Published in final edited form as:

J Am Soc Inf Sci Technol. 2009 February ; 60(2): 264–274. doi:10.1002/asi.20979.

How to Interpret PubMed Queries and Why It Matters

Lana Yeganova,
Contractor, Computational Biology Branch, National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bldg. 38A, 8600 Rockville Pike, Bethesda, MD 20894

Donald C. Comeau,
Computational Biology Branch, National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bldg. 38A, 8600 Rockville Pike, Bethesda, MD 20894

Won Kim, and
Computational Biology Branch, National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bldg. 38A, 8600 Rockville Pike, Bethesda, MD 20894

W. John Wilbur
Principal Investigator, Computational Biology Branch, National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bldg. 38A, 8600 Rockville Pike, Bethesda, MD 20894

Abstract

A significant fraction of queries in PubMed™ are multiterm queries without parsing instructions. Generally, search engines interpret such queries as collections of terms, and handle them as a Boolean conjunction of these terms. However, analysis of queries in PubMed™ indicates that many such queries are meaningful phrases, rather than simple collections of terms. In this study, we examine whether or not it makes a difference, in terms of retrieval quality, if such queries are interpreted as a phrase or as a conjunction of query terms. And, if it does, what is the optimal way of searching with such queries. To address the question, we developed an automated retrieval evaluation method, based on machine learning techniques, that enables us to evaluate and compare various retrieval outcomes. We show that the class of records that contain all the search terms, but not the phrase, qualitatively differs from the class of records containing the phrase. We also show that the difference is systematic, depending on the proximity of query terms to each other within the record. Based on these results, one can establish the best retrieval order for the records. Our findings are consistent with studies in proximity searching.

Introduction

This research project was initiated as a part of the continuous work at the National Library of Medicine™ (NLM) at the National Institutes of Health to improve information access and retrieval from MEDLINE™, a collection of approximately 17 million bibliographic records as of fall 2007. All records have titles and about half have abstracts. PubMed™ is a search

engine, developed and maintained by the National Center for Biotechnology Information at NLM, which works on MEDLINE™. Our major effort in this article is to examine queries and search approaches with the objective of providing MEDLINE™ users with improved retrieval results.

When processing a query, most search engines, by default, interpret a query as a collection of terms and use the logical AND operation on these terms to retrieve records containing the query terms. That search approach is called co-occurrence searching because it retrieves records where the query terms co-occur. The PubMed™ search engine is based on a co-occurrence search strategy.[1] For example, at the time of writing, PubMed™ translated the query *sudden death syndrome* to a search of *sudden* AND *death* AND *syndrome*. Some search engines, including PubMed™, allow a user to search for the exact query phrase when the query is in quotes. That search approach is called exact phrase searching.

A quoted query is an indication that users have a phrase in mind. With respect to an unquoted query, it is important to identify what users have in mind when they type in a query. Do they intend the query to be a meaningful phrase, or do they simply type terms describing the topic of interest in some random order? Search results may vary depending on how the query is interpreted by the search engine. For example, one could type in the query *sudden death syndrome*, looking for the records where the words *sudden*, *death*, and *syndrome* all occur. With the same query, one could also be looking for the records containing that exact phrase, rather than records where the query terms simply co-occur. At the time of writing, searching MEDLINE™ for the exact query phrase *sudden death syndrome* retrieved 174 records, while searching MEDLINE™ for a conjunction of query terms (*sudden* AND *death* AND *syndrome*) retrieved 6,237 records. Are the additional 6,063 records retrieved by the co-occurrence search as relevant to the query as the 174 records retrieved by the exact phrase search? What is the optimal retrieval approach?

Analysis of queries in PubMed™ demonstrates that the majority of queries are multiterm. Many of these multiterm queries include parsing indicators that could help the search engine determine whether the query is a phrase or not. We discuss these in more detail in the Description of Queries in PubMed™ section. However, a significant fraction of multi-term queries have no such indicators and consist of content words only. We refer to these as queries with no parsing indicators. These queries have no explicit indication of whether the query was intended to be a phrase or a collection of words.

In this article, we focus on multiterm queries with no parsing indicators and address the following two questions:

- What are these queries, phrases, or collections of terms?
- Does it really make any difference whether the query is handled as a phrase or as a collection of words, and, if it does, what is the optimal way to handle such a query?

[1]The Appendix has a detailed description of PubMed's retrieval process.

To address the first question, we studied multiterm queries submitted to PubMed™ that had no parsing indicators. A significant number of these queries are found in the MEDLINE™ records as exact phrases, suggesting that they could, indeed, be meaningful phrases. We analyzed whether the user intended the query to be a meaningful phrase by looking at the preference of the word order in the query. For example, if a user typed in a two-word query, *word1 word2*, and either of the queries *word1 word2* or *word2 word1* appear in published text, we found that it was predominantly the order *word1 word2* that was present in the published text. This suggests that users, in many cases, have a phrase in mind when they create a multiterm query with no parsing indicators. Based on these observations, we conclude that a reasonable fraction of multiterm queries with no parsing indicators are phrases and the users may expect or prefer retrieval of the same form.

To examine the second question of whether it makes a difference how the query is handled, we developed a machine learning based retrieval evaluation method. This machine learning technique is based on the assumption that the best answer to a query, which is a meaningful phrase, is a record containing that phrase. Using our evaluation method, we demonstrate that a machine learner can determine the patterns that describe the class of records containing the search phrase and the class of records containing the search terms, but not the phrase. Based on these patterns, the machine learner can distinguish between the classes. Moreover, we show that the distinction is systematic, depending on the proximity of search terms to each other. Based on term proximity, we can establish the best order of retrieval for the records.

In the next section, we study the queries submitted to PubMed™ to examine their composition. In the Retrieval Evaluation Framework and Justification section, we explain a general framework for our machine learning based retrieval evaluation method. The Retrieval Evaluation Experiments and Results section uses that method to compare several classes of records. We describe a series of experiments that reveal the differences between classes of records that correspond to various retrieval strategies. These results are consistent with findings in the area of proximity searching, which we review in the Proximity Searching section. In the Discussion section, we discuss our approach, and we draw conclusions in the Conclusions section.

