Combining Parts of Speech, Term Proximity, and Query Expansion for Document Retrieval

Eric LaBouve* Lubomir Stanchev†
Department of Computer Science and Software Engineering
California Polytechnic State University
San Luis Obispo, CA, USA
*elabouve@calpoly.edu †lstanche@calpoly.edu

***Abstract*—Document retrieval systems recover documents from a database and order them according to their perceived relevance to a user's search query. This is a difficult task for machines to accomplish because there exists a *semantic gap* between the meaning of the terms in a user's literal query and a user's true intentions. The main goal of this study is to modify the Okapi BM25 document retrieval system to improve search results for textual queries and unstructured, textual corpora. This research hypothesizes that Okapi BM25 is not taking full advantage of the structure of text inside documents. This structure holds valuable semantic information that can be used to increase the model's accuracy. Modifications that account for a term's part of speech, the proximity between a pair of related terms, the proximity of a term with respect to its location in a document, and query expansion are used to augment Okapi BM25. The study resulted in 87 modifications which were all validated using open source corpora. The top scoring modification from the validation set was then tested under the Lisa corpus and the model performed 10.25% better than Okapi BM25 when evaluated under mean average precision.**

Keywords: Semantic Analysis, Document Retrieval, Query Expansion, Term Proximity, Search, Okapi BM25

I. Introduction

One of the most pervasive document retrieval engines in everyday society is Google's search engine. Google's search engine works well because documents on the Internet are highly structured with HTML elements and RDF triples that explicitly define the contents of web pages. These metadata elements are used as training data by Google's page rank algorithm [16] to score a document based on the document's popularity among other web pages. Although a thorough explanation of the page rank algorithm is outside the scope of this paper, due to the technical nature of the page rank algorithm, Google's search engine fails when users are trying to search for documents that are unpopular. This is a problem because documents can be both unpopular and relevant to the user's search query. As a result, popular documents are circularly discovered by many individuals and unpopular, yet relevant documents go unnoticed. The solution is to develop a system that scores a document based on its contents rather than its perceived popularity among other documents in the same corpus.

It is important to research alternative ways to rank documents for a handful of reasons. First, systems that primarily rely on training data will not operate well if the domain of the training data is disjoint from the domain of the deployment environment [3], [20]. Such systems would need to be retrained to maintain accuracy, which can take a significant period of time and substantial effort from developers to collect and clean training data. Okapi BM25 does not require any training data, so it can be plugged into any corpus and operate effectively. Second, the structure of a corpus cannot always be assumed. Not all documents inside a corpus contain structured content that a system can use to its advantage, such as HTML elements, RDF triples, or bibliographic citations [22]. A system which is agnostic to any preexisting document structure would be a powerful tool for search engines because it could be applied to a larger set of corpora. Third, a document retrieval system that can operate effectively in any environment would speed up development for minimum viable products because developers would not have to spend time gathering training data or introducing metadata into unstructured documents to support structured search.

Ranking documents is a difficult and unsolved problem for many reasons. First, users are unpredictable because they have different expectations for the degree of relevance and ordering of documents across every query. Second, there commonly exists a mismatch between the meaning of the user's literal search query and the true meaning behind what the user intended to type. This mismatch between the user's query and the user's desire is known as the *semantic gap*. Closing the *semantic gap* is a fundamental area of research in nlp [23], [12], [13], [9]. Third, understanding the true nature behind a user's query is made more difficult because languages are dynamic with respect to time and location. For example, text that reads, Eric likes chips, can be interpreted in America as, Eric likes tortilla chips, whereas a user from Australia might had intended the statement to mean, Eric likes French fries, since French fries are called chips in Australia. Fourth, search engines are expected to return relevant documents quickly despite the fact that information is growing exponentially. So data structures and algorithms must be built to perform efficiently. This becomes a huge problem when there are millions or billions of documents. Therefor system designers must decide if it is reasonable to search through every document in the corpus or just a subset of documents.

Developing the perfect document retrieval system is an unsolvable problem because building a perfect system is much

like designing a black box. In real world circumstances, the true relevance rating for a document, according to a query, is not observable. As described in Section II, previously proposed upgrades to Okapi BM25 are inadequate because they only optimize the model on one theme of parameters, such as term proximity or query expansion. Little research has been done to optimize Okapi BM25 across many modification themes. If successfully created, such a model would be able to take advantage of a wider variety of contextual information inside a document. This research hypothesizes that Okapi BM25 can be modified to take advantage of many contextual themes such as a term's part of speech, the proximity of related terms to each other, the proximity of terms within a document, and terms derived from query expansion to improve the model's accuracy.

