Drew Tatum
CSC 575
Final Project Report
Winter 2021

DePaul Course Information Retrieval System

The purpose of this project is to improve DePaul's course information retrieval system from the perspective of a student taking courses at DePaul University. There are a few limitations right now that exist when using DePaul's Campus Connect to search for courses to register. As of Winter 2021, queries that only match either the course ID or terms in the course's title will return the course. For example, the query "SQL" will only return courses that have SQL within the course name regardless if the term is within the course description. Another limitation is the system uses an exact term match when searching its system. Since the current system does not implement any type of stemming or lemmatization queries that are misspelled or are the plural form of a word like "regressions", it will not retrieve courses that have the term "regression" in its course title. One last notable restriction of the current retrieval system is that multiple topics cannot be searched at the same time. For example, the query "SQL and Regression" looks for courses that both have SQL and regression within the course title. If no course exists, which is currently true, no courses are returned. Even though a few courses have regression and SQL within their course title separately, the current system is not able to provide the results the user desires. The goal of this information retrieval system is to help fix these current issues that are impacting students from exploring different courses and from noting further potential development to improve their experience. Additionally the new information retrieval system implements a weighted similarity scheme based on the location of the queried term. For example, if the term appears in the course title it will be given a higher weighted value than terms appearing in the course description. This is based on the intuition that a course's title will provide the main intellectual focus of the topic while the course description can provide topics that are briefly discussed. The objective of this weighing scheme is to provide the user results closer to their true desired query output.

The data for this project was obtained using a web crawler created specifically for this project. The web crawler went through all the potential courses that are part of the Data Science Computational Methods curriculum and saved all the course ID's and course numbers within a temporary list. [1] Afterwards the crawler began visiting each course's specific URL based on the course ID list to obtain the name of the course and its course description. A dictionary was created with the course ID being the key with the course name being a value part of the nested key "Topic" while the course description being a value part of the nested key called "Info". Additionally a mapping dictionary was created so that each course was given a numeric value in incrementing order starting at 1 as the key with the course name as the value alongside its URL for part of the retrieval system. Both these dictionaries were saved in JSON format to disk to be used for later purposes of the retrieval system.

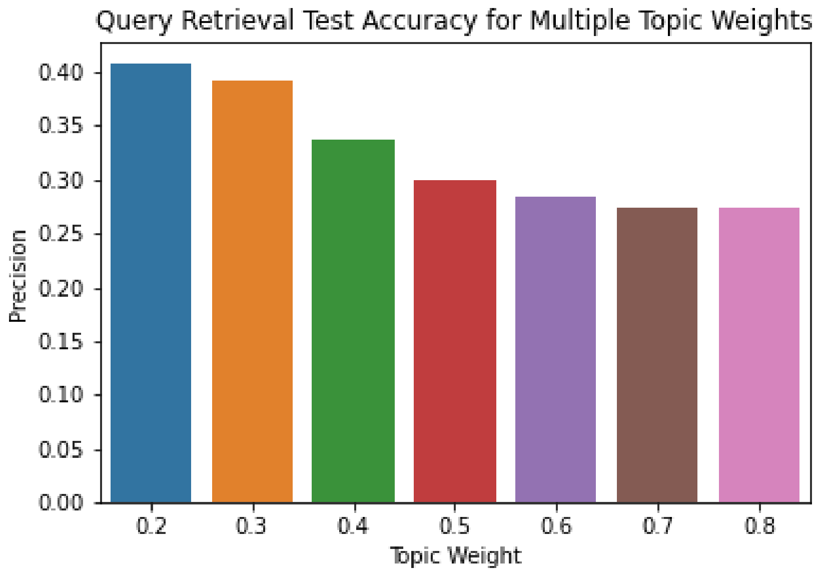
The inverted indexing system took as input the JSON file that contained the course ID, Topic, and Info parameters. Before indexing the system, a few linguistic preprocessing techniques were applied to the keywords obtained by the web crawler. Stop words from the English language using the natural language toolkit python module were removed. [2] Additionally all forms of punctuation were removed from the bag of words. Next, the NLTK Porter Stemming algorithm was applied to the tokens to obtain their final form before creating the inverted index. From this inverted index, an index and postings list were created. The index was a dictionary that contained the tokenized term as the key with document frequency and total term frequency as the values. The postings list was a dictionary that contained the tokenized term as the key with tuples of the document ID with the term frequency in that document as the values. Since the goal of this information retrieval system was to provide a weighting system based on the course topic and course information, two separate indexes and postings lists were created. The first was in respect to all tokens within the course description while the second was a substantially smaller token list with respect to the course topic. In this analysis the course information index and postings list had 682 unique tokens after linguistic processing while the course topic index and postings list had only 75 unique tokens. The inverted indexing system saved these four JSON format files (2 indexes and 2 postings list) to disk to be used by the retrieval system.

