

Finding compounds

John Goldsmith

October 8, 2008

The goal of this first project is to write a program to identify compounds. English has a lot of compounds: some are identified with hyphens, and most are not (e.g., *airplane*, *mailman*, *bookkeep*, *brainwash*, *hotdog*). It is easy to write an algorithm that finds compounds, but it is not easy to write an algorithm that does it well. So a significant part of this project is working on the method by which you determine how well your algorithm in fact works. We will use the now standard notions of precision and recall to measure success.

In the first assignment, you will do three things: 1. Get the data from which you will discover compounds (this is the Brown corpus). 2. Each student will decide for 100 words whether they are compounds or not. We will merge these lists for the second week. The required format is simple: you create a text file. Each line has one word. Put a space inside the word if and only if it is a compound, and use the space to break the compound into the two correct parts. E.g: hot dog Harvard If you are not sure if a word is a compound or not (is *outrage* a compound?), then do not put it on your list. This list is our “gold standard.” 3. You will write pseudo-code for the algorithm you propose to implement. 4. 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 that you will make in cooperation with your class-mates.

In the second assignment, you will write the code, run it, and evaluate its results on the gold standard that you have cooperatively built. 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.

By the way, here is a link to a list of some 2200 compounds that someone posted on the internet: <http://www.rickwalton.com/curricul/compound.htm>. Calculate how many of these words are identified as compounds by your algorithm, when these words are added to the Brown Corpus.

This problem is probably a lot harder than it looks. You might have to worry about not identifying words like “meat”, “menace,” and “mean” as compounds. In addition, you are likely to run into the data-sparsity problem: *you* may know that some string of letters forms a word in English, but the evidence of that may not appear in your data.

Precision and recall

We have a set of items being tested on some binary decision (in this case, the item is the space between letters inside a word, and we are testing to see whether that item is the boundary between two parts of a compound). The gold standard indicates “gold-standard yes (positive)” for some such positions, and all other positions are “negative” (or “gold-standard negatives”). On the other hand, your algorithm produces its own predicted “positives” and “negatives”; and these are rated as *true* or *false* depending on whether the gold-standard says that they are correct or not.

Precision measures how good the specifically identified items are that your algorithm identifies; recall measures what proportion of the items it tried to find were in fact found.

$$\text{precision} = \frac{[\text{true positives from algorithm}]}{[\text{total positives of your algorithm}]}$$

$$\text{recall} = \frac{[\text{true positives from algorithm}]}{[\text{gold-standard positives}]}$$

The terms come originally from the field of document retrieval, where the user would input some string of words, and the document retriever (think Google) would return a set of documents. We care both about not getting irrelevant junk sent to us, and we care about getting all the documents in the set that really are relevant to what we want to learn about.