The task of improving Okapi BM25 was approached by first designing a large number of potential modifications that could be incorporated into the model. These modifications were then combined and validated against four datasets: Cranfield, Adi, Medline, and Time. The best performing model was then tested against the Lisa corpus. The results show that the top performing model positively increases the mean average precision of the original Okapi BM25 model by 10.25% and takes advantage of parts of speech, term to term proximity, term to document proximity, and query expansion.

II. Related Research

The following related research are attempts to improve Okapi BM25 [19], which is shown in Equation 1. The model takes two inputs, a document vector d_j and a query vector q, and loops through each term t_i that appears in both the document and the query. The score for a term is the product of the inverse document frequency idf_i, the term frequency tf_{ij}, and the query term frequency qtf_i. The overall score for a pair of vectors is the sum of the values generated for each term t_i. Higher scores indicate that two vectors are similar to each other and lower scores indicate that two vectors are dissimilar to each other.

$$OkapiBM25(d_j, q) = \sum_{t_i \in q, d_j} idf_i \times tf_{ij} \times qtf_i \qquad (1)$$

A noteworthy attempt to expand Okapi BM25 was conducted by Cummins *et al.* [7], [6]. The researchers used a genetic algorithm to evolve the model with 12 functions that measured the distances between terms, the product between term frequencies, and the length of a document. Their algorithm favored mutations that resulted in higher mean average precision scores when ran against 69,500 documents and 55 queries. The top three models produced using their genetic algorithm all used a minimum distance proximity measure between pairs of terms and an average distance proximity measure between pairs of terms.

Bhatia *et al.* categorized the semantic relationship between pairs of terms in multi-term queries [4]. The two categories that Bhatia *et al.* defined were "topic modifying" and "topic collocating." Topic modifying is where one query term represents a subject and the other query term modifies that subject. Topic collocating is where multiple query terms represent a single topic. Their research concluded that emphasizing the proximity of topic modifying and topic collocating terms can result in a more precise system and that the appearance of a single topic modifying or topic collocating term holds little semantic meaning unless it is found next to its related term(s). Unfortunately, Bhatia *et al.* did not suggest a technique to identify topic modifying and topic collocating terms inside documents or queries. As a result, the research presented in this paper attempts to identify topic modifying and topic collocating terms by their parts of speech.

Some researchers have found success when analyzing spans, which are segments of text from a document that incorporate all query terms, or a subset of the query terms. Successful experiments by Rafique *et al.* utilize an equation that assigns a greater reward to documents when the first occurrence of each query term appears in close proximity to each other [18]. The researchers designed 25 custom queries and their results show an increase in mean average precision by around 15%. Other research by Song *et al.* introduced a technique to replace term frequency with an algorithm that splits a documents into non overlapping segments that contain one or more query terms [21]. In other words, the term frequency tf_{ij} is modified to measure the density of non overlapping spans. Their system was tested on TREC disks 9, 10, and 11 and their results show an increase in precision when observing the first five and ten documents returned by around 0.3% for disk 9, 10.4% for disk 10, and 4.4% for disk 11.

Other successful modifications take advantage of a term's position in a document. BM25F [17] considers a term to be more relevant to a query depending on its location within HTML tags. The relevance scores for BM25F are computed similarly to Okapi BM25, with the addition that for each term, a heuristically set hyper parameter is used to scale up or scale down the term's score depending on the term's surrounding HTML tags. Unfortunately, BM25F requires HTML metadata, which is limited to web documents. With the success of BM25F, researchers Blanco *et al.* built a model to generalize BM25F to unstructured text [5]. Their approach splits a document into "virtual regions," much like a spans. Terms in these regions are weighted proportionally to the section's statistical significance. Their algorithm was tested against five large corpora containing 95 million documents and reported a consistent increase in mean average precision over both Okapi BM25 and BM25F.

The last significant theme of modifications used to modify Okapi BM25 is query expansion. Query expansion is an attempt to add related terms to a query in order to express the original query in a more detailed way so that more relevant documents can be identified. There are three major areas of query expansion as identified by Ooi *et al.* [15]: query expansion using corpus dependent knowledge models, query expansion using relevance feedback, and query expansion using language models. Fox's research [8] on corpus dependent

knowledge models demonstrates that recall can be improved if the knowledge model is built from a corpus that shares the same lexical relations as the test corpus. Similarly, research by Vechtomova's *et al.* suggests that systems will perform worse if they use knowledge models constructed from a global point of view. As a result, they advise that document retrieval systems use models constructed from local points of view [24]. The same results were obtained by other researchers such as Kuzi *et al.* who were able to improve mean average precision using language models trained on the same corpus that their document retrieval system was analyzing [10]. However, other research [2] claims that query expansion altogether is a bad idea because it will inevitably hurt a system's recall due to vocabulary mismatch or a system's precision due to topic drift [1]. These results were shown from both a global and a local point of view and implies that query expansion is not an effective mechanism for improving document retrieval systems.