The retrieval system uses the four dictionaries from the inverted indexing system and the mapping dictionary created from the web crawler to return the most similar courses to a user based on their query. Once a user issues a query, the text goes through the same linguistic processing measures that the text from the web crawler went through including stop word removal, removal of punctuation, and Porter Stemming. Afterwards, the postings list creates a matrix vector space representation of the terms and document IDs. Term frequency inverse document frequency weights (TFxIDF) are computed onto the vector space representation using the inverse of the index's document frequency multiplied by the terms frequency in each document ID. The inverse of the document frequency is obtained from each index while the term frequency is obtained from the postings list. The TFxIDF values are normalized based on their vector norms to return a value between 0 and 1. The next step is the stemmed query is converted in a vector space representation with the same dimensions as the terms in the term by document ID matrix representation. This allows Cosine Similarity to be measured with the query vector representation to each vector representation of the document IDs. Since the goal of this project was to provide a weighted similarity score based on terms that appear in the course title vs the course description, two similarity scores are calculated. The first similarity score is based on the course description vector space model while the second similarity score is based on the course title vector space model. In this analysis, both models returned their top 10 similar document IDs alongside with their similarity values. The final part of the Cosine Similarity value is computed by using different weights for each similarity value where the total similarity will be between 0 and 1. The weight for the course title, 0.2, is given as a input for the system based on system testing using the MedLine test corpus collection. [3] The weight for the course description is 0.8 (1-title_weight). Since both systems return their top 10 documents, there can potentially be more than 10 documents that are scored using this weighted scheme since the top 10 documents might not overlap. After the weighting scheme, the top 3 courses are sorted and returned to the user. The courses are returned by opening their URL in a separate tab for the user. This is done by using the mapping dictionary which stores the course's URL.

As mentioned earlier, the weighted scheme is based on its performance using the MedLine test corpus collection of 1,033 documents and 30 test queries. The MedLine corpus was chosen due to the nature of its documents. The document collection contains article titles and their article description. The article titles were treated similarly how the course topic was treated. Likewise with the article description, it was treated the same as the course description. After the same linguistic processing steps, 2 inverted indexes were created. The first one resulted in an index and postings list that used the article topics. The second inverted index followed the same steps but was for the article description. The system was evaluated using its top 8 documents retrieved for each 30 test queries. Its precision and recall were measured with more emphasis on precision since most queries had more than 8 relevant documents. The system performed hyper parameter tuning for the title weight. Values of .2, .3, .4, .5, .6, .7, and .8 were given as input and later compared to see which weighting scheme resulted in a higher precision model. A weight of .2 resulted in the highest precision of .41 across all 30 test queries (See Figure 1). This value was given to the title weight in the course catalog information retrieval model.

The results and implementation of this information retrieval system resulted in a course retrieval system without any of the three limitations mentioned above. The system was able to provide courses relevant to the user using a weighted similarity measurement based on the course's name and description. Additionally, it offered the ability to search multiple topics and provide relevant information about them. That being said, there still are a few limitations for further development. One big limitation is the sample size of the documents. For now, the courses indexed are only the ones part of the Data Science Computational Methods course catalog. Also, there is no way to filter the courses based on the time of the year the class is offered or for keyword analysis on different synonyms or similar phrases. Further research needs to be done to validate the appropriate value for the course topic weight parameter. Relevance feedback could be used in the future to help approximate this value. Future steps include creating a server side for the index/dictionary and postings list based on their relational database structure.

Figure 1:



References:

[1] *MS data science: Computational*. MS Data Science | Computational | DePaul CDM. (n.d.). Retrieved March 15, 2022, from https://www.cdm.depaul.edu/academics/Pages/current/Requirements-MS-In-Data-Science-Computational-Methods.aspx

[2] NLTK. (n.d.). Retrieved March 15, 2022, from https://www.nltk.org/

[3] *Test collections*. Glasgow IDOM - Test collections. (n.d.). Retrieved March 15, 2022, from http://ir.dcs.gla.ac.uk/resources/test_collections/