To: Michael Wollowski
From: Alexandre van der Ven de Freitas and Ethan Campbell
Campus Mailbox: CM1453
Regrading: CSSE 413 IR assignment
Date: November 03, 2014

Information Retrieval Analysis

Introduction

This assignment covers information retrieval techniques, an essential tool that can be used on an IBM Watson-like system. Both students decided to implement the required BM25, skip bi-grams and passage term matching.

1. BM25

Okapi BM25 is a ranking function that ranks a set of documents based on the query terms appearing in each document [1]. Given a query Q containing the keywords $q_1,...,q_n$, the BM25 score of a document D is defined:

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i).\frac{f(q_i,D).(k_1+1)}{f(q_i,D) + k_1.(1 - b + b.\frac{|D|}{avgdl})}$$

where $f(q_i,D)$ is q_i's term frequency in the document D, |D| is the length of the document D in words, and avgdl is the average document length in the text collection. k_1 and b are free parameters: $k_1 \in [1.2,\ 2.0]$ and b = 0.75. $IDF(q_i)$ is the inverse document frequency weight of the query term q_i:

$$IDF(q_i) = log\left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}\right)$$

where N is the total number of documents and $n(q_i)$ is the number of documents containing q_i.

For this implementation, the value k_1 = 1.2. Also, for calculating IDF, the calculated value in the argument of the logarithmic function must be an absolute value.

2. Skip bi-gram

Skip bi-grams count the number of times two keywords are either next to each other or separated by one word. Let Q equal the complete set of skip bi-grams from a text. Let P equal the set of all combinations of 2 keywords.

scoreP = (P ∩ Q) / P

scoreQ = (P ∩ Q) / Q

Then the overall score is set equal to the harmonic mean of scoreP and scoreQ.

*score = (2 * scoreP * scoreQ) / (scoreP + scoreQ)*

3. Passage term Matching

Its scorer evaluates each passage on the basis of which terms it contains [2] . Given the terms t_1, …, t_n, for each passage a score p_i is computed as the sum of inverse document frequency (IDF) values of matching terms, normalized by the sum of IDF values of all terms in the question:

$$p_i = \frac{\sum_{j=1}^{n} w_{ij}}{\sum_{k=1}^{n} IDF(t_k)}$$

where w_{ij} is defined as $IDF(t_j)$ if passage i contains term t_j and 0 otherwise. IDF(t) is defined:

$$IDF(t) = log\left(\frac{N}{c(t) + 1}\right)$$

where c(t) is the number of documents that contain the token t and N is the total number of documents. IDF is used as weights for matches/mistmatches instead of uniform weights because not all tokens are equally important.

Result Analysis

For the BM25 using the keywords specified in the assignment command, in addition to the keywords: second president, impeached, general, world war, founding, democrat, republican, independent, congress, law; the following ordered result from the most likely to less likely was obtained:

[('docs\\GeorgeWBush.txt', 2.105430787331442), ('docs\\JamesMonroe.txt', -2.2852619380579133), ('docs\\JohnQuincyAdams.txt', -2.6003206102507326), ('docs\\MartinVanBuren.txt', -3.6821991020569205), ('docs\\WilliamHenryHarrison.txt', -3.764179923582801), ('docs\\AbrahamLincoln.txt', -3.7931985528770182), ('docs\\RutherfordHayes.txt', -3.8490865081513834), ('docs\\AndrewJohnson.txt', -4.255736358935788), ('docs\\DwightEisenhower.txt', -4.417296347488059), ('docs\\UlyssesGrant.txt', -4.766328253290934)]

Using the same set of keywords, for the passage term matching we have:
[('docs\\GeorgeWBush.txt', 0.25064858447619776), ('docs\\JamesMonroe.txt', 0.13602816713003116), ('docs\\JohnQuincyAdams.txt', 0.12861457312716482), ('docs\\GeorgeBush.txt', 0.12417820234376004), ('docs\\JohnTyler.txt', 0.12417820234376004), ('docs\\AndrewJohnson.txt', 0.11812085499009338), ('docs\\BillClinton.txt', 0.1107072609872271), ('docs\\GeraldFord.txt', 0.10558584677303845), ('docs\\GeorgeWashington.txt', 0.0921149054165055), ('docs\\WilliamHenryHarrison.txt', 0.0921149054165055)]

Comparing the two results we can see that the 1st, 2nd and 3rd match. Both results have the following presidents in their top 10: George W. Bush, James Monroe, John Quincy Adams, William Henry Harrison and Andrew Johnson.

Using the same set of keywords, the skip bi-grams returns:

[('docs\\WilliamTaft.txt', 0.025899280575539564), ('docs\\WarrenHarding.txt', 0.02521008403361345), ('docs\\WilliamMcKinley.txt', 0.02510460251046025), ('docs\\ZacharyTaylor.txt', 0.02455661664392906), ('docs\\WoodrowWilson.txt', 0.024226110363391652), ('docs\\UlyssesGrant.txt', 0.02419354838709677), ('docs\\WilliamHenryHarrison.txt', 0.02419354838709677), ('docs\\ThomasJefferson.txt', 0.023426061493411424), ('docs\\RutherfordHayes.txt', 0.02074688796680498), ('docs\\MillardFillmore.txt', 0.020718232044198894)]

Comparing all three results, the skip bi-grams is quite different with its rankings and inclusions. The results all have William Henry Harrison in their top 10 matching presidents.

In order to test the accuracy of the algorithms, we used the keywords: lincoln, civil war president, assassinated president. It is common knowledge that the first result for this search should be Abraham Lincoln, so the algorithm BM25 returns as the first result Andrew Johnson with Abraham Lincoln being the second result, which is a little bit odd. For the term passage matching, it returns Abraham Lincoln as the top result followed by Andrew Johnson.

Although passage term matching seems to be more efficient, when less than 3 terms are included in the keywords list, the algorithm returns all zeroes (0.0) for all documents' scores.

Conclusion

References
[1] - Wikipedia - http://en.wikipedia.org/wiki/Okapi_BM25
[2] - J. W. Murdock, J. Fan, A. Lally, H. Shima, B. K. Boguiraev - Textual evidence gathering and analysis - Watson Papers, IBM Labs.