The presented related research shows that document retrieval systems can be improved when optimized for a single theme of modifications such as term to term proximity, semantic analysis, term to document proximity, or query expansion. There is limited research on ways to combine all the aforementioned themes into a unified system. The remainder of this paper will demonstrate how a variety of modification themes can be combined to improve the Okapi BM25 document retrieval system.

III. Solution / Implementation

This section discusses how documents are stored, details how Okapi BM25 is extended to enable many modifications, and describes the set of modifications that were designed and used to produce the final system.

A. The Inverted Index

In order to blend multiple modification themes, an inverted index is built to efficiently access information inside documents. A basic inverted index maps terms found inside a corpus to a list of postings. Each posting corresponds to a single document and holds the term frequency for the key term in this document. Postings are extended in this research to include the key term's absolute word positions, the absolute sentence positions, and the corresponding parts of speeches. This information is stored in three separate lists whose sizes are equal to the number of times the key term appears within the document. Note that keeping a list for the parts of speech is necessary because a term can take on multiple parts of speech depending on context. To simplify later contextual analysis, words are assumed to only be nouns, verbs, adjectives, or adverbs. Finally, to reduce the number of keys in the inverted index, stop words are filtered out and the remaining words are stemmed using a Porter Stemmer. Figure 1 provides the structure of the inverted index.

B. Extending Okapi BM25

With the construction of a more intimate inverted index, a more extensible version of Okapi BM25 can be built to utilize many modifications. The score generated for a single term is modified to include a collection of boosts. Each activated modification contributes a single boost value (except in the case of query expansion which will be discussed later) and boosts are added to a term's score once all modifications have been executed. The weight of a boost for a single modification on a query term is important to consider. It is not sufficient to simply add a constant value to the term's score because a term's significance to the overall query may very depending on context. Adding a constant value could even negatively harm the system's performance for short queries because static values will represent a high proportion of a short query's overall score, thus over-weighting the modification. Therefor, a term's score is boosted proportionally to the absolute value of the term's unmodified Okapi BM25 score. The absolute value must be taken because the Okapi BM25 formula is logarithmic and can result in negative numbers if a term has a high document frequency. Equation 2 gives the boosting function used for all modifications. In Equation 2, *OkapiScore* is the original score calculated from Okapi BM25 and *Influence* is a modification specific value that is responsible for scaling a term's score. *Influence* values range from zero to two and are either determined through training, chosen heuristically, or computed algorithmically. Only once all modifications are executed for a particular term does the system reduce the list of boosts and add the total to the term's original Okapi BM25 score.

$$Boost = (|OkapiScore| \times Influence) - |OkapiScore| \quad (2)$$

It is worth mentioning that some modifications require precomputation. For example, one modification performs query expansion on terms that score the highest inverse document frequencies. The reasoning behind this modification is that terms that have high inverse document frequencies are relatively rare within the corpus and exhibit high significance to the query. Therefor, such terms may make good candidates for query expansion. To determine which terms to expand from the query, the inverse document frequency values must be known ahead of time. Similarly, other modifications enforce that the query be traversed in the order in which the terms are listed in the query.

Key Term → [<Document ID, Term Frequency, [Sentence Indices], [Absolute Indices], [Parts of Speech]>, ...]

Fig. 1. The structure of the inverted index. Angle brackets represent a posting and ... represents postings that are not shown in this diagram.

C. Proposed Modifications

This section describes relevant algorithms and equations that are used to derive a modification's *Influence* score. The modifications have been categorized into four themes to help clarify the overall explanation. The first theme analyzes a term's part of speech in isolation. The second theme includes modifications that analyze the distance between pairs of query terms found within a document. The third theme analyzes the

position of a single query term with respect to its location within a document. Finally, the fourth theme explores methods for query expansion.

1) Parts of Speech: The simplest set of modifications introduced in this research is to scale the *Influence* of an individual term according to its part of speech. *Influence* values for this suite of modifications are set after training the modifications on the Cranfield corpus until a local maximum mean average precision value is reached. Since it is unknown whether or not the *Influence* value should be greater than one or less than one for a particular part of speech, each part of speech is paired with two modifications. The first modification has an *Influence* value that is greater than one and the second modification has an *Influence* value that is less than one. These two sets of modifications are categorized as "Boost Up" and "Boost Down" respectively. Table I summarizes all of the part of speech modifications.

TABLE I
PARTS OF SPEECH THEMED MODIFICATIONS. I IS SHORT FOR *Influence*.

