Modulated Information Retrieval Task on Domain-Specific Collections

Owen Cozine, Phoebe Goldman, Julia Murillo, & Katelyn Wang
New York University, Department of Computer Science

Abstract
Information retrieval systems return documents from a collection that are relevant to a particular information query. The efficacy of a system's subtasks (e.g. removal of stop words, normalization of capitalization, etc.) is greatly dependent on the language used in the collection of queries and documents. We attempt an ad-hoc information retrieval task on the Cranfield 1400 Collection and NFCorpus, completing the task successfully on the Cranfield corpus. Using well-documented evaluation measures, we compare the efficacy of various auxiliary, open-source, information retrieval subtasks on this English language corpus. We find the features implemented from spaCy, particularly the stopwords list, to be the most effective at producing relevant results in the Cranfield corpus.

1 Introduction and Motivation

Information retrieval has been a topic of interest in Natural Language Processing (NLP) for decades. Formally, information retrieval consists of returning relatively small amounts of relevant material from a larger collection of documents that satisfy a particular information need, typically presented in the form of a question or query (van Rijsbergen, 1979). With the rise of the Internet and the widespread use of search engines, the everyday person expects the answer to any question that might cross their mind to be at their fingertips. These answers, however, lay amid an increasingly large number of documents that need to be reviewed. Information retrieval tasks serve to narrow down this number so that the user needs only review the most relevant documents, and therefore have only become more relevant in the pursuit of optimizing user experience in a world of ever-expanding user expectations. As large-scale systems, by necessity, are not optimized for finding domain-specific information, smaller scale retrieval systems may become more important in academic realms that require such specific relevant information, like that of medicine (Pohorec et al., 2009). What material is relevant is entirely dependent on the domain of the collection of documents being queried, as even in English-language academia (the focus of this study), style and language differ greatly among different domains and countries of origin. The word "function," for instance, has different meanings in physical science (e.g. a linear function of the distance around a cylinder) and nutrition (e.g. a patient's artery function). This study seeks to explore the efficacy of different features on a baseline information retrieval system using two corpora from different academic domains: aerodynamics and nutrition.

Despite contemporary use of information retrieval for large-scale search engines, initial research into information retrieval centered around sets of queries and documents that use formal, domain-specific language. The pioneering Cranfield experiments completed a set of information retrieval tasks on collections of abstracts from aerodynamics articles, developing queries that also fall in the aerodynamics domain (Manning et al, 2009). This narrow focus requires relatively fewer considerations when trying to determine what information might actually be relevant to a particular query as compared to large-scale searches. Besides the obvious fact that all abstracts and queries are centered around the same general topic, the task

was relatively simplified by a few other factors that come from using a domain-specific, formal language collection, including: the high probability that terms used in queries are well-defined within that particular domain, the likelihood that queries and documents developed in an academic setting do not contain any spelling errors or other typos, and the unlikelihood of any slang or idioms being used.

2 State of the Art

As mentioned above, in recent years there has been increased interest in domain-specific language information retrieval. This is due in no small part to the continuously growing field of medicine and medicine-related information that might be queried, including both academic scientific research and patient information. Outlined below are some of the articles within this burgeoning topic. While the present study examines well-documented methods in the context of domain-specific language, these articles are on the cutting edge and introduce new methods of information retrieval that can be tailored to particular domains.

In order to create an information retrieval system that is compatible with highly-specific medical domains, Fautsch & Savoy (2010) suggest an adaptation of the vector-space model that accounts for a term's relevance in the target domain, among other adjustments. (The basics of the vector-space model are described below, in Section 4.1.) The authors proposed that, since the system is domain-specific, a term's frequency within that corpus would be a relevant measure to consider when determining the weight of that term. Their model was evaluated on four separate domain-specific collections in English and German against the classic vector-space model and a probabilistic mode; they used the classical vector-space model's performance as a baseline performance goal and the probabilistic model as an upper-level performance goal. The authors found that their adapted model significantly outperformed the classic model in all collections and performed similarly to the probabilistic model in the German collections. Fautsch & Savoy therefore concluded that an adapted vector-space model could be beneficial for creating effective domain-specific information retrieval systems.

Castells et al. (2018) similarly suggest an adaptation of the vector-space model in order to adapt to domain-specific corpora. The authors found that for ontology-based information retrieval systems, a vector-space model that uses annotations of the documents returned better results than a classic vector-space model system. While this study has recognized limitations, primarily from the automatic system used to annotate the documents examined, it is an interesting step in adapting an older method to fit modern domain needs.

