CS 6200 Information Retrieval Final Project Report

Spring 2019 —Professor Rukmini Vijaykumar

Brian Desnoyers
Khoury College of Computer Sciences
Northeastern University
Boston, MA, U.S.A.
bdesnoy@ccs.neu.edu

John Goodacre
Khoury College of Computer Sciences
Northeastern University
Boston, MA, U.S.A.
goodacre.j@husky.neu.edu

Akshay Kulkarni
Khoury College of Computer Sciences
Northeastern University
Boston, MA, U.S.A.
kulkarni.akshay@husky.neu.edu

I. Introduction

This project will explore the development and evaluation of several search engines based for a provided test collection of scholarly journal abstracts. Baseline systems will be improved via query enhancement techniques, such as word embedding query expansion and stemming. A snippet generation and spelling error correction model will also be developed to improve the baseline retrieval systems.

II. Background

A. Dataset

This project utilizes a dataset consisting of 3024 abstracts from the Communications of the ACM (CACM) published between 1958 and 1979 [1].

B. Baseline Models

This project involves the development of several information retrieval systems, with an evaluation and comparison of their performance in terms of retrieval effectiveness. This involves four baseline models.

The first baseline model is a term frequency–inverse document frequency (tf-idf) vector space model [2]. This model involves computing distances between document and query vectors, through metrics such as cosine distance. These vectors are computed based on the tf-idf values for each term within the vocabulary, which is product of the term frequency within the document and the reciprocal document frequency of the term [2].

The second baseline model is a binary independence model (BIM) [3], which does not incorporate term weights, thus assuming that documents are binary vectors. This method attempts to estimate the probability that a particular document is relevant, assuming that terms are independent given document relevance [3], similar to a Naïve Bayes classifier.

The third baseline model is a Best Matching 25 (BM25) ranking [4], which expands the binary independence model to incorporate term weights for both queries and documents. The BM25 ranking algorithm is thus a probabilistic model.

The final baseline model is Lucene, which is an open-source indexing and searching tool [5], and is used within common search engine implementations, such as Apache Solr [6] and Elasticsearch [7]. Lucene uses its own scoring function which is based on tf-idf [5].

C. Query Enhancement

1) Query Expansion: Query expansion is a method used to improve retrieval system performance and relevance by adding additional terms to a query [8]. Several methods for query expansion exist, including pseudo-relevance feedback and word embedding-based query expansion. Pseudo-relevance feedback uses highly ranked documents to select new terms for query expansion and has been shown to be effective both with and without improvements, such as a term proximity heuristic for selecting expansion terms near query terms [9]. Another modern query expansion technique involves the use of word embeddings, which are commonly used in natural language processing. Word embeddings map co-occurance information to a lower dimensional space, often using a skip-gram or continuous bag of words model. Query expansion terms can be identified by finding nearest neighbors to query terms via these models [10].

2) Stopping: Stopping is a technique used to filter out specific words, known as stop words. Ideally these stop words will be common across many documents in the collection and thus have limited semantic meaning or impact on document relevance. Words might carry little meaning from a frequency (or information theoretic) point of view, or alternatively from a conceptual (or linguistic) point of view. Words that occur in many of the documents in the collection carry little meaning from a frequency point of view, because a search for documents that contain that word will retrieve many of the documents in the collection. By removing the very frequent words, the document rankings will not be affected that much.

While these stop words can be removed at index time, but are often used only at query time as this allows for more flexibility during search. Stop word removal on the basis of frequency can be done easily by removing the words with the highest frequencies in the document collection. As a result of stopping the most 200-300 frequent words, indexes may be between 30% and 50% smaller [11].

3) Stemming: Stemming is the process of reducing all words with the same root (or, if prefixes are left untouched, the same stem) to a common form, usually by stripping each word of its derivational and inflectional suffixes. There are various stemming strategies developed for different purposes [12]. Some stemming algorithms utilize a stem dictionary and others a suffix list. Many stemming algorithms, designed to improve IR performance and document relevance, do not use a stem dictionary, but an explicit list of suffixes, and the criteria for removing suffixes. Stemmers of the popular stemmer family, the Porter stemmers, have adopted this approach [13].

D. Snippet Generation and Highlighting

A classical approach for snippet generation is extractive summarization using an extension of Luhn's Algorithm [14]. This algorithm involves the identification of key phrases within sentences containing query terms. The query terms identified within these key phrases can be highlighted to the user via the search engine interface.