Category	Affects	Parameters	Summary
Boost Up	Nouns	$I > 1$	Nouns have higher *Influence*
Boost Up	Adjectives	$I > 1$	Adjectives have higher *Influence*
Boost Up	Verbs	$I > 1$	Verbs have higher *Influence*
Boost Up	Adverbs	$I > 1$	Adverbs have higher *Influence*
Boost Down	Nouns	$I < 1$	Nouns have lower *Influence*
Boost Down	Adjectives	$I < 1$	Adjectives have lower *Influence*
Boost Down	Verbs	$I < 1$	Verbs have lower *Influence*
Boost Down	Adverbs	$I < 1$	Adverbs have lower *Influence*

2) Term to Term: Refining the idea of topic modifying and topic collacating terms from [4], this paper assumes that pairs of terms are considered related when the first term is either an adjective or adverb and the second term is either a noun or a verb. The idea behind this assumption is that a modifying term, such as an adjective or adverb, contains the most semantic meaning if found next to or near its corresponding subject. For example, in the query "Red cars for sale," the term "red" is semantically insignificant if it is found in a document that does not also contain the word "car."

Three different categories of modifications are built to evaluate pairs of semantically related terms. Shown in Table II under "Modifiers", the first two modifications exclude the score from adjective and adverb query terms unless the term that immediately follows in the document is the query's subject. These two modifications do not require *Influence* values because they are simply removing the scores from adjectives and adverbs that are not found next to their corresponding subjects.

The second set of modifications can be found under the "Bigrams" category of Table II. This set of modifications considers the possibility that completely removing a term's impact from a document's score might have a negative consequence on a system's overall accuracy. Instead, the "Bigrams" category rewards a document for containing pairs of adjacent query terms. Unlike the "Modifiers" category, the "Bigrams" category does not exclude any term scores. The bigrams are constructed from one of three ways: all adjacent query terms, only adjacent adjective and noun query terms, or only adjacent adverb and verb query terms. Since the significance of a bigram is unknown, the *Influence* values for bigram modifications are computed by training the modifications on the Cranfield dataset until local maximum mean average precision values between one and two are discovered. The boost value is then applied to the subject.

The third set of modifications are located under the "Close Pairs" category in Table II and are designed to boost modifiers and subjects that may not appear directly adjacent to each other. For example, in the query "Red and blue cars for sale," the term "red" does not appear right next to the term "car." However, the term "red" still modifies the term "car." When there exists a separation between the modifier and the subject, Equation 3 is used to determine the appropriate *Influence* value between the two query terms. The boost is applied to the subject.

$$Influence = max(m * x + (b - m), 1) \tag{3}$$

In Equation 3, x is an integer value that represents the minimum distance between a pair of query terms found within a document. The minimum value of x is equal to one because if two terms are found right next to each other, the difference between their absolute locations is equal to one. m is a negative value that represents the rate at which the reward for two terms should diminish. Equation 3 does not apply to modifiers and subjects that span across multiple sentences to avoid situations where related query terms may appear near each other but in unrelated contexts. For example, if the modification is searching for instances where the terms "red" and "car" appear close to one another, the sentence "She has red hair. Her car is blue" would not be considered by the modification. Finally, $(b-m)$ is the y intercept for the function. The y intercept is adjusted for the fact that when x is equal to one, the value of the function is equal to b.

3) Term to Document: When scoring documents, it is not only important to gather documents that relate to the query somewhere within the document. It is also important to gather documents that relate to the query at the front of the document. The proposition is that if users expect relevant information to appear at the start of documents, then a term's score should be positively rewarded for occurring earlier in a document. Table III contains modifications that reflect this proposition. Each modification in Table III uses Equation 4 to reward terms based on a term's first occurrence in a document.

$$Influence = \frac{K * dl_j - idx_i}{dl_j} \tag{4}$$

Equation 4 is a linear function where dl_j is the document length of document j, which is measured as the sum of all the terms in the document. idx_i is the absolute index location of term i, where the first term in the document has an idx_i value of zero. K is an integer hyper parameter which dictates the upper bound for the function. During experiments, K

TABLE II
TERM TO TERM THEMED MODIFICATIONS. I IS SHORT FOR *Influence*.

Category	Affects	Parameters	Summary
Modifiers	Adjectives	N/A	Ignore adjectives not found next to nouns
Modifiers	Adverbs	N/A	Ignore adverbs not found next to verbs
Bigrams	All	$I > 1$	Reward a document for containing any adjacent bigrams from the query
Bigrams	Adjectives and Nouns	$I > 1$	Reward a document for containing adjacent adjective, noun bigrams
Bigrams	Adverbs and Verbs	$I > 1$	Reward a document for containing adjacent adverb, verb bigrams
Close Pairs	All	$I = max(-0.25x + 2, 1)$	Reward a document for containing terms that are close together
Close Pairs	Adjectives and Nouns	$I = max(-0.25x + 2, 1)$	Reward a document for containing adjectives and nouns that are close together
Close Pairs	Adverbs and Verbs	$I = max(-0.25x + 2, 1)$	Reward a document for containing adverbs and verbs that are close together

is heuristically set to two because terms at the front of a document might be twice as important as terms that appear at the end of a document. Notice that the function never penalizes for a term's position. At the very worst case, a term's score is unmodified if it is located at the very end of a document.

