CS454 Assignment 1

Benjamin Eavenson

September 22, 2020

1 Introduction

For this assignment we were required to implement the TF-IDF and BM25 search algorithms and examine how they performed on a CSV database.

2 TF-IDF Queries and Results

ranking.tf_idf("vineyard", 5):

[('1106', 0.00018143362842374427), ('913', 0.00014916698849273087), ('289', 0.00014592319442 8654), ('1855', 0.00014281749804965067), ('902', 0.00013984126308915463)]

ranking.tf_idf("merlot", 5):

[('1333', 0.00018677666270295947), ('1345', 0.00016690044870116 04), ('4887', 0.00015084816010900448), ('3910', 0.00014261821243480 31), ('4031', 0.00014261821243480 31)]

ranking.tf_idf("blueberry", 5):

[('887', 0.00035851516874071477), ('1007', 0.00035851516874071477), ('1489', 0.00028054910928 8638), ('1510', 0.00026885610581 01166), ('3924', 0.00026336770658 22916)]

ranking.tf_idf("merlot blueberry", 5):

[('887', 0.00035851516874071477), ('1007', 0.00035851516874071477), ('1345', 0.00030565015980 41519), ('1489', 0.00028054910928 8638), ('1510', 0.00026885610581 01166)]

ranking.tf_idf("blueberry merlot blueberry", 5):

[('887', 0.00071703033748142 95), ('1007', 0.00071703033748142 95), ('1489', 0.00056109821857 7276), ('1510', 0.00053771221162 02332), ('3924', 0.00052673541316 45832)]

3 BM25 Queries and Results

ranking.bm25("vineyard", 5):

[('1106', 4.13801512844258), ('913', 3.948944172792035), ('289', 3.926518247127191), ('902', 3.8824219508717532), ('252', 3.818103809660521)]

ranking.bm25("merlot", 5):

[('1333', 4.257723897733983), ('1345', 4.137509214558125), ('4887', 4.023896529004658), ('2824', 3.916356519145733), ('3910', 3.623899134299997)]

ranking.bm25("blueberry", 5):

[('887', 5.192806549029246), ('1007', 5.192806549029246), ('1489', 4.8979087521375995), ('1510', 4.842903370327804), ('4458', 4.789119730686847)]

ranking.bm25("merlot blueberry", 5):

[('1345', 7.728316696623085), ('1250', 6.811524555490099), ('475', 6.75384543213151), ('1467', 6.641368898328853), ('1887', 6.4795072181995295)]

ranking.bm25("blueberry merlot blueberry", 5):

[('887', 20.729849251503207), ('1007', 20.729849251503207), ('1489', 19.552607847178784), ('1510', 19.3330246098345), ('4458', 19.118318606168213)]

4 Comparison

I chose to examine how TF-IDF and BM25 performed differently when given the query "merlot blueberry".

TF-IDF:

[('887', 0.00035851516874071477), ('1007', 0.00035851516874071477), ('1345', 0.0003056501598041519), ('1489', 0.00028054910928863 8), ('1510', 0.0002688561058101166)]

BM25:

[('1345', 7.728316696623085), ('1250', 6.811524555490099), ('475', 6.75384543213151), ('1467', 6.641368898328853), ('1887', 6.4795072181995295)]

I noted that every result returned by BM25 contained both 'merlot' and 'blueberry', while most of the results returned by TF-IDF only contained 'blueberry'. I suspect that this may be because the IDF term in BM25 is normalized

to a lesser value with a logarithm, while TF-IDF does no such normalization on its IDF term.

I calculated how many documents contain each term and found that 255 documents contain 'merlot', while 155 documents contain 'blueberry'. This would cause TF-IDF to put a greater emphasis on documents containing blueberry, but wouldn't affect BM25's ranking as much because of its lesser weight on IDF.

The query frequency term in BM25 shouldn't have much of an impact with this query because each term only appears once in the query.