Description of Queries in PubMed™

The goal of this section is to understand what the person issuing a query had in mind when he or she typed in a query. To do so, we examine the 14 million queries collected from the PubMed™ search engine during 5 days in June 2006. We refer to this set of 14 million queries as PMQ2006. This data has no user or time specific information. The graph in Figure 1 depicts the distribution of these queries.

In examining these PMQ2006 queries, we found that 15% are single word queries and 85% are multiword queries, which consist of character strings separated by spaces. We observed that some multiword queries contain strings, such as ‘AND,’ ‘[author],’ or ‘+,’ which suggest how to parse that query. We refer to queries that contain such strings as queries with parsing indicators. While we do not attempt to parse these queries, we point out that the indicators they contain could be used to parse a query. These queries represent 45% of all

queries and are discussed in detail in the next section. However, 40% of all queries had no such indicators but consisted of multiple content words only. What are such queries, exact phrases, or collections of terms? We look at these queries in detail in the Queries with No Parsing Indicators section in an attempt to understand how to interpret them. We find that many such queries are phrases, rather than collections of words.

Queries With Parsing Indicators

The queries with parsing indicators were those that included the Boolean operators (AND, OR, NOT), tags ([author], [mesh], [journal], etc.), quoted expressions, prepositions, conjunctions, and special characters (!@#$%^&*()=+;:'<>?\/).

Boolean operators AND, OR, or NOT appeared in 23% of the queries. We did not differentiate between the uppercase and lowercase operators. We observed that such queries had subqueries on both sides of an operator, which often were a concept or a meaningful phrase as opposed to a simple collection of words., such as, for example, the following query:

placenta abruption AND neonatal mortality

We found that 8% of queries were quoted or included PubMed™ tags, like '[author]," [journal],'or '[mesh],'which indicate an author search, a journal title search, or a search in the Medical Subject Headings (MeSH™), a controlled vocabulary developed by NLM. Examples are as follows:

"gastric gland" growth factor

"central venous pressure"[mesh]

In addition, 7% of the queries contained prepositions, which also indicated that the query was, most likely, a phrase. Examples are as follows:

induction of labor

recombination in bacteria

The remaining 7% of the queries in this group, contained one or more of the following special characters '!@#$%^&*(){}=+;:'<>?\/'. Most of these characters provided us with clues about the query. For example, very often the characters '&' or '+' were used instead of the Boolean operator AND, as in the following query:

p53 mutant + apoptosis

Similar to the queries that contained the Boolean operator 'AND,'queries with commas and semicolons looked like lists of words, author's names, or phrases separated by punctuation marks. Examples are as follows:

antioxidant, stress, pain

vitamin E, brain hemorrhage.

Queries With No Parsing Indicators

About 40% of the queries were multiword queries that had neither prepositions nor explicit Boolean operators, quotes, or any other non-alphanumeric characters. These were queries that consisted of content words only. Among this group of 40%, 22% were two-word queries, 11% were three-word queries, 5% were four-word queries, and the remaining 2% were queries with five or more words. Altogether, they represented more than 1 million queries per day. Should one handle such multiword queries as phrases or as Boolean conjunctions of query terms?

To examine these queries in more detail, we randomly chose 10,000 unique queries for each query length (two, three, and four words) from the PMQ2006 dataset. Then, given a query, we retrieved records from MEDLINE™ that contained the query terms and placed these records in one of the three non-overlapping categories as follows:

- *P*—exact phrase match—records containing the precise query
- *S*—sentence level co-occurrence—records containing the query terms within a single sentence but not the exact phrase
- *R*—record level co-occurrence—records containing the query terms in the record but not all of the terms in a single sentence

In this article, we do not perform query expansion, such as with MeSH™. Given this data, we computed the percentage of retrieved records that satisfy the exact phrase match, the percentage that satisfy sentence-level co-occurrence, and the percentage that satisfy record-level co-occurrence. Figure 2 summarizes the distribution of retrieval over these three categories, averaged over the 10,000 test queries.

Figure 2 shows that given a two-word query, on average, 29% of the retrieved records contain the exact query phrase, 32% of the records satisfy sentence-level co-occurrence, and the remaining 39% satisfy record-level co-occurrence. For three and four-word queries, the percentages of records that contain the exact query phrase declines. Similarly, the percentage of records that contain the query terms within a sentence declines. Finally, the percentage of records that satisfy record-level co-occurrence increases.

Next, we compute the percentage of queries that retrieved one or more records containing the precise query phrase. Each of these queries, if searched for as a phrase, would produce a nonempty retrieval. Similarly, we compute the percentage of queries that are not found as a phrase but that retrieved one or more records satisfying sentence-level co-occurrence. Finally, we compute the percentage of queries whose terms did not co-occur within a single sentence but that retrieved one or more records satisfying the record-level co-occurrence. Table 1 summarizes these percentages for the two-, three- and four-word queries. Note that these numbers do not indicate whether a particular query retrieved one or multiple records.

Results presented in Table 1 indicate that 54% of the two-word queries retrieved one or more records that contained the exact query phrase. In addition to these, 23% of the two-word queries retrieved one or more records where query terms appeared within a sentence. A further 3% of the queries retrieved one or more records that merely contained the query

terms. The remaining 20% of the queries did not retrieve any record that contained all of the query terms. This failure rate is higher than it would be in PubMed™ because we do not perform query expansion. Corresponding results for three- and four-word queries are also presented in Table 1. These results suggest that a substantial number of queries are meaningful phrases, especially for the shorter queries.

We further analyzed the two-word queries with no parsing indicators to determine whether the user intended the query to be an exact phrase. We looked at the significance of word order in the two-word queries. Given a two-word query, *word1 word2*, we retrieved records from MEDLINE™ that included either the original query, *word1 word2*, or the reversed query, *word2 word1*. For each such query, we compared the number of records that contained the phrase *word1 word2* with the number of records that contained the reverse phrase *word2 word1*. Based on the sample of 1,594 two-word queries that appeared at least 10 times in any order, we found that 90% of these preferred the *word1 word2* order, 7% preferred the reversed order, and there was no significant preference for the remaining 3%. In any single case, $p < 0.05$ using the sign test (Larson, 1982).