TABLE III
TERM TO DOCUMENT THEMED MODIFICATIONS. I IS SHORT FOR *Influence*.

Category	Affects	Parameters	Summary
Is Early	All	$I = \frac{2 \times dl_j - idx_i}{dl_j}$	Rewards terms that appear early in a document
Is Early	Nouns	$I = \frac{2 \times dl_j - idx_i}{dl_j}$	Rewards nouns that appear early in a document
Is Early	Adjectives	$I = \frac{2 \times dl_j - idx_i}{dl_j}$	Rewards adjectives that appear early in a document
Is Early	Verbs	$I = \frac{2 \times dl_j - idx_i}{dl_j}$	Rewards verbs that appear early in a document
Is Early	Adverbs	$I = \frac{2 \times dl_j - idx_i}{dl_j}$	Rewards adverbs that appear early in a document

4) Query Expansion: This research explores global query expansion methods so that the resulting model can be portable to any corpus. Three methods for query expansion are implemented, the first uses the WordNet API, the second uses a graph generated through WordNet, and the third uses word embeddings generated from Word2Vec.

The first method for query expansion uses the APIs exposed by WordNet. WordNet is a project by Princeton University that organizes the semantic relationships between English terms into cognitive synonyms [14]. Using the APIs is non-trivial because words may have multiple definitions and parts of speech. In order to look up the correct word in WordNet, the Lesk algorithm [11] is implemented to perform word sense disambiguation. With a logical guess of the proper definition and part of speech of a term, cognitive synonyms can then be extracted from WordNet. Unfortunately, WordNet does not provide the strength of similarity between a term and its cognitive synonyms, so we set the *Influence* value for all WordNet API expansion terms to 0.9. The assumption is that expansion terms have a slightly lower probability of being relevant than the original query term, but still maintain a high *Influence* value because they are cognitive synonyms. With the *Influence* value set, a query term's score is computed as the sum of the term's original Okapi BM25 score, plus a collection of boost values for each expansion term. In order to generate the boost values for each expansion term, Equation 2 is modified so that the $OkapiScore$ variable is the score generated by Okapi BM25 applied to each expansion term. Table IV provides a summary of the WordNet API modifications. All three query expansion categories will contain modifications that not only target all available query terms, but will also target nouns, verbs, adjectives, adverbs, low inverse document frequency terms, and high inverse document frequency terms.

TABLE IV
WORDNET API MODIFICATIONS FOR THE QUERY EXPANSION THEME. IDF IS SHORT FOR INVERSE DOCUMENT FREQUENCY. I IS SHORT FOR *Influence*.

Category	Affects	Parameters	Summary
WordNet API	All	I = 0.9	Expand all query terms
WordNet API	Nouns	I = 0.9	Expand query terms that are nouns
WordNet API	Verb	I = 0.9	Expand query terms that are verbs
WordNet API	Adjectives	I = 0.9	Expand query terms that are adjectives
WordNet API	Adverbs	I = 0.9	Expand query terms that are adverbs
WordNet API	Low IDF	I = 0.9	Expand query terms that score a low inverse document frequency value
WordNet API	High IDF	I = 0.9	Expand query terms that score a high inverse document frequency value

Although WordNet does not quantify the similarity between terms, recent research shows that similarity scores can be calculated if the WordNet database is arranged in a probability graph [23]. Semantically similar terms can then be discovered from the probability graph by computing random walks from the node that represents the unexpanded query term. A random walk is performed by stochastically selecting an out edge (proportionally to the edge's weight) and traversing to the node pointed to by this edge. This process is repeated once more to reach a depth of two. The nodes that are traversed the most often after running the random walk algorithm 1000 times represent the semantically similar terms. The similarity score between the unexpanded term and an expansion term

is then the proportion of times the expansion term's node was visited after 1000 random walks. This proportion is used as the expansion term's *Influence* value. A minimum acceptable similarity score between the query term and the expanded terms is heuristically set in order to avoid query drift. Table V summarizes the WordNet Graph query expansion modifications. Notice that the modifications target various parts of speech and inverse document frequencies.

TABLE V
WORDNET PROBABILITY GRAPH MODIFICATIONS FOR THE QUERY EXPANSION THEME. RWSS IS SHORT FOR RANDOM WALK SIMILARITY SCORE. I IS SHORT FOR *Influence*.

