Informatics 2A: Tutorial Sheet 5 (Week 7)
Part-of-speech tagging

Shay Cohen

1. Suppose you are given a text corpus of 100,000 tokens involving 10,000 distinct word types. Assume for the purpose of this question that the corpus perfectly obeys Zipf's law. (This wouldn't be possible in practice, since the number of occurrences of each word type must be a whole number.)

   (a) Estimate how many word types are needed to account for *half* the tokens in the corpus. (Your calculation need not be too precise.)

   (b) The word *about* is the 60th most common word in the corpus. Estimate how many times it occurs.

2. The following are some of the POS tags from the Penn Treebank tagset, illustrated with sample words

NN	Noun, singular (*cat*)
NNS	Noun, plural (*cats*)
VBD	Verb, past tense (*sang*)
VBG	Verb, gerund form (*singing*)
JJ	Adjective (*huge*)
RB	Adverb (*hugely, soon, very*)
PRP	Personal pronoun (*you*)
PRP$	Possessive pronoun (*your*)
IN	Preposition (*on, up*)
DT	Determiner (*the, some*)
CC	Coordinating conjunction (*and, but*)
WRB	Wh-adverb (*where, why*)

   Tag the following text, ignoring punctuation. You may assume the above tags are sufficient for this purpose. Highlight any points at which you think difficult or debatable tagging decisions arise.

   > I was walking down the high street yesterday when I noticed an old man acting suspiciously. He was peering into various shop windows and writing things in a notebook. When he spotted me, he stuffed the notebook into his pocket and wandered off.

   (For this question, getting the 'right answer' is less important than having the experience of trying.)

3. Consider the following (artificial) sentence:

   The old man the lifeboats

   Use the version of *bigram tagging* described in the lectures to tag this sentence, using the tags DT, N, V, Adj and the following frequency data. (Rows correspond to potential POS tags for the word in question; columns

correspond to the POS tag of the preceding word.) You may assume *the* and *lifeboats* can only be tagged as DT and N respectively.

old	DT	N	V	Adj
N	8	2	3	2
V	0	0	0	0
Adj	34	5	13	17

man	DT	N	V	Adj
N	102	45	15	86
V	0	11	4	4
Adj	0	0	0	0

4. Now use the *Viterbi algorithm* to tag the sentence

   The old man the lifeboats

   using the following transition and emission probabilities. Include explicit backtrace pointers in your Viterbi matrix.

	DT	N	V	Adj
start	0.4	0.3	0.1	0.2
DT	0	0.6	0	0.4
N	0.05	0.3	0.4	0.25
V	0.4	0.3	0.1	0.2
Adj	0.1	0.5	0.2	0.2

Transitions

	lifeboat	man	old	the
DT	0	0	0	0.5
N	0.2	0.3	0.2	0
V	0	0.1	0	0
Adj	0	0	0.4	0

Emissions

   You may want to use a calculator to help with the arithmetic. Note too that in the transition matrix, rows represent the 'previous state', and columns represent the 'next state' (the opposite of the convention in question 3).

5. (Optional, but highly recommended) In written English, the *E-deletion* rule states that when a suffix beginning with e or i is added to a stem ending in e, the (first) e is deleted. Examples: *love^ing* ⇒ *loving*, *queue^ed* ⇒ *queued*. (There are some exceptions to the rule, e.g. *canoeing*, but let's not worry about these.)

   Design a finite-state transducer that applies this rule, in the same style as the transducer for E-insertion in the lecture slides. The inputs to the transducer should be strings over $\{a, \ldots, z, \hat{}, \#\}$, where # denotes a word boundary and ^ a morpheme boundary. The outputs should consist of words separated by word boundaries, using the alphabet $\{a, \ldots, z, \#\}$.

   You should take care to work within the framework of non-deterministic transducers as defined in the lectures, in which each transition may be associated with (at most) one input and one output character. If your transducer involves a large number of states, you need only draw a representative sample to illustrate the general pattern.

   To simplify the task slightly, you may assume that the one-character string e is not a morpheme in English.