For the three- and four-word queries, switching the order of the query terms results in six combinations for the three-word queries and 24 combinations for the four-word queries. Nonetheless, we chose to consider only two possibilities for these longer queries. For the three-word case, we considered the original three-word query *word1 word2 word3* versus any other permutation of the query terms. Similarly, for the four-word case, we considered the original query *word1 word2 word3 word4* versus any other permutation of the four query terms. Any permutation of the query terms, except the original, is referred to as a reversed order.

Based on the sample of 1,883 three-word queries that appeared in any order in at least 10 records, we found that 95% of these preferred the *word1 word2 word3* order, 3% preferred the reversed order, and there was no order preference for the remaining 2%. Correspondingly, based on the sample of 313 four-word queries that appeared in any order in at least 10 records, we found that 84% of these preferred the *word1 word2 word3 word4* order, 14% preferred the reversed order, and there was no significant order preference for the remaining 2%. Again, in any single case, $p < 0.05$ using the sign test. The results are summarized in Table 2.

Results in this section reveal that many multiword queries, consisting of content words only, are meaningful phrases. Logically, when querying with such a phrase one might expect or prefer retrieval of the same form. Therefore, it could be beneficial to treat such queries as phrases whenever searching as a phrase generates results. The distinction, however, is valuable only if the records that contain query terms, but not the exact query phrase, are qualitatively different from the records that contain the exact query phrase.

Retrieval Evaluation Framework and Justification

We present a retrieval evaluation framework to compare classes of records using machine learning methods. Currently, most retrieval methods are evaluated based on how the retrieval

output compares to a gold standard. A gold standard is a set of records judged to be relevant to a query that provides a benchmark against which to measure the quality of search results. This approach is used at the annual Text Retrieval Conference (TREC), run by the National Institute of Standards and Technology (NIST) since 1992 (Voorhees, 2002). Every year, NIST develops a list of queries, called topics, and provides large test collections and uniform scoring procedures. For every TREC topic, the conference participants apply their retrieval methods. Then, human experts judge a select set of records based on retrieval by all participants and produce a gold standard. Retrieved records are taken to be relevant or not relevant to the query, based on their presence or absence in the gold standard. The difficulty with this approach is that a gold standard is created by human experts, which makes the evaluation both expensive and time consuming. Therefore, it would be very desirable if we could avoid manually creating a gold standard.

We suggest using an automated retrieval evaluation method that utilizes machine learning to compare different retrieval approaches without a manually created gold standard. We develop our method based on the following two hypotheses:

> H1. As a class, the records in P systematically differ from the records in $C(= S \cup R)$.
>
> H2. The systematic differences between P and C reflect characteristics of these two sets that make records in P more likely than records in C to be useful to a searcher querying with a phrase.

It is important that the reader see why these two hypotheses are highly plausible. This is primarily based on the fact that phrases tend to be much more specific than individual words. A phrase of two or more words tends to define a specific topic of discussion because the meaning of each word is constrained by the context of the other words in the phrase. On the other hand, if the words occur, but not as a phrase, the individual words may occur with varied meanings and different relationships to each other. To illustrate this point, we searched MEDLINE™ with the query *restless legs syndrome* as a phrase and as a Boolean conjunction of query terms. A typical record containing that exact search phrase is entitled

> Treatment of the **restless legs syndrome**.

As the title indicates, the above record is very likely to talk about the query subject. On the other hand, an irrelevant record entitled

> Three cases of drug-induced akathisia due to antiemetics during cancer palliative care

is retrieved by Boolean conjunction of the query terms because it contains the following text:

> This **syndrome** consists of subjective … and objective components (**restless** movement, including rocking on one's feet, walking in position shuffling and tramping the **legs**, and crossing and uncrossing one's legs while sitting).

Of course, not every record containing the search phrase is focused on the query subject, as the phrase could be mentioned in passing. Likewise, it is possible for a record to contain the query terms, but not the phrase, and still be talking about the query subject. But these cases tend to be less common. The phrase *restless legs syndrome* happens to be a MeSH™ term,

and upon examining the sets *P* and *C* in MEDLINE™, we found that 75% of documents in *P* have the MeSH™ term “restless legs syndrome” assigned to them, whereas only 47% of the documents in *C* have this MeSH™ term assigned.

Not only are the hypotheses H1 and H2 highly plausible, but we are in a position to investigate H1 using machine learning methods. If the patterns of word usage for records in *P*, Π_P, and records in *C*, Π_C, are different and the machine learning method is able to identify that difference while training, it will have distinguished between the two classes of records in the test set in a systematic way. If, on the other hand, the patterns of these two classes of records are not significantly different or the machine learner was not able to capture the difference, then we will not have succeeded in distinguishing between the two sets.

In our investigation, we used Naïve Bayes (NB) as the machine learning method as it has been shown to deal successfully with a large variety of text classification problems (Lewis, 1998). We used single words as features and the Binary Independence (Lewis) formulation of NB. Given a training set—a set of records that have already been classified as *positive* or *negative*—the learning method computes the weights of the features, based on the probability of a feature appearing in the *positive* and *negative* sets. These weights are then used to score the records from a test set according to the features (words) they contain. Based on that score, a method ranks records from *most likely to be positive* (high score) to *most likely to be negative* (low score). For convenience, we will refer to such scores as NB scores.

Once the machine learning algorithm is applied and test records are scored, we employ the Receiver Operating Characteristic (ROC) metric (Fawcett, 2006) to evaluate the ability of the classifier to distinguish between the two classes. An ROC curve is a graphical representation of true positives on the y-axis versus false positives on the x-axis as the records in the test set are examined in rank order from highest to lowest NB score. The area under the ROC curve (AUC) is called the ROC score of a classifier, and is the probability that the classifier ranks or scores a randomly chosen *positive* record higher than a randomly chosen *negative* record. The value of the ROC score typically ranges from 0.5 to 1.0, where 0.5 represents a classifier that is completely unsuccessful and 1.0 a classifier that is perfectly discriminating. In our application, it is more suggestive to report our scores as *rROC = 1-ROC*. We do this so that higher scores correspond to sets *C*, which are more like *P*. If ROC is high, discrimination is good and *C* is found to be far from the good set *P*. In that case, *C* is relatively bad and gets a low rROC score. If, on the other hand, the ROC is low, then we are having a hard time distinguishing *C* from the good set *P*, and this correlates with a good *C* and a high rROC score.