Category	Affects	Parameters	Summary
WordNet Graph	All	I =RWSS	Expand all query terms
WordNet Graph	Nouns	I = RWSS	Expand query terms that are nouns
WordNet Graph	Verb	I = RWSS	Expand query terms that are verbs
WordNet Graph	Adjectives	I = RWSS	Expand query terms that are adjectives
WordNet Graph	Adverbs	I = RWSS	Expand query terms that are adverbs
WordNet Graph	Low IDF	I = RWSS	Expand query terms that score a low inverse document frequency value
WordNet Graph	High IDF	I = RWSS	Expand query terms that score a high inverse document frequency value

The last query expansion category that is implemented uses English word vectors generated using the Word2Vec algorithm [12], [13]. The vectors are 300 dimensions large and were generated from the Google News corpus[1], which is a large database of text containing three billion running words and about three million unique words. The main advantage to using word vectors for query expansion is that the similarity between any two word vectors can be computed using the cosine similarity function. The resulting similarity score is used as the *Influence* value. In order to find the top similar word vectors for a given term, the cosine similarity is computed between the query term and all other word vectors. Since this is a time consuming procedure, query expansion terms are computed ahead of time. Lastly, a lower bound similarity score of 0.5 is heuristically set to avoid query drift. A summary of Word2Vec modifications can be found in Table VI.

IV. EXPERIMENTAL PROCEDURE

Modifications are validated against four publicly available information retrieval benchmarks[2]: Cranfield, Adi, Medline, and Time. Each benchmark contains a set of documents, a set of queries, and an exhaustive list of relevance scores for all query-document pairs. In total, there are just under 3,000 documents and 35 queries in the validation set. The Lisa benchmark is used as the testing set and contains 5,872 documents and 370 queries.

[1]GoogleNews corpus: https://github.com/mmihaltz/word2vec-GoogleNews-vectors

[2]Available at http://ir.dcs.gla.ac.uk/resources/test_collections

TABLE VI
WORD2VEC MODIFICATIONS FOR THE QUERY EXPANSION THEME. CSS IS SHORT FOR COSINE SIMILARITY SCORE. I IS SHORT FOR *Influence*.

Category	Affects	Parameters	Summary
Word2Vec	All	I = CSS	Expand all query terms
Word2Vec	Nouns	I = CSS	Expand query terms that are nouns
Word2Vec	Verb	I = CSS	Expand query terms that are verbs
Word2Vec	Adjectives	I = CSS	Expand query terms that are adjectives
Word2Vec	Adverbs	I = CSS	Expand query terms that are adverbs
Word2Vec	Low IDF	I = CSS	Expand query terms that score a low inverse document frequency value
Word2Vec	High IDF	I = CSS	Expand query terms that score a high inverse document frequency value

The experimental procedure is split up into three validation rounds and one testing round. During the first validation round, all 41 modifications described in the previous sections are ran independently on the four validation corpora. The modifications that result in a higher mean average precision than the original Okapi BM25 model on at least two validation benchmarks will then be used in validation rounds two and three.

Round two combines modifications within the same theme. An example modification that can be validated in this round could be a query expansion modification that uses WordNet API, WordNet Graph, and Word2Vec. This new set of modifications is then validated in the same way as round one and the most successful modifications are used in validation round three.

Round three combines modifications from rounds one and two from across multiple themes. An example modification from this round might use query expansion on WordNet API, WordNet Graph, and Word2Vec, boost terms based on their proximity within a document, boost terms based on their part of speech, and boost a document's score when query bigrams appear within a document. These modifications are then validated against Cranfield, Adi, Medline, and Time.

Once all three validation rounds are completed, a single modification across all three rounds is selected to be tested. The best performing modification is the one which resulted in the greatest sum of the differences between the modification's mean average precision and the unmodified Okapi BM25's mean average precision across each validation benchmark. Equation 5 is the precise definition for how a modification's score is calculated. Allow B to be the set of validation benchmarks {Cranfield, Adi, Medline, Time}, mod to be a modification, and $MAP(m, b)$ to be the mean average

precision value of model m applied to benchmark b.

$$\sum_{b \in B} (MAP(mod, b) - MAP(Okapi\ BM25, b)) \quad (5)$$

V. Results

In validation round one, query expansion themed modifications had minor effects on the model's mean average precision because only a few expansion terms were discovered for each query. In validation round two, expansion techniques from WordNet API, WordNet Graph, and Word2Vec were combined to increase the number of expansion terms discovered for each query. Unfortunately, combining all three expansion techniques lead to a decrease in the model's mean average precision, most likely due to topic drift. The two best performing query expansion modifications were the WordNet Graph modifications that expanded nouns and bottom inverse document frequency terms. These two modifications resulted in small increases to the model's mean average precision on the Adi benchmark and almost negligible affects on the rest of the validation benchmarks.