Scells et al. (2018) propose not only an information retrieval system for domain-specific documents but an entire open-source framework under which a domain-specific searching application may be made. The proposed framework consists of four major components written in Go: a common query representation, a parser and compiler to modify queries to fit that representation, a pipeline for completing information retrieval experiments, and a pipeline that can be written in domain-specific language. The experimentation pipeline is complete with different modules that might be relevant in an information retrieval system, much like the ones

outlined in our study. This novel software, if utilized properly, can provide researchers the ability to more easily determine what factors make an effective information retrieval system for their domain of interest.

3 Corpora

3.1 Cranfield 1400 Collection

The Cranfield 1400 Collection is a relatively old collection of aerodynamics journal articles, together with 226 queries, 1,400 documents, and 1,837 evaluations. The relevance evaluation document contains query-document pair scores descending order. We used a development set of queries and documents to develop our system and the entire collection as our test corpus.

3.2 NFCorpus

NFCorpus includes non-technical English queries about nutrition and academic medical paper documents. While the 3,244 queries are topics, video descriptions, article and video titles extracted from NutritionFacts.org, the 9,964 medical documents are mostly from PubMed. We attempted to use the development and test sets to test our system performance on retrieving answers that contain nutrition and medical terms.

3.3 A Note on the BOLT Corpus

Our original view for this project was to explore the efficacy of different modules on a large corpus of informal language documents and queries – namely, the DARPA Broad Operational Language Translation (BOLT) corpus (Chen et al., 2018). This corpus consists of pilot, development, and test sets that each in turn consist of discussion threads from forums in English, Mandarin, and Arabic and queries relevant to these threads. This corpus was developed with the goal of combining information retrieval tasks with machine translation tasks on informal language, such that an information retrieval system would be able to handle queries and to return relevant documents in all three languages.

The test corpus is incredibly large, with just the English language portion containing over three million words. The original authors, therefore, did not score the relevance of all query-document pairs, instead only scoring those pairs retrieved by their system. For our purposes, this meant that some pairings produced by our system might not have had relevance assessments from the original authors, unless we were able to exactly replicate the original system's results, which was supremely unlikely for a couple of reasons. For one, our system would only be considering queries and documents in English, whereas the original system used all three languages. Even if we had used all queries and documents in all three languages, the likelihood of four undergraduate students being able to replicate a project from the Language Data Consortium in half a semester is completely improbable, if not actually impossible. As we did not have enough time or resources to determine relevance scores ourselves (again, we were working against a time limitation and with an incredibly large corpus), and we did not want to

consider hiring out the relevance assessments (as other studies using BOLT have done), we were left a little lost. The original BOLT source code does provide a script that can be used to predict the relevance assessment of query-document pairs that were not originally assessed. We spent about a week figuring out how to make our system output compatible with this script before determining that it was impractical for us to try to use for this project because of the computation time necessary and the unreliability of the assessment scores it produced.

All in all, we spent the majority of our time on this project focusing on a corpus that we did not end up being able to use. We originally were using the Cranfield 1400 Collection as a way to explore the modules we wanted to use while we figured out how best to use the BOLT corpus, and then as a formal language foil to the BOLT corpus' informal language. We therefore created our system and chose the modules we would use with the underlying assumption that we would be comparing formal and informal language information retrieval. For our final tests, we kept only those that we thought would be at all relevant to the two corpora we did end up using. Had we used the BOLT corpus, we would have also included idiomatic dictionaries.

4 Methodology

4.1 The Vector Space

Three primary modes of completing an information retrieval task have emerged through the decades of research: Boolean retrieval, the vector-space model, and probabilistic retrieval (Manning et al., 2009). All three models provide different ways of determining relevance between queries and documents, with the Boolean model being the most simplistic. Probabilistic models of information retrieval are often favored for use in contemporary systems, as they use machine learning techniques to dynamically optimize the information retrieval task (Manning et al., 2009; van Rijsbergen, 1979). For the purposes of this study, however, we focus on the use of the vector-space model.

In the vector-space model, both types of given data – queries and documents – are represented as term vectors that hold each word in the given query or document and that word's particular weight. A common weight metric, and the one used in this study, is the Term Frequency-Inverse Document Frequency (TF-IDF) score. Term frequency (*tf*) is defined as the proportion of times a term *t* appears in a document to the total number of terms in that document. Inverse document frequency (*idf*) is defined by Meyers (2021b) as "the reciprocal proportion of documents that contain the term *t*, normalized with a log function," as shown by the equation:

$$idf_t = ln(\frac{NumberOfDocuments}{NumberOfDocumentsContaining(t)})$$

The TF-IDF score for a particular term, therefore, is the product of its *tf* and *idf* measurements. With TF-IDF scores, terms that are characteristic of a particular document have higher scores than those that are solely common (based on *tf* scores) or unique to that document (based on *idf* scores).