Our approach is to apply NB machine learning and to test H1. To the extent that we find H1 to be true, we can then use the results of the machine learning to rank documents by NB score that corresponds to their likelihood to be members of *P* (high score) or *C* (low score). Hypothesis H2 then applies with the claim that NB scores reflect differences in not only the probability of being a member of *P* but also the probability of being relevant. We use the term relevant merely to mean that a record is useful to a person issuing a query. Although

relevance can have various philosophical meanings (Saracevic, 2007a), (Saracevic, 2007b), it is purely pragmatic in our context.

Retrieval Evaluation Experiments and Results

To generate the data for our experiments, we used queries from PMQ2006 with no parsing indicators, which the order preference test indicated were phrases (as presented in Table 2). To ensure that there is enough data for machine learning, we required that these query phrases retrieve at least 200 records in each of the sets defined in the paragraph below. From these queries, we randomly selected 100 instances each of two-, three-, and four-word queries. We study these 300 queries and refer to them as query phrases.

For each such query phrase, we retrieved records in the following sets. Set *P*(phrase) contains records that match the query phrase precisely. Set *S*(sentence) contains records that match the query terms within a single sentence but not as a phrase. Set *R*(record) contains records that match all query terms but not within a single sentence. The sets *P*, *S*, and *R* are defined so that they do not overlap, and the union of sets $P \cup S \cup R$ represents the universe of records that contain all the terms of the query phrase. Each of these sets contain at least 200 records for every query phrase. This definition of the sets is identical to the one in the Queries with No Parsing Indicators section.

We randomly partition the sets *P*, *S*, and *R* into training and test sets

$$P = P^{Train} \cup P^{Test}$$
$$S = S^{Train} \cup S^{Test}$$
$$R = R^{Train} \cup R^{Test},$$

so that the training set is two-thirds of the whole, and test set is the remaining one-third.

We also define a set *C*(co-occurrence) as a union of sets *S* and *R* to represent the records where the query terms co-occur but not as a phrase. Clearly, set *C* contains at least 400 records. Set *C* is already partitioned into training and test parts $C = C^{Train} \cup C^{Test}$, where

$$C^{Train} = S^{Train} \cup R^{Train}$$
$$C^{Test} = S^{Test} \cup R^{Test}.$$

Now, we describe a series of experiments designed to determine whether a machine learner can distinguish between the records in the above sets.

Experiment 1

The first experiment addresses the hypothesis that the records containing the query phrase exactly, set *P*, and the records where the query terms co-occur, but not as a phrase, set *C*, are qualitatively different.

Sets P^{Train} and C^{Train} are used to train the classifier. The classifier is applied to learn the difference between the records in these two sets. We define a comparison test set $CvsP$ as a union of records in sets P^{Test} and C^{Test}

$$CvsP = C^{Test} \cup P^{Test}.$$

Based on training results, we score and rank records in the comparison test set $CvsP$ and use the rROC score to evaluate how well we distinguish records in sets P^{Test} and C^{Test}.

Results presented in Table 3 indicate that for a two-word query, a record containing all query terms, but not the phrase, is ranked higher than a record containing the exact query phrase 21% of the time. This implies that for 79% of the time, records containing the exact query phrase are ranked higher than records containing query terms but not the phrase. These results indicate that indeed there is a difference between records containing the exact query phrase and records where query terms co-occur. We applied the Wilcoxon's signed rank test (Larson, 1982) to find if that difference is significant, and we confirmed that it is with a value of $p < 10^{-16}$. Moreover, the important finding is that the machine learning approach is able to capture the difference between records in sets P and C.

What does it mean to say the machine learner learned the difference between the set P and set C? It learned a pattern for P and one for C. Because we are using single words as features, this means the words in the records in set P are different from the words in the records in set C. An rROC score of 0.21 means that the patterns are different enough and the differences are reliable enough for a document from set P to be chosen over a document from set C approximately four out of five times. This means there is a significant difference in the patterns for set P and for set C. This confirms hypothesis H1.

Experiment 2

While the exact phrase search strategy retrieves a set of records that, we assume, is an ideal response to the query, in practice, this search strategy is restrictive and, in many cases, retrieves no records. Given that limitation, we consider an intermediate class of records that contains the query terms within a single sentence. That class is less restrictive than the class of records matching the exact query phrase and, at the same time, is not as broad as the class of records where the query terms co-occur anywhere in the record. With this in mind, we evaluate whether or not records that contain the query terms within a sentence, S, are more like records that match the query search phrase, P, than records in which the query search terms simply co-occur, R.

In this experiment, we do not change our training and use sets P^{Train} and C^{Train} to train the classifier. We use the sets P^{Test}, S^{Test}, and R^{Test} to define three comparison sets $SvsP$, $RvsP$, and $RvsS$ as follows:

$$SvsP = S^{Test} \cup P^{Test}$$
$$RvsP = R^{Test} \cup P^{Test}$$
$$RvsS = R^{Test} \cup S^{Test}.$$

Test set *SvsP* is designed to compare records in sets S^{Test} and P^{Test}. Likewise, test set *RvsP* is designed to compare records in sets R^{Test} and P^{Test}, and test set *RvsS* compares records in sets R^{Test} and S^{Test}.

For each of these comparison test sets, we compute the rROC score. Table 4 presents these results and, on the final line, repeats the results from Table 3 for comparison. We observe that machine learning can distinguish between the three sets of records *P*, *S*, and *R* for two-, three-, and 4-word queries. For example, with two-word queries, a record from set S^{Test} is ranked higher than a record from set P^{Test} 23.9% of the time. However, a record from set R^{Test} is ranked higher than a record from set P^{Test} only 18.8% of the time. This indicates that records containing query terms within a single sentence are more like records that match the query phrase exactly as compared to records where the query search terms simply co-occur. The Wilcoxon's signed rank test confirms that the difference in rROC scores for the test sets *SvsP* and *RvsP* is significant.