The term to document themed modifications had the largest and most positive impact on the validation benchmarks. Across validation rounds one and two, term to document modifications increased the mean average precision against three out of the four validation benchmarks. After counting the occurrences of parts of speech given by python's nltk library, documents on average were composed of 50% nouns, 20% verbs, 20% adjectives, 5% adverbs, and 5% miscellaneous parts of speech. Generally, modifications that targeted parts of speech that compose a larger majority of document terms gave rise to larger swings in the accuracy, and vice versa. Although there is no clear best modification, the first top scoring modification targeted all parts of speech and the second top scoring modification targeted nouns and adjectives.

Across validation rounds one and two, some part of speech themed modifications performed better when the *Influence* of a particular part of speech was decreased, such as the adjectives and adverbs, and other modifications performed better when the *Influence* of a particular part of speech was increased, such as the nouns. The best performing modification from this theme positively boosted nouns and adjectives and left the verbs and adverbs unchanged. Generally, precision was positively affect when nouns and adjectives were boosted up and adverbs were boosted down.

Lastly, term to term themed modifications produced relatively small changes in mean average precision. This might be because the probability of two terms with specific parts of speech appearing chronologically near each other in a document is a rare event. The "Modifiers" modification category performed especially poorly. The "Bigrams" modification category performed a little better, but still did not improve the mean average precision on more than two benchmarks. The "Close Pairs" modification category performed poorly when analyzing all adjacent query terms, but resulted in slightly positive precision scores when the modification was limited to either adjectives and nouns or verbs and adverbs.

All together, 87 models were created across all validation rounds. From this set, the top scoring model was determined using Equation 5. After scoring was completed, it was discovered that the top scoring model was created in validation round three. This model combines three modification themes. From the query expansion theme, the model uses the WordNet Graph to expand the bottom inverse document frequency query terms. From the term to document theme, the model rewards nouns and adjectives for appearing closer to the start of a document. Lastly from the term to term theme, the model rewards adjectives and nouns for occurring near each other inside documents. This model and the unmodified version of Okapi BM25 were then tested using the Lisa benchmark. When Okapi BM25 was ran against Lisa, the resulting mean average precision value was 0.357 and when the top model was ran against Lisa, the top model scored a mean average precision value of 0.393. The difference between these results represents a 10.25% improvement. After inspecting the Lisa set for potential biases, document titles were removed from the benchmark to keep inherent document structure to a minimum. Both models were then reran against Lisa. This time, Okapi BM25 scored a mean average precision value of 0.304 and the top model scored a mean average precision value of 0.326, representing a 7.31% increase in performance.

The performances between the two models can also be compared using weighted average recall, where the recall scores are weighted proportionally to the number of relevant documents in each query. When ran against Lisa, Okapi BM25 returned recall scores of 0.145, 0.237, and 0.332 and the modified model returned recall scores of 0.155, 0.224, and 0.343 on the first 5, 10, and 20 documents returned for each query. From the first 5 document returned, the top model obtained a recall that was 7.27% better than Okapi BM25. Then the recall dipped bellow Okapi BM25 once 10 documents were returned by around -5.56%. However in the long term, after 20 documents were returned, the top model returned a recall score that was 3.17% better than Okapi BM25.

The performances of both Okapi BM25 and the top model can be displayed on a precision-recall curve to gain more granular insight into how the mean average precision is affected by the weighted average recall. In Figure 2, the blue line represents Okapi BM25 and the red line represents the top model. From the graph, it is clear that the modified system scores a higher mean average precision value than Okapi BM25 at all recall levels, except at the recall range between 0.08 and 0.12. Despite this small range of values, the top model consistently outperforms the original Okapi BM25 model at short term and long term recall levels.

VI. Conclusion and Future Research

We have demonstrated a process to derive and validate many modifications for Okapi BM25. From the models that were created, the best performing model was selected and tested against the Lisa benchmark. This model combines query expansion, term to term proximity, and term to document

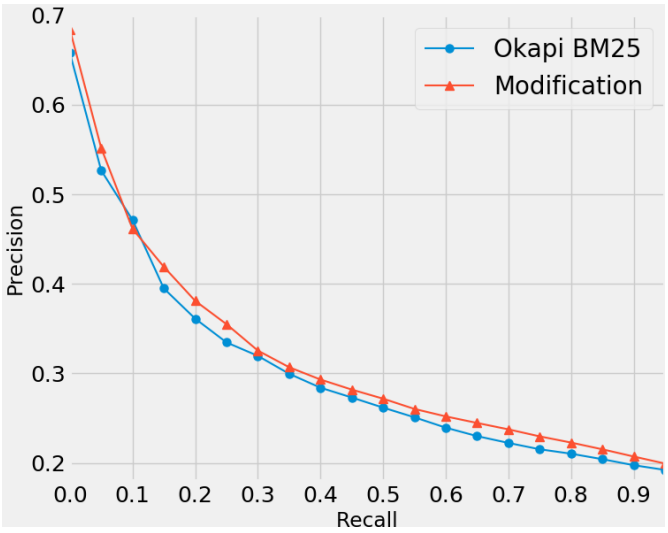

Fig. 2. Precision-recall curves for Okapi BM25 and the top model at recall bucket sizes 0.1.