Relevance between document and query vectors are then calculated from their term vectors. In this study, we used cosine similarity scores, which are the cosine of the angle between the query vector *q* and document vector *d*, given by the equation:

$$Similarity(q,\ d) = \frac{q \cdot d}{\sqrt{q^2 \times d^2}}$$

Cosine similarities scores are between *0* and *1*, with scores closer to *1* denoting higher similarity between the query and document vectors.

4.2 Baseline System

We first created a baseline system based off of the vector-space model described above using Python. It takes a set of queries and documents as its input. For each query, the system creates a term vector containing each word in the query along with its TF-IDF score from within the queries. The system then creates a complementary term vector for each document, which each contain the words the query and document share and their TF-IDF score from within the documents. Cosine similarity scores between vectors for each query and document are then calculated and returned.

4.3 Modules

We used features from two open-source, NLP, Python libraries: Natural Language Toolkit (NLTK) and spaCy. Both libraries are widely used to develop NLP systems, though they were created with different purposes in mind. NLTK was originally created as part of a computational linguistics course at the University of Pennsylvania, and is therefore primarily used for teaching NLP and creating systems in an academic, research-based setting (Bird et al., 2019). The primary goal of spaCy, on the other hand, is one of commercial production (Explosion).

We used the following features from NLTK (NLTK Project, 2021):

- Stopwords Corpus – a list of 26,220 English words that are typically not query-document specific (eg. “a”, “while”, “you”) and that can be filtered out of query and document vectors
- English Snowball stemmer – a system that takes a word as input and returns its stem

We used the following features from spaCy (Explosion):

- Lookup Lemmatizer – a system that takes a work as input, downcases it, and checks it against a lookup table of English words to figure out its stem, which it then returns
- Stopwords - a list of 326 of the most common English words that can be filtered out of queries and document vectors

We tried to use word2vec, a pretrained model for word vectors provided by spaCy. The computational time this process took, however, exceeded 24 hours with the Cranfield development corpus, so we did not include it in our final test run.

In addition to these open-source features, we also used the Python String function *.lower()* and string *.punctuation* to downcase words (when not using the spaCy lemmatizer) and to access a list of punctuation to filter out of queries and documents, respectively. We also used the list of stopwords provided by Professor Meyers for Homework 4.

All six modules could be implemented individually or in any of the available permutations. Our entire modulated system code is submitted with this paper as supplementary materials (__init__.py, cranfield.py).

4.4 Evaluation Metrics

To evaluate the results of our system against the original Cranfield and NFCorpus relevance assessments, we used Mean Average Precision (MAP) scores. As the output is ranked, this measurement of goodness seemed to be the most appropriate. We used a modified version of the scoring script provided by Professor Meyers for Homework 4. It is submitted with this paper with the supplemental materials (cranfield_score.py).

5 Evaluation

Below is sample output from the final run of our system on the Cranfield test set.

Test	**MAP**	**Test**	**MAP**	**Test**	**MAP**
simple	0.218	spacy_stop_punct	0.266	downcase_nltkstopwords_snowballstemmer	0.269
downcase	0.218	spacy_stop_ stem	0.257	downcase_punct_snowball stemmer	0.206
punct	0.206	downcase_punct	0.206	punct_nltkstopwords_snowball stemmer	-
nltkstopwords	0.269	downcase_nltkstopwords	0.269	downcase_punct_nltkstopwords_ snowballstemmer	-
snowballstemmer	0.218	downcase_snowballstemmer	0.218		
spacy_full_ normalization	**0.281**	punct_nltkstopwords	-		
spacy_stop	0.245	punct_snowballstemmer	0.206		
spacy_lemmatize	0.184	nltkstopwords_snowballs stemmer	0.2688		

* Note: all tests that include spaCy also include downcase

In evaluating our system results, one must be aware that the small size and limited topics of Cranfield collection may have skewed the result and influenced our accuracy. The following section will analyze the modules performance based on the MAP score of the system output against the original collection's relevance evaluation.

Among all modules, the different stopword lists seem to be the most impactful; we can see significant increase in MAP score in both NLTK and spaCy modules as the stopword list is put into use. The stopword list from Homework 4, however, consistently scored lower than the NLTK stopwords list.