Experiment 3

These results show that we could benefit from using records with exact phrase, P^{Train} and records where the query terms co-occur but not within a single sentence, R^{Train}, to train the NB classifier. Using these more distant sets of records to train the classifier could enhance the classifying power of the machine learner. Sets P^{Test}, S^{Test}, and R^{Test} are used to define three comparison test sets *SvsP*, *RvsP*, and *RvsS*, as before, and the rROC score is used as a measure of performance.

Table 5 shows that the results of this experiment are superior to the results in Experiment 2 as seen in Table 4. By refining the training sets, we were able to better distinguish set S^{Test} records from set R^{Test} records and demonstrate more strongly than in the previous experiment that records from S^{Test} are more like records from P^{Test} than are the records from R^{Test}. For example, for two-word phrases, a record from set S^{Test} is ranked higher than a record from set P^{Test} 25.8% of the time. However, a record from set R^{Test} is ranked higher that a record from set P^{Test} only 17.7% of the time. Comparable results are true for three and four-word queries.

Experiment 4

Now, we compare records that contain the exact query phrase to records where the query terms co-occur within windows of varying spans in a single sentence.

To this end, we further refine our tests sets using the concept of word windows. As in the previous experiment, the most distant sets, P^{Train} and R^{Train}, are used to train the classifier. We define the following comparison test sets:

$$S_N vsP = S_N^{Test} \cup P^{Test},$$

where S_N^{Test} is a subset of set S^{Test}, which contains only the records where query terms co-occur within a *N*-word span. In this experiment, we compare set S_N^{Test} records with set P^{Test} records, as we vary the window size *N*.

Results listed in Table 6 show that the rROC score for two-word queries increased monotonically as the word window span narrowed. It rose from 25.8% within the sentence to 28.2% within a nine-word span and to 32.4% within a three-word span. The increase is even greater for three- and four-word queries. For the four-word queries, the rROC score grew from 28% within a sentence to 36.7% within a five-word span.

Results of this experiment demonstrate that records with query terms occurring in close proximity to each other are more like records that contain the exact query phrase as compared to records with query terms occurring farther apart. The extent to which records where query terms co-occur differ from the records containing the exact phrase is proportional to the distance between the query terms in the sentence. Moreover, our retrieval evaluation tool enables us to quantify the differences between sets of records.

Proximity Searching

Having developed proximity searching using purely automated means it is useful to compare with other researchers' work using manually prepared gold standards.

Proximity searching can retrieve relevant records missed by exact phrase searching while avoiding the large number of records retrieved by co-occurrence searching. It is a flexible search methodology that can be tailored by changing the size of the window, which controls the range of records retrieved by the search.

There is substantial evidence in the literature that records with query terms in close proximity are more relevant than records with terms scattered throughout the record. In some retrieval systems, proximity search capability is explicitly available as a search option. In other systems, proximity is implicitly implemented in the ranking methods.

One of the earliest investigations on proximity searching in text retrieval systems was presented by Keen (1992). He described different systems that utilize the proximity searching capability. In an attempt to standardize the proximity searching terminology, eight variables describing proximity searching were identified and that could determine the retrieval outcome. They include span within which the proximity operates (sentence, paragraph, record, etc), maximal allowable distance between the words, whether stop words are counted, whether the simple occurrence of words is measured or some function of word counts and distance between them, etc. He compared the search results of the Boolean AND strategy with the search results of four proximity strategies: search within the paragraph, within the sentence, within the 10-word span, and five-word span. For the last two search strategies, sentence boundaries were ignored. As the search span narrowed, precision

increased from 26% (Boolean AND) to 29% within the paragraph, 33% within the sentence, 36% within the 10-word span, and 43% within the five-word span. Recall percentages dropped with decrease in span but did not go below 59%. These results were based on the averages of seven searches on 6,004 bibliographical records consisting of title, abstract, and two controlled vocabulary fields.

Siadaty, Shu, & Knaus (2007) suggested using sentences as text windows for searching multiterm queries. The authors proposed that the co-occurrence of the query terms in a sentence provides evidence for the existence of a relationship between the words. They developed a system, which ranks MEDLINE™ records retrieved by PubMed™. As a part of their ranking system, they compared the group of records where all query terms appeared in at least one sentence in an abstract with the group of records that did not have such a sentence but contained all query terms anywhere within the record text, including abstract title and MeSH™ terms. Based on examining two queries, they showed that, on average, the precision dropped from about 0.6 for sentence-level co-occurrence to about 0.2 for record-level co-occurrence. With only two queries, this is suggestive but hardly a definitive result.

Butcher, Clarke, & Lushman (2006) integrated term proximity into the existing vector space retrieval method Okapi BM25. Given a query, they computed the relevance score of the record as a sum of the standard TFxIDF score and a proximity measure score, which is proportional to the inverse of the squared distance between the query terms. The authors found that the use of term proximity was more important for larger text collections. For example, precision of the top 10 records with proximity-enhanced BM25 was 0.6 (compared to 0.529 with the regular BM25), and 0.561 of the top 20 records (compared to 0.494 with the regular BM25) for their large collection, GOV2, consisting of 25.2 million records. On their smaller collection, TREC45-CR, consisting of 528,155 records, they did not notice an improvement. Using various size subsets of the GOV2 database, the authors verified that the relative gain achieved by proximity-enhanced BM25, compared to the original BM25, increased as the underlying text collection grew. They did not have enough evidence, however, to support the hypothesis that term proximity is important when the search engine is dealing with longer records.

Beigbeder and Mercier (2005) developed a record-ranking model based on the degree of proximity of query term occurrences in a record. In their model, every query term has an influence region, defined by an influence function of some parameter k, which controls the extent of the term's influence. The function has a maximum value of 1 in each place where a query term appears in the record. It decreases with a constant slope down to zero at a distance of k words on each side of the term. Given two query terms, the proximity value is positive in a region between the terms where both influence functions are positive. If the terms appear too far from each other, then the proximity value is zero. Setting the parameter k to a low value requires proximity at the phrase level, while high values allow the required proximity to approximate sentence or paragraph levels. The best performance was obtained for parameter values of $k \geq 20$. However, for values of $k > 3$, their model was consistently better than a vector retrieval model.