III. Proposed Implementation

A. Baseline Models

The baseline models will be implemented in Python. The tf-idf vector space model will be implemented by computing ranking scores based on cosine distance of tf-idf vectors computed based on a document index. This will utilize a vocabulary generated from the index, with Laplace smoothing. Similarly, the BIM and BM25 baselines will be implemented in Python. The final Lucene-based model will be developed using PyLucene [15].

B. Query Enhancement

Query expansion will be performed via pseudo-relevance feedback and Word2vec-trained [16] word embeddings. For pseudo-relevance feedback, 6 query terms will be selected based on the 10 top ranked documents to find the expansion terms. Due to the limited size of the training set, we will utilize a pre-trained set of word embeddings from about three million words from Google News [17]. This will utilize a skip-gram model [17]. The Gensim library will be used to perform initial processing of these word embeddings [18]. Stopping will utilize the provided project stop list. Stemming will utilize the provided query and document files.

C. Snippet Generation and Highlighting

Snippet generation will be performed via extractive summarization using an extension of Luhn's Algorithm. Sentences will be ranked based on a computed significance factor, which is the number of query terms within the key phrase divided by the total number of words in the key phrase. Initial query terms will be extracted and highlighted within the terminal output (`\033[1m term \033[0m`).

D. Combining Approaches

To combine approaches, a final run will combine stopping with the word embedding query expansion technique.

E. Extra Credit

A spelling error interface will be implemented.

F. Planned Contributions

Brian will provide the wrapping for the existing baseline models. Akshay will modify the ranking function of the BM25 model to create the binary independence model and develop the tf-idf vector space model. Brian will develop the query expansion model based on word2vec-trained word embeddings. John will perform the stopping and stemming runs. Brian will develop the snippet generation and highlighting system. The team will collaborate on the spelling error interface.

References

[1] B. Croft, D. Metzler, and T. Strohman, "Search engines: Information retrieval in practice, 2008."

[2] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[3] C. T. Yu and G. Salton, "Precision weighting-an effective automatic indexing method," Cornell University, Tech. Rep., 1975.

[4] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford *et al.*, "Okapi at trec-3," *Nist Special Publication Sp*, vol. 109, p. 109, 1995.

[5] A. Białecki, R. Muir, G. Ingersoll, and L. Imagination, "Apache lucene 4," in *SIGIR 2012 wWrkshop on Open Source Information Retrieval*, 2012, p. 17.

[6] T. Grainger and T. Potter, *Solr in Action*, 1st ed. Greenwich, CT, USA: Manning Publications Co., 2014.

[7] M. S. Divya and S. K. Goyal, "Elasticsearch: An advanced and quick search technique to handle voluminous data," *Compusoft*, vol. 2, no. 6, p. 171, 2013.

[8] E. N. Efthimiadis, "Query expansion." *Annual review of information science and technology (ARIST)*, vol. 31, pp. 121–87, 1996.

[9] Y. Lv and C. Zhai, "Positional relevance model for pseudo-relevance feedback," in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '10. New York, NY, USA: ACM, 2010, pp. 579–586. [Online]. Available: http://doi.acm.org.libproxy.mit.edu/10.1145/1835449.1835546

[10] F. Diaz, B. Mitra, and N. Craswell, "Query expansion with locally-trained word embeddings," *arXiv preprint arXiv:1605.07891*, 2016.

[11] D. Hiemstra and F. de Jong, "Statistical language models and information retrieval: natural language processing really meets retrieval." 2001.

[12] J. B. Lovins, "Development of a stemming algorithm," *Mech. Translat. & Comp. Linguistics*, vol. 11, pp. 22–31, 1968.

[13] E. Airio, "Word normalization and decompounding in mono-and bilingual ir," *Information Retrieval*, vol. 9, no. 3, pp. 249–271, 2006.

[14] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of research and development*, vol. 2, no. 2, pp. 159–165, 1958.

[15] A. Vajda, "Pulling java lucene into python: Pylucene," *Retrieved March*, vol. 23, p. 2008, 2005.

[16] T. Mikolov, K. Chen, G. S. Corrado, and J. A. Dean, "Computing numeric representations of words in a high-dimensional space," May 19 2015, uS Patent 9,037,464.

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[18] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.