Overall, the modules implemented from spaCy do better in improving the score compared to those from NLTK. We believe that this is due not only to the fact that this library is more user and production friendly, but also that it uses more up-to-date algorithms compared to NLTK, which in turn yields better results.

Regarding punctuation, those with lower frequencies (e.g. : ; ? !) have less influence than commas and periods on the final score, as might be expected. The scope of the Cranfield corpus might also affect this outcome; given that the Cranfield corpus consists only of abstracts from scientific articles, and this is to be expected. Question marks and exclamation marks are more likely seen in informal language corpora, while colons and semicolons might be more relevant in the bodies of those articles.

Due to the fact that NFCorpus provides informal queries and term-heavy documents, our system did not provide usable results compared to the given relevance judgment files. We believe the system needs further improvements to achieve an identical level of MAP score to the system provided by the corpus contributors (Boteva et.al, 2016).

6 Conclusions and Future Directions

During the process of building a modulated IR system on domain-specific collections, our team took the TF-IDF approach and incorporated NLTK and spaCy features in our modulated system, and evaluated the system performance with the Cranfield 1400 Collection. Overall, the spaCy modules performed well on the topics of aerodynamics. Overall, our system was not sophisticated enough to detect synonyms (e.g. layman words and medical terminologies), and therefore was unable to produce meaningful results from the NFCorpus.

In future work, we would try accessing the TREC corpus, which has sufficient data in various domains and forms that support the development of an open domain IR system. To improve the compatibility of different domains, we also hope to explore the learning-to-rank modules such as *RankBoost,* which provides weak-ranker systems a way to retrieve and learn relevant information better from sparse and term-heavy data sets like NPCorpus and Cranfield.

7 Author Contributions

In completing this project, we adhered to the roles and responsibilities layed out in our project proposal. Owen Cozine researched relevant modules that could be implemented on top of the baseline system and executed the final test run; he also helped troubleshoot programming bugs. Phoebe Goldman created our system and adapted Professor Meyers' MAP scoring system to fit this study's needs. Julia Murillo researched background information on general and domain-specific information retrieval and wrote the majority of the paper. Katelyn Wang researched relevant corpora and analyzed the final test results; she also wrote the sections of the paper relevant to those topics.

References

Boteva, V., Gholipour, D., Sokolov, A., & Riezler, S. (2016, March). A full-text learning to rank dataset for medical information retrieval. *European Conference on Information Retrieval*, 716-722, Springer, Cham. doi: 10.1007/978-3-319-30671-1_58

Bird, S., Klein, E., & Loper, E. (2019). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit.* Retrieved online at http://www.nltk.org/book/.

Castells, P., Fernandez, M., & Vallet, D. (2018). An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Transactions on Knowledge and Data Engineering, 19*(2), 261-272. doi: 10.1109/TKDE.2007.22.

Chen, S., Fore, D., Strassel, S., Lee, H., & Wright, J. (2018). BOLT English SMS/Chat. *Linguistic Data Consortium*. https://doi.org/10.35111/1whw-ea40.

Explosion (n.d.). s*paCy 101: Everything you need to know.* spaCy. https://spacy.io/usage/spacy-101.

Fautsch, C. & Savoy, J. (2010). Adapting the tf idf vector-space model to domain specific information retrieval. *SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing*, 1708-1712. https://doi.org/10.1145/1774088.1774454.

Manning, C. D., Raghavan, R. & Schütze, H. (2009). *Introduction to Information Retrieval*. Cambridge University Press. Retrieved online at https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf.

Meyers, A. (2021a). *Homework Number 4.* Personal Collection of A. Meyers, New York University, New York NY.

Meyers, A. (2021b). L*ecture 5: Information Retrieval and Terminology Extraction.* Personal Collection of A. Meyers, New York University, New York NY.

NLTK Project (2021). *NLTK: Documentation*. NLTK. https://www.nltk.org/

Pohorec, S., Verlič, M., & Zorman, M. (2009). Domain specific information retrieval system. *13th WSEAS International Conference on COMPUTERS*. https://www.researchgate.net/publication/228650503_Domain_specific_information_retrieval_system.

Scells, H., Locke, D., & Zuccon, G. (2018). An information retrieval experiment framework for domain specific applications. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval,* 1281-1284. https://doi.org/10.1145/3209978.3210167.

van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworth-Heinemann. Retrieved online at http://www.dcs.gla.ac.uk/Keith/Preface.html.