In their work, Clarke, Cormack, & Burkowski (1995) introduced a shortest substring search model, which searched for the smallest spans in the text record that contained query words. A score was assigned to each of these spans based on the inverse of the span length. The record score was computed as the sum of the scores of the selected spans. This model participated at the TREC-4 ad-hoc retrieval task. For over 65% of the topics, the average precision with this model was above the median average precision for all other models.

Haas and Losee (1994) looked at the composition of text windows. They pointed out that various syntactic and semantic relationships may be found within a limited span of words, and that it is possible to exploit them all by using windows, rather then separately interpreting each relationship and combining the results. They confirmed the previously suggested idea that a window of size 11 words provides the most natural grouping of words. Hence, their research supports the claim that if the query is a phrase, then a useful answer to that query may be found within a limited text window.

Discussion

We found that a significant fraction of queries in PubMed™ are multiterm queries without parsing instructions. We also found evidence that a large fraction of these queries are frequently used phrases in MEDLINE™ records. This raised the question of how they should best be interpreted as queries. It also suggests that users may expect answers focused on the specific subject of the phrase. Using our machine learning methodology we have been able to adduce evidence that documents containing the exact query phrase as a class are qualitatively different from documents merely containing the query words and are more likely to be relevant as answers to the query.

The findings in the area of proximity searching show that records with the query terms appearing in close proximity to each other are more relevant to the query than records with query terms appearing further apart. Specifically, they show that the quality of retrieval changes depending on proximity of query terms to each other within the records. All these outcomes of proximity theory are consistent with our results. The advantage of our method is that we have obtained our results using solely automated techniques.

We find that records with the exact phrase, *P*, are different from records that merely contain the words in the query, *R*. rROC scores distinguishing these sets are less than 0.18 for all phrase lengths considered. Clearly, these sets are significantly different and the machine learner was able to learn corresponding patterns that were used to recognize the differences in the test set. This provides strong proof for hypothesis H1. Although we do not have a proof of hypothesis H2, there are at least three lines of argument that provide support for it. First, we note that it provides an explanation for the systematic differences seen in our machine learning. If records that contained only the words, but not the phrase, were just as relevant to the subject of the phrase as the records containing the phrase, it is hard to see why or how they could have markedly different distributions of words in them. Second, empirical evidence, such as the example of the phrase *restless legs syndrome*, shows that when authors use a phrase, indexers are more likely to judge such a document as relevant to the subject of the phrase. (We have confirmed this phenomenon for several examples but

have not made a systematic study.) Third, the findings in studies of proximity searching also suggest that a text containing a phrase is more likely to be judged relevant to that phrase as a query than is text that contains the same words at a distance from each other.

Based on our results, we can establish the best retrieval order for the records. Hence, our system can be used as a ranking method in search engines. A simple approach would be to rank the records retrieved based on proximity of query terms to each other. A more sophisticated approach would be to recognize that relevance is not limited to proximity and rank the retrieved records in accordance with the machine learning NB score assigned by our method. This approach could rank some informative records from sets S and R higher than set P records that simply contain the query phrase in passing.

Our approach could also be used as assistance in creating a gold standard. While automated methods cannot completely replace human review, they could be used to boost the process of generating the gold standard. Once a sufficient number of records have been identified as relevant by human experts, one may extract new records that look like these relevant records. This approach is similar to relevance feedback, as described in Ruthven (2003). It provides a direction of search for the possibly relevant records by providing a set of neighboring records to the existing relevant ones.

A more straightforward design of our machine learning-based method might have been to train using the records that contain query terms against the rest of MEDLINE™, defined as

$$M' = M \setminus (P \cup S \cup R),$$

where M stands for all of MEDLINE™. Or, one could also train using records containing an exact query phrase, set P, against M'. In fact, we started with such an approach. Although the method could learn the difference between sets $P \cup S \cap R$ and M', the result was not effective in distinguishing the sets P, S, and R from each other. We believe there are two factors that are relevant to this issue. First, such training does not directly train the machine learner to distinguish P, S, and R. Second, NB may perform poorly when trained on unbalanced training sets (Sohn, Kim, Comeau, & Wilbur 2008), and clearly the sets M' and P are unbalanced, in terms of the number of records they contain. To deal with these problems and improve our results, we performed the training as described in the paper.

Conclusions

A significant fraction of queries in PubMed™ are frequently used phrases in MEDLINE™ records. We find evidence of benefit in interpreting such queries as phrases for retrieval purposes. Using our automatic retrieval evaluation framework based on machine learning methods, we demonstrate that records with query terms occurring in close proximity to each other are more like records that contain the exact query phrase as compared to records with query terms occurring farther apart. The extent to which records where query terms only co-occur differ from the records containing the exact phrase is proportional to the distance between the query terms in the sentence. The results generated by our evaluation method are

consistent with the results of researchers studying proximity searching based on manually judging the retrieved records. The appeal of our evaluation method is that it does not require a manually constructed gold standard.

Acknowledgments

The authors are supported by the Intramural Research Program of the National Institutes of Health—National Library of Medicine™.

References

Beigbeder, M., Mercier, A. An information retrieval model using the fuzzy proximity degree of term occurences. Proceedings of the 2005 ACM Symposium on Applied Computing (SAC); New York: ACM; 2005. p. 1018-1022.

Buttcher, S., Clarke, C., et al. Term proximity scoring for ad-hoc retrieval on very large text collections. Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; New York: ACM; 2006. p. 621-622.

Clarke, C., Cormack, G., Burkowski, F. Shortest substring ranking. In: Harman, K., editor. Proceedings of Fourth Text Retrieval Conference (TREC-4); Darby, PA: Diane Publishing; 1995. p. 295-304.

Fawcett T. An introduction to ROC analysis. Pattern Recognition Letters. 2006; 27:861–874.

Haas S, Losee R. Looking in the text windows: Their size and composition. Information Processing and Management. 1994; 30(5):619–629.

Keen EM. Some aspects of proximity searching in text retrieval systems. Journal of Information Science. 1992; 18(2):89–98.

Larson, HJ. Introduction to probability theory and statistical inference. New York: John Wiley & Sons; 1982.

