

Project 2: Word discovery

John Goldsmith

October 30, 2008

The goal of this the project is to write a program to discover words in a text in which the boundaries between words is not indicated—for example, a text of English in which the spaces have been removed. We have discussed some approaches to this problem in class, and you have also read about some approaches. You can replicate what you have read or heard, but I would like you to use your imagination to come up with new ways of solving the problem, or interesting variants on older ways.

Your goal, then, is to write an algorithm that will take a text in which no spaces are present. You may assume that the text is given to you with one sentence per line. I mean here “sentence” in the normal sense of the word, and you can use this to be sure that each line begins with a new word and ends at the end of a word. Please do not assume that you know what counts as punctuation in your language: treat all symbols as potentially part of a word. If you can discover what symbols are punctuation, that’s fine.

In the first assignment, you will do three things:

1. Get the data from which you will work. This is the Brown Corpus, which you have already used. There is a slightly revised version of it uploaded now, so please re-download it. There is also a copy of it with the spaces taken out, and that is the file that you will work on—the Brown Corpus itself is your gold standard.
2. You will write pseudo-code for the algorithm you propose to implement.
3. You will write code for the function that will evaluate precision and recall for the output of your algorithm (you will actually run your algorithm next week), given the “gold standard” file, which is the original file with spaces in it.

Special comment on size: The Brown Corpus is relatively large – it’s over 50,000 lines. It is possible that your program will be too slow to deal with the whole corpus. If that is the case, then pick two or three intermediate sizes (for example, 5,000 and 10,000 lines), and run and evaluate your program on those smaller version. Determine whether the quality of your results depend on corpus size. If your program can run on the whole corpus, then you don’t need to do that (although you might find it interesting to see how dependent your program is or isn’t to the size of the data).

Special comment on evaluation. You will compute *two* sets of precision and recall figures: one for types, and one for tokens. For the *token* evaluation, your goal is to predict which pair of adjacent letters on each line had a space between

them in the original. For the *type* evaluation, your goal is different. First, you make a alphabetized list of all of the words in the (gold standard) corpus—using the corpus where the words are separated by spaces. This list will serve as your gold-standard for the type-evaluation of precision and recall. You are attempting to reconstruct that list of words. Precision is the ratio of the number of words that your algorithm proposed that are also in the gold-standard word list, divided by the number of words that your algorithm proposed. Recall is the number of words that your algorithm discovers that are also in the gold-standard list, divided by the number of words in the gold-standard word-list.

In the second assignment, you will write the code,¹ run it, and evaluate its results on the gold standard. You will give the quantitative results, and you will explain in English what it does well that you are satisfied or happy about, and you will identify where it appears to go wrong. You will identify changes that you would like to make to your algorithm if you had more time.

In the third assignment, you will implement some changes in the algorithm for week 2, and evaluate (precision and recall again) any improvements in performance. You will explain those changes in English, and suggest where you would make your next changes if you had the time to do so.

Extra credit to anyone who runs and evaluates their program on a language other than English (but you must choose a language in which the standard orthography includes spaces between words).

¹You are not obliged to use the idea that you described in week 1.