TRA301/COS401: Homework 2
Due Date: March 5, 2009

1. **Spelling Correction Tool:** Now there is a new crisis at your company, ManTra. There is an important client who wants documents translated from English to Chinese. However, the English-Chinese translation software is unable to translate quite a few words because the documents have several misspelt words. You have to rise to the occasion and design an English spelling correction program.

   (a) (**10 points**): Implement a character-based edit finite-state transducer (*EditFST*) with insertion cost=1, deletion cost=1, substitution=2, using the FST toolkit. How many states and arcs does the FST have?

   (b) (**10 points**): Generate the top three variants and their scores for each word in the file "2-1b.txt" using *EditFST*. Show the FST operations for this task.

   (c) (**10 points**): The FSA from Homework 1, problem 1a (aka 1-1a) represents a large set of valid English words. Let us call it *ModelFSA*. Use *ModelFSA* to filter out non-English words generated in problem 1(b) above to find at least one valid English word and its edit-distance from the input word. You might have to generate more than three variants to find an English word. Show the FST operations to achieve this task.

   (d) (**10 points**): Generate the top five *valid* English words in the ascending order of their edit-distance from the input word. Show the FST operations for this task.

2. **Segmentation Problem:** As you might know, Chinese language does not add spaces between characters to indicate word boundary. The notion of a word token, however artificial it might be, is useful for further processing of texts. In this exercise, you will design a word segmentation system that takes a sequence of characters and inserts word boundary symbols at the appropriate locations. For example, the character sequence *thebirdflew* is word-segmented as *the ˆ bird ˆ flew.*

   (a) (**5 points**): Describe how the word segmentation problem can be viewed as a token tagging problem. What is the tagset you would use and what do the tags represent?

   (b) (**10 points**): For the character sequences in file "2-2b.txt", using FSTs or by writing code in your favorite programming language, design a program to segment the character sequences into word sequences. Use the English word list, "2-2bWordlist", to determine if a character sequence is a valid English word. Show the top 3 word segmentations for each character sequence.

   (c) (**5 points**): In 2(b), a word boundary at all characters is equiprobable. However, that is not the case in English. 's' is more likely to end a word than 'q'. Using the text in file "2-2c.txt", train the probability $P(boundary|character)$ for each of the 26 characters in the English alphabet.

   (d) (**10 points**): Using the probabilities from 2(c), redo 2(b).

   (e) (**20 points**): Using the text "2-2c.txt", build a bigram language model, i.e. compute $P(w)$ and $P(w_{i+1}|w_i)$ from the corpus. Use add-one smoothing technique to estimate the probability of events not observed in the corpus.

   (f) (**10 points**): Use the bigram model to score the segmentations in 2(d).