Computational Linguistics I *CMSC723, Fall 2010*

P4: Discourse

Hand in at: `http://www.cs.utah.edu/~hal/handin.pl?course=cmsc723`.

Introduction

This project is all about discourse and document structure. It is relatively easy, in comparison to previous projects, so I will not give you as much help as before. This project is basically about TextTiling.

We will be working with Wikipedia data, and trying to restore document boundaries, section breaks and paragraph breaks. Your “training” data is in `wiki.tr` and your test data is in `wiki.te`. The first column in both of these files is the *type of break* that occurs **before** the line in question. The meanings are: 0=no break, 1=paragraph break, 2=section break, 3=document break. Our goal is to “guess” these values for the text in `wiki.te`, for which all sentences are marked zero.

For instance, in `wiki.tr`, you’ll find:

```
3       ‘‘ Main article : Animax ’’...
1       Being Sony ’s first attempt to offer a...
0       The series that contain more than 25 e...
0       In total , it is possible to find in o...
0       Also , at the end of every series , th...
0       As of January 2007 , they rely on a se...
2       Locomotion was bought by Sony Pictures...
0       During the time since the bought of th...
0       The network officially ceased to exist...
0       From then on , the network has been tr...
0       Of all the programming broadcast previ...
0       As of February 2006 , all those series...
1       The non-anime shows produced by MTV Ne...
0       ‘‘ Bob and Margaret ’’ is shown ( as o...
0       ‘‘ The Critic ’’ is actually broadcast...
...
```

This means that the first sentence (3) begins a new document (duh). The next sentence begins a new paragraph. There is then a section break between the sentence “As of January...” and the sentence “Locomotion was brought...” and so on.

1 TF-IDF *(*25%)

It will prove helpful to *downweight* the importance of frequently occuring terms when computing vector similarities. The tf-idf weighting scheme says that the importance of a term should be computed as tf × log(N/df), where tf is the number of times that term occured in the current document, and df is the total number of documents that contain that term (N is the total number of documents). (For OOV terms, set df=1.)

Compute IDF values for all words in the training set, after lowercasing every word. Hint: $N = 3000$ and here is the beginning of an idf table sorted by df. Store these values in `wiki.idf`.

word	df	log(N/df)
.	2999	0.000333388901237574
,	2998	0.000666888987703774
the	2996	0.001334223013136660
of	2996	0.001334223013136660
and	2996	0.001334223013136660
in	2992	0.002670228555878890
to	2985	0.005012541823544190
a	2985	0.005012541823544190
as	2874	0.042907501011276600
by	2869	0.044648751668882000

2 Discourse Similarities *(*30%)

Write a script to compute similarities at every sentence break of a file, based on 20 words before and 20 words after. The first word in the new sentence should be considered part of the "after" and not part of the "before." The similarities should be cosine similarities over tf-idf vectors. Recall that the cosine similarity is defined as:

$$\frac{\sum_w a_w b_w}{\sqrt{\sum_w a_w^2}\sqrt{\sum_w b_w^2}} \tag{1}$$

where, in this case, a and b are the two tf-idf vectors.

Using a window size of 20, I get the following cosine similarities for the first ten sentences in `wiki.tr`:

```
0
0
0.0409609410469252
0.000545138602764856
0.0219926615969466
9.9104717116977e-09
3.74324345506299e-10
0.0832709715135095
2.25654235688686e-07
6.23821320852918e-05
```

Save the results (should be 190723 lines long) in `wiki.sim`.

(WU1) Produce a plot akin to that in the TextTiling paper (also on the blog) where you show the similarities over time, versus the "true" segmentation. If you have `xgraph` installed, you can run the following command to produce something like this:

```
% ./plot_results.pl wiki.tr wiki.sim 1 200 | xgraph
```

This will only produce the graph for the first 200 sentences, because otherwise it's too hard to tell what's going on.

3 TextTiling *(*25%)

Now, we need to threshold the similarities to get predictions. We actually need *three* threshold values to separate documents, sections and paragraphs.

For starters, let's just use a single threshold of 0.1. Any similarity less than 0.1 will be considered a three, and anything at least 0.1 will be considered a zero. If you apply this to `wiki.sim` you can evaluate against the truth:

```
% cat wiki.sim | ./threshold.pl 0.1 0.1 0.1 | ./evaluate.pl wiki.tr -
confusion matrix:
S \ T | 0 1 2 3
------+------------------------------
   0  | 39247 8820 3061 4
   1  | 0 0 0 0
   2  | 0 0 0 0
   3  | 94351 27751 14493 2996

  recall @ 0: 0.76756238754596
  recall @ 1: 0
  recall @ 2: 0
  recall @ 3: 0.0214627017501128
total recall: 0
```

Here, `threshold.pl` is my script for thresholding.

In the evaluation, we see a confusion matrix (T=truth, S=system) and then our system's recall for finding 0s, finding 1s, 2s and 3s. Since we never guessed 1 or 2, we'll have a recall of zero for those. Our *overall recall* is the geometric mean of the individual recalls. Since some of these were zero, our overall recall is zero.

We can now set different thresholds to get our recall to at least be non-zero:

```
% cat wiki.sim | ./threshold.pl 0.01 0.1 0.5 | ./evaluate.pl wiki.tr -
confusion matrix:
S \ T | 0 1 2 3
------+------------------------------
   0  | 2246 584 101 0
   1  | 37001 8236 2960 4
   2  | 25068 5530 2078 52
   3  | 69283 22221 12415 2944

  recall @ 0: 0.766291368133743
  recall @ 1: 0.170867824318998
  recall @ 2: 0.0634930334881447
  recall @ 3: 0.0275492920842574
total recall: 0.12301913977018
```

So that's a 12.3% recall. Pretty crummy but at least it's doing something.

(WU2) Consider all thresholds of the form $k \times 10^{-j}$ for k in $1, 2, 5$ and j in $0, 1, 2, 3, 4$. Find the set of three thresholds that does best from a total recall perspective. What is this set, and what recall do you get?

4 Better TextTiling *(*20%)

There are many things that are potentially broken about our TextTiling, which (partially) explains its poor performance. Here, you'll try to do better.

- We are using full (lowercased) words, rather than stemmed versions. You could try stemming (google for "Porter stemmer", or just truncate each word at the first four or five characters).
- We are not throwing out stopwords, though of course they are getting low weight from idf scores.
- We are only considering lexical identity; perhaps synonyms should be matched instead. You could find synonyms and other relations in WordNet.
- We are only using a window of 20 words. Perhaps this is too small (or too large). And perhaps you want different window lengths for the different types of breaks that we're trying to find.
- Discourse cue phrases at the beginnings of sentences, or other Wikipedia specific constructions.

(WU3) What did you do? How well did you do on the training data? What about test data?

As always, your performance will be based on your performance on test data. Using the best performing thresholds from the previous section on test data, I get an total recall of 0.0. You can evaluate at `http://www.cs.utah.edu/~hal/tmp/texttiling.pl`. Your score for this section will be based partially on your writeup (10%) and partially on your test performance (10%). For every 1% better than 12.3 you get, you will get 1% credit (up to 10). In addition, the best three teams will get 8/5/3 points extra credit.