Lewis, DD. Naive (Bayes) at forty: The independence assumption in information retrieval. Proceedings of the 10th European Conference on Machine Learning; 1998. Retrieved October 21, 2008, from https://eprints.kfupm.edu.sa/52684/1/52684.pdf

Ruthven I, Lalmas M. A survey on the use of relevance feedback for information access systems. The Knowledge Engineering Review. 2003; 18(2):95–145.

Saidaty M, Shu J, Knaus W. Relemed: sentence-level search engine with relevance score for the MEDMLINE database of biomedical abstracts. BMC Medical Informatics and Decision Making. 2007; 7(1)

Saracevic T. Relevance: A review of the literature and a framework for thinking on the notion in information science. Part II: Nature and manifestations of relevance. Journal of the American Society for Information Science. 2007a; 58(13):1915–1933.

Saracevic T. Relevance: A review of the literature and a framework for thinking on the notion in information science. Part III: Behavior and effects of relevance. Journal of the American Society for Information Science. 2007b; 58(13):2126–2144.

Sohn S, Kim W, Comeau DC, Wilbur WJ. Optimal training sets for Bayesian prediction of MeSH assignment. Journal of the American Medical Informatics Association. 2008; 15(4):546–553. [PubMed: 18436913]

Voorhees, E. Lecture Notes in Computer Science. Vol. 2406. New York: ACM; 2001. The philosophy of information retrieval evaluation; p. 355-370.

Appendix

Retrieval in PubMed™

The PubMed™ search engine uses the co-occurrence search strategy, with a considerable amount of query expansion called automatic term mapping. The automatic term mapping feature uses additional resources to translate and expand the query.

The most significant additional resource used is the MeSH™ database, a controlled vocabulary of biomedical terms used for indexing journal articles in MEDLINE™, a collection of biomedical abstracts at NLM. For example, a PubMed™ search on the term *cancer* is automatically expanded to include the MeSH™ term for cancer, *neoplasm*. For a multiword search phrase, PubMed™ attempts to recognize the whole search phrase as a concept in MeSH™, as in the phrase *ear infection*. The search is then expanded to include the medical term for ear infection, *otitis*.

If a multiword search phrase is not recognized as a MeSH™ concept, PubMed™ breaks apart the phrase and repeats the automatic term mapping process until a match is found. For example, the phrase *common cold vitamin C* is expanded to the following search:

common cold AND (*vitamin C* OR *ascorbic acid*)

If no subset of contiguous words is recognized as a concept in the MeSH™ database, then a regular co-occurrence search takes place in which the search is translated to a search of query words combined by the Boolean operator AND.

PubMed™ handles a query enclosed in double quotes as a phrase. In that case, automatic term mapping is bypassed, and the phrase is searched in another database, the list of indexed phrases. If the query phrase is found in the list of indexed phrases, then PubMed™ retrieves the set of records that contain the query phrase; otherwise PubMed™ treats the search phrase as if it was entered without quotes. Handling a query as a phrase is very important; however, it is currently limited to the list of indexed phrases in PubMed™.

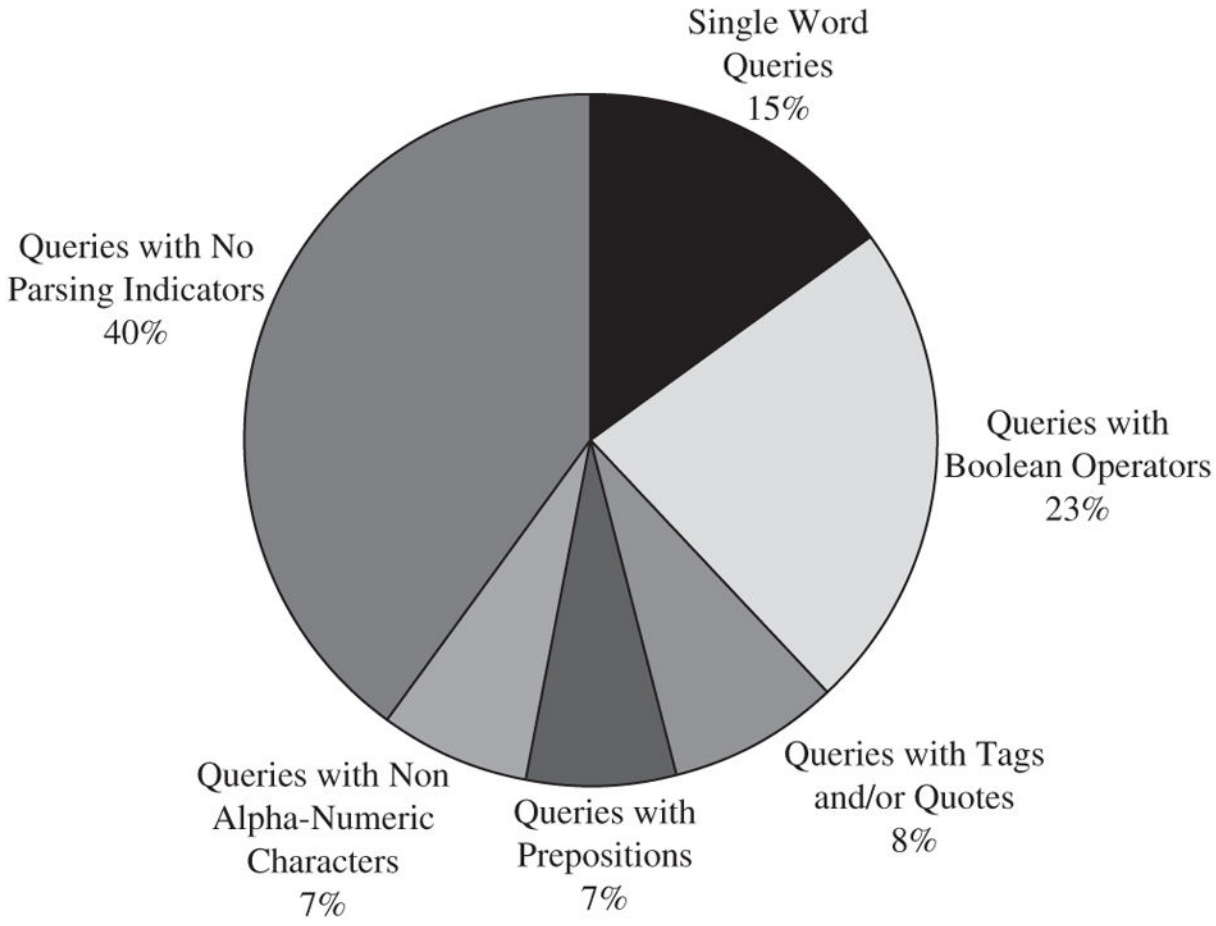
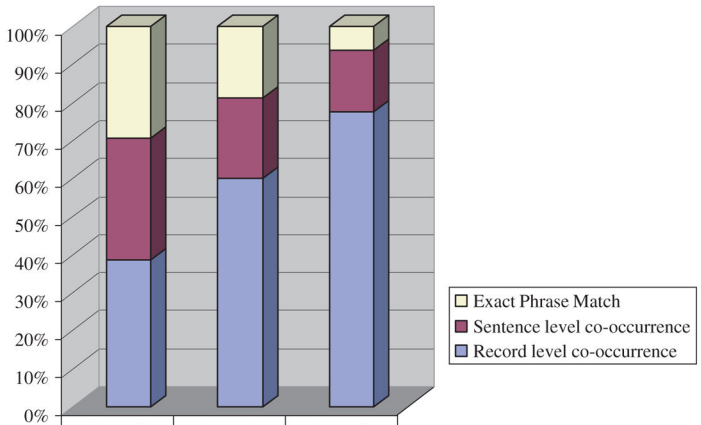