proximity across various parts of speech. In conclusion, a model that combines many modification themes can be built to outperform Okapi BM25 in mean average precision and weighted average recall for most recall levels.

One area for future research would be to extend Okapi BM25 to take advantage of more sophisticated natural language processing and grammar rules. For example, conjunction words can be identified to help locate the main subject of multi-clause sentences. The subject terms can then be weighted proportionally to their perceived significance.

References

[1] B. Al-Shboul and S. H. Myaeng. Analyzing topic drift in query expansion for information retrieval from a large-scale patent database. In *2014 International Conference on Big Data and Smart Computing (BIGCOMP)*, pages 177–182, Jan 2014.

[2] A. Bakhtin, Y. Ustinovskiy, and P. Serdyukov. Predicting the impact of expansion terms using semantic and user interaction features. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 1825–1828, New York, NY, USA, 2013. ACM.

[3] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, 6(1):20–29, June 2004.

[4] M. P. S. Bhatia and A. Kumar. Contextual paradigm for ad hoc retrieval of user-centric web data. *IET Software*, 3(4):264–275, August 2009.

[5] R. Blanco and P. Boldi. Extending bm25 with multiple query operators. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 921–930, New York, NY, USA, 2012. ACM.

[6] R. Cummins and C. O'Riordan. Evolving local and global weighting schemes in information retrieval. *Inf. Retr.*, 9(3):311–330, June 2006.

[7] R. Cummins and C. O'Riordan. Learning in a pairwise term-term proximity framework for information retrieval. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 251–258, New York, NY, USA, 2009. ACM.

[8] E. A. Fox. Lexical relations: Enhancing effectiveness of information retrieval systems. *SIGIR Forum*, 15(3):5–36, Dec. 1980.

[9] K. Jain, A. Jain, T. Srivastava, and NSS. Intuitive understanding of word embeddings: Count vectors to word2vec, Jun 2017.

[10] S. Kuzi, A. Shtok, and O. Kurland. Query expansion using word embeddings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 1929–1932, New York, NY, USA, 2016. ACM.

[11] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, pages 24–26, New York, NY, USA, 1986. ACM.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[14] G. A. Miller. Wordnet: A lexical database for english, 1995.

[15] J. Ooi, X. Ma, H. Qin, and S. C. Liew. A survey of query expansion, query suggestion and query refinement techniques. In *2015 4th International Conference on Software Engineering and Computer Systems (ICSECS)*, pages 112–117, Aug 2015.

[16] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[17] J. R. Pérez-Agüera, J. Arroyo, J. Greenberg, J. P. Iglesias, and V. Fresno. Using bm25f for semantic search. In *Proceedings of the 3rd International Semantic Search Workshop*, SEMSEARCH '10, pages 2:1–2:8, New York, NY, USA, 2010. ACM.

[18] M. I. Rafique and M. Hassan. Utilizing distinct terms for proximity and phrases in the document for better information retrieval. In *2014 International Conference on Emerging Technologies (ICET)*, pages 100–105, Dec 2014.

[19] S. Robertson and S. Walker. Okapi/keenbow at trec8. In *The Eighth Text REtrieval Conference (TREC8)*, page 151162. Gaithersburg, MD: NIST, January 2000.

[20] H. Sanders and J. Saxe. Garbage in, garbage out: How purportedly great ml models can be screwed up by bad data. Technical report, July 2017.

[21] R. Song, J.-R. Wen, and W.-Y. Ma. Viewing term proximity from a different perspective. Technical report, May 2005.

[22] L. Soulier, L. Ben Jabeur, L. Tamine, and W. Bahsoun. Bibrank: A language-based model for co-ranking entities in bibliographic networks. In *Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '12, pages 61–70, New York, NY, USA, 2012. ACM.

[23] L. Stanchev. Creating a similarity graph from wordnet. In *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*, WIMS '14, pages 36:1–36:11, New York, NY, USA, 2014. ACM.

[24] O. Vechtomova, S. Robertson, and S. Jones. Query expansion with long-span collocates. *Inf. Retr.*, 6(2):251–273, Apr. 2003.