Some 18c projects

Nathanael Fillmore
October 1, 2011

Outline

1. Basic cleanup.
2. Language detection.
3. Duplicate detection.
4. Alignment.
5. Decade comparison.

Basic cleanup.

In order to correct many of the most common errors introduced due to OCR, we preprocessed the text of each document using the following simple rules:

- Replace “ ’ d” $\rightarrow$ “’d”, e.g., “reform ’d” $\rightarrow$ “reform’d”.
- Replace “& c” $\rightarrow$ “&c”, e.g., “& c” $\rightarrow$ “&c”.
- Replace “- ” $\rightarrow$ “”, e.g., “Spi- rit” $\rightarrow$ “Spirit”.
- Replace “-” $\rightarrow$ “ ”, e.g., “He boldly hiccups-but he cannot” $\rightarrow$ “He boldly hiccups but he cannot”.
- Remove all characters other than a-z, A-Z, 0-9, “&”, and “ ”.
- Lowercase everything.

The rules were performed one after another.

Examples:

- `http://localhost:9000/doc/raw/47515`
- `http://localhost:9000/doc/tok/47515`

Language detection

What counts as “English”? E.g., there are

- English-language books with non-English quotations,
- Foreign-language dictionaries, and
- Non-English texts with English facing-page translations.

Our criterion:

- A document is “English” if “substantially” more than half its text is in English.

Language detection

For each document:

- Sample six 150-word contiguous blocks of text.
- Send each block separately to Google Translate's language detector.
- Get back six votes, each for "English" or "not English".
- If at least three votes are for "English", classify as "English".

Language detection

Evaluation:

- Labeled 250 documents by hand.
- The following table shows the number of these documents that received k "English" votes from Google Translate ($0 \leq k \leq 6$), grouped according to whether I labeled them as truely "English" or "not English".

votes	#English	#not English	#total
0	0	9	9
1	0	5	5
2	0	3	3
3	2	1	3
4	6	2	8
5	21	1	22
6	199	0	199

- On the entire corpus, 9570 documents out of 112040 total (8.5%) were classified by the procedure as non-English.

`http://localhost:9000/isenglish`

Duplicate detection

Why care about duplicates?

- Want to know how many unique documents are in the database.
- Want to study publication history.
- Want to exclude duplicates from some kinds of downstream analysis.
- Want to use clusters of duplicates to build better statistical models.

Duplicate detection

What is a duplicate?

- Not completely obvious. Might as well choose something simple.
- Jaccard index:

$$g(D_1, D_2) = \frac{\text{number of terms in common}}{\text{number of terms total}}$$

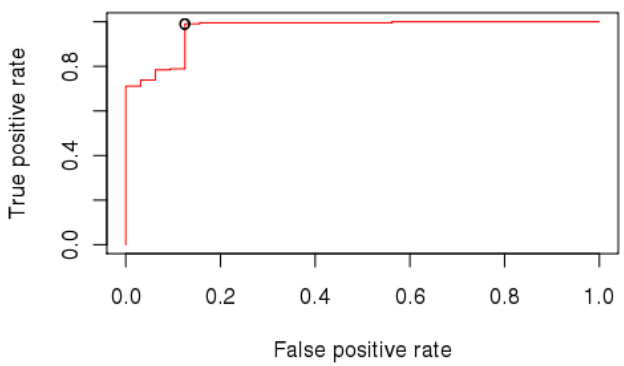
  where D_1 and D_2 are documents.
- By thresholding g, we get a family $\{f_t : t \in [0, 1]\}$ of duplicate-or-not classifiers:

$$f_t(D_1, D_2) = \begin{cases} 1, & g(D_1, D_2) > t \\ 0, & \text{otherwise} \end{cases}$$

Duplicate detection

What threshold works best, and how well does it work?

- For each ECCO-TCP document C:
  - Let D_C be the document in ECCO that maximizes the Jaccard index $g(C, D_C)$.
  - Check by hand whether C and D_C are actually duplicates. (Out of 250 pairs, I marked 218 as duplicates.)
- ROC curve:



$$FPR = \frac{\text{number said to be positive but actually negative}}{\text{number actually negative}}$$

$$TPR = \frac{\text{number said to be positive and actually positive}}{\text{number actually positive}}$$

"said to be positive" means that $g(C, D_C) > t$

"actually positive" means that I marked (C, D_C) as duplicates

Circle marks $t = 0.35$, the threshold used for further analysis.

Duplicate detection

- When we apply the duplicate scheme to the whole corpus, using a threshold $t = 0.35$, 33769 documents (30%) are marked as duplicates of earlier documents.

  `http://localhost:9000/isduplicate`

- More interesting to look at connected components:

  `http://localhost:9000/dupconncomp`

Alignment

For each pair of documents:

- If the documents are long (more than 1000 words), divide and conquer:
  - Make a collection of anchors:
    - Look for a good set of shared n-grams of words. (Try $n = 100, 50, 25, 10, 5$, in that order. Limit to 80 n-grams.)
    - Extend the n-grams to longer shared fragments if possible.
    - If no anchors are found, use the trivial no-overlap alignment.
  - Recurse for each between-anchor fragment.
  - Concatenate the recursion output and the anchors.
- If the documents are short: use Smith-Waterman on characters.

Intuition:

- The documents share a lot of long word n-grams, despite high errors. By anchoring to these n-grams we can speed up the alignment considerably.

`http://localhost:9000/alignment`

`http://localhost:9000/al/40078/49008`

Decade comparison

Was there large-scale vocabulary change in written English across decades in the 18th century?

- For each document, make a bag-of-words count vector, after discarding all words occurring <100 or >5000000 times in the corpus.
- For each pair of decades:
  - Average the count vectors within each decade.
  - Compute the cosine between the average vectors.
  - 10,000 times, permute the decade labels and repeat the previous two steps.
  - Let r be the number of times a permuted cosine was less than the observed cosine.
  - The fraction $(r + 1)/(10000 + 1)$ gives an estimate of the probability that a cosine under a random permutation of decade labels would be smaller than the observed cosine.

Decade comparison

	00s	10s	20s	30s	40s	50s	60s	70s	80s	90s
00s	1.0000	0.9935	0.9917	0.9870	0.9779	0.9763	0.9646	0.9565	0.9484	0.9463
10s		1.0000	0.9923	0.9900	0.9802	0.9759	0.9664	0.9588	0.9528	0.9500
20s			1.0000	0.9958	0.9904	0.9889	0.9802	0.9734	0.9657	0.9628
30s				1.0000	0.9932	0.9905	0.9847	0.9788	0.9718	0.9692
40s					1.0000	0.9962	0.9896	0.9873	0.9814	0.9794
50s						1.0000	0.9935	0.9904	0.9844	0.9834
60s							1.0000	0.9927	0.9914	0.9878
70s								1.0000	0.9942	0.9917
80s									1.0000	0.9958
90s										1.0000

	00s	10s	20s	30s	40s	50s	60s	70s	80s	90s
00s	1.0000	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
10s		1.0000	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
20s			1.0000	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
30s				1.0000	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
40s					1.0000	0.0015	0.0001	0.0001	0.0001	0.0001
50s						1.0000	0.0001	0.0001	0.0001	0.0001
60s							1.0000	0.0001	0.0001	0.0001
70s								1.0000	0.0002	0.0001
80s									1.0000	0.0001
90s										1.0000

(top) The observed cosines between count vectors, averaged by decade. (bottom) The levels of the observed cosines against the permuted cosines. Both matrices are symmetric, and only the upper triangles are shown.