FIG. 1.

Distribution of queries in the PMQ2006 dataset. Multiword queries with parsing indicators (including Boolean Operators 'AND,' 'OR,' 'NOT,' PubMed tags, quotation marks, commas, and stop words) represent 45% of all queries; 40% are multiword queries with no parsing indicators and 15% of the queries are single word queries.



	2-word queries	3-word queries	4-word queries
Exact Phrase Match	0.293	0.188	0.062
Sentence level co-occurrence	0.321	0.212	0.162
Record level co-occurrence	0.386	0.600	0.776

FIG. 2.

Distribution of records between the non-overlapping sets P (exact phrase match), S (sentence level co-occurrence), and R (record level co-occurrence) for the categories of two-, three-, and four-word phrases. The results are averaged over 10,000 unique test queries in each category.

TABLE 1

Percentage of queries by record types.

Record Type	Two-word queries	Three-word queries	Four-word queries
$P \neq \emptyset$	54%	36%	15%
$P = \emptyset$ & $S \neq \emptyset$	23%	34%	39%
$P = \emptyset$ & $S = \emptyset$ & $R \neq \emptyset$	3%	8%	18%
$P = \emptyset$ & $S = \emptyset$ & $R = \emptyset$	20%	22%	28%

Note. Percentage of queries that were found as an exact phrase (row 1); percentage of queries that were not found as an exact phrase, but whose terms were found within a single sentence in the records (row 2); percentage of queries whose terms did not co-occur within a single sentence in any record, but did co-occur within whole records (row 3). Last row of the table gives the percentage of queries that did not succeed. The results are based on 10,000 unique queries each, for the case of two, three and four-word queries. Our failure rate is higher than it would be in PubMed, because we do not perform query expansion.

TABLE 2

Significance of the word order in the two-, three-, and four-word queries.

Query size	Original order preferred	Reversed order preferred	No significant order preference
Two-word queries	90%	7%	3%
Three-word queries	95%	3%	2%
Four-word queries	84%	14%	2%

TABLE 3

rROC scores for set *CvsP* based on training *P* vs *C*.

	Two-word queries	Three-word queries	Four-word queries
CvsP	0.210	0.200	0.212

Note. rROC scores for set *CvsP* averaged over 100 search phrases for two-, three-, and four-word queries with no parsing indicators. Naïve Bayes machine learning method uses sets P^{Train} and C^{Train} to train the classifier.

TABLE 4

rROC scores for sets *SvsP*, *RvsP*, and *RvsS* based on training *P* vs *C*.

	Two-word queries	Three-word queries	Four-word queries
SvsP	0.239	0.247	0.255
RvsP	0.188	0.177	0.191
RvsS	0.429	0.398	0.413
CvsP	0.210	0.200	0.212

Note. rROC scores for three test sets *SvsP*, *RvsP*, and *RvsS* averaged over 100 search phrases for two-, three-, and four-word queries with no parsing indicators. Naïve Bayes machine learning method uses set P^{Train} and C^{Train} to train the classifier. The last row of the table repeats the results from Table 3 for comparison.

TABLE 5

rROC scores for sets *SvsP*, *RvsP*, and *RvsS* based on training *P* vs *R*.

	Two-word queries	Three-word queries	Four-word queries
SvsP	0.258	0.266	0.280
RvsP	0.177	0.166	0.174
RvsS	0.392	0.365	0.362

Note. rROC score for three test sets *SvsP*, *RvsP*, and *RvsS* averaged over 100 search phrases for two-, three-, and four-word queries with no parsing indicators. Naïve Bayes machine learning method uses set P^{Train} and R^{Train} to train the classifier.

TABLE 6

rROC scores comparing set P with the subsets of set S of varying window spans.

Window size	Two-word queries	Three-word queries	Four-word queries
$S_3 vsP$	0.324		
$S_4 vsP$	0.317	0.345	
$S_5 vsP$	0.306	0.340	0.367
$S_6 vsP$	0.299	0.331	0.346
$S_7 vsP$	0.292	0.323	0.334
$S_8 vsP$	0.286	0.315	0.326
$S_9 vsP$	0.282	0.308	0.322
$S_{10} vsP$		0.303	0.317
$S_{11} vsP$			0.313
$SvsP$	0.258	0.266	0.280

Note. rROC scores for two-, three-, and four-word queries averaged over 100 queries with no parsing indicators. Naïve Bayes machine learning method uses set P^{Train} and R^{Train} to train the classifier. Comparison test sets are designed to compare the records of set P^{Test} to the subsets of set S^{Test} defined by varying the window spans. The last line repeats results from Table 5 for comparison.