look at the recognizer, and the rules it is using.)

Question 3

Recognize the surface string `traveler`. What is the result? Now, explain *why* this result obtains, by observing the sequence of “Lexicons” traversed by the pc-kimmo engine. You must turn tracing on. To turn on tracing, enter:

```
set tracing on
```

at the `PC-KIMMO`> prompt. You probably SHOULD have the session copied to a file by entering

```
log <~your-dir/output-filename>
```

OK, what fix-up does this suggest? (Hint: Compare this to *doer*. You must still be able to recognize *traveled* correctly; it isn’t so easy as one might think to do this correctly; I simply want to get you to think about the organization of the Lexicons.)

Question 4

Recognize the surface string `flier` with tracing turned on. This WILL generate a LOT of output, so *please* direct your output to your own athena directory! Keep in mind that you can refer to the PC-KIMMO documentation manual if you need to.

Paying attention to the sequences of lexical and surface characters processed in this example, comment briefly on the statement that "the recognizer only makes a single left-to-right pass through the string as it homes in on its target in the lexicon." Is this so? Try to characterize as precisely you can the kind of situation in which the behavior that you observe will arise.

Writing up this part

We'd like to get by with as little paper as possible for assignments. (Call it ecological necessity.) To this end, we'd like you to write up a web page on Athena for your lab report.
If you don't know how to create a web page, now is as good a time as any to learn.

Please remember that collaboration is *encouraged*, but please do write down the names of your collaborators at the beginning of the report. Also, please remember that cloned reports are not acceptable.