SingleCite: Towards an improved Single Citation Search in PubMed

Lana Yeganova, Donald C Comeau, Won Kim,
W John Wilbur, Zhiyong Lu
National Center for Biotechnology Information, NLM, NIH, Bethesda, MD, USA
{yeganova, comeau, wonkim, wilbur, luzh}@mail.nih.gov

Abstract

A search that is targeted at finding a specific document in databases is called a Single Citation search. Single citation searches are particularly important for scholarly databases, such as PubMed®, because users are frequently searching for a specific publication. In this work we describe SingleCite, a single citation matching system designed to facilitate user's search for a specific document. We report on the progress that has been achieved towards building that functionality.

1 Introduction

PubMed, a search engine that works on MEDLINE®, processes on average 3 million queries a day and is recognized as a primary tool for scholars in the biomedical field (Falagas, Pitsouni, Malietzis, & Pappas, 2008; Lu, 2011; Wildgaard & Lund, 2016). Given the significance of PubMed, improving query understanding offers tremendous opportunities for providing better search results. In this work we present SingleCite, a single citational matching tool designed with the goal to improve current single citation searching functionality in PubMed.

PubMed queries are generally being classified as informational or navigational. Informational queries, also known as topical searches, such as *colon cancer*, or *familial Mediterranean fever*, are intended to satisfy information needs on a search topic. They tend to retrieve many documents, the information need is typically not satisfied with just one result, and the user does not know in advance which document will be the most useful. Navigational queries, also called known-item queries (Ogilvie & Callan, 2003), such as *Katanaev AND Cell 2005, 120(1):111-22*, are intended to retrieve a specific publication. Processing navigational queries requires techniques rather different from those used for information searches, and includes access to structured citation data, syntactic parsers, and intelligent metadata (volume, issue, page, date fields) parsers. Parsing and managing citations is a critical task of digital libraries and has been studied extensively (Anzaroot & McCallum, 2013; Kim, Le, & Thoma, 2008; Zhang, Cao, & Yu, 2011). Addressing navigational queries is particularly important for scholarly citation databases, including PubMed, where navigational searches constitute about half of all queries (Islamaj, Murray, Névéol, & Lu, 2009; Yeganova, Kim, Comeau, Wilbur, & Lu, 2018), unlike general search domain where they represent a significantly smaller portion (Jansen, Booth, & Spink, 2007). Moreover, because of specificity of the expected response, retrieving the correct document is of great importance.

Users that have a specific document in mind, frequently enter a query they believe uniquely identifies that document. A specific document may be accessed in various ways. Author name(s) queries and title queries are two most frequent navigational search patterns (Yeganova et al., 2018). Other search patterns include combinations of author(s), year, key words, journal, volume, issue, page and date fields. Not all navigational queries lead to retrieving a single citation. Author name queries may retrieve several PMIDs written by the same person. Similarly, title queries targeted at retrieving a document with a particular title, may be interpreted as key words and retrieve multiple matching documents. Our single citation matching tool is not intended to handle such queries. It is designed for queries that provide enough information to establish a high probability match between a query and a single correct document. When such a document is found, PubMed redirects a user directly to that document, instead of a summary page which generally contains many retrieved results.

Proceedings of the BioNLP 2018 workshop, pages 151–155
Melbourne, Australia, July 19, 2018. ©2018 Association for Computational Linguistics

Here we present SingleCite, a single citation matching algorithm designed to retrieve a high probability match for a navigational query targeting a unique document. The algorithm establishes a query-document mapping by building a regression function to predict the probability of a retrieved document being the target based on three variables: the score of the highest scoring retrieved document, the difference in score between the two top retrieved documents, and the fraction of a query matched by the candidate citation. We demonstrate the advantage of our method by comparing it with the currently existing single citation matching scheme in PubMed and manually annotating a random sample of 1,000 queries on which the two methods disagreed. We also apply SingleCite on 1 million zero-hit PubMed queries and recover a single citation match for 3.3% of them.

2 Methods

To create the mapping between a query and a candidate PubMed document we propose an algorithm that predicts the probability of a retrieved document being the target given a query. We propose three variables to measure the success of match between a query and a PubMed record: the log odds score of the top scoring pmid, the difference between log odds scores of the two top scoring pmids, and the fraction of alpha-numeric query characters that match the record. In the next subsection we address the details of how we compute the log odds score between a query and a PubMed record. Then we describe how we build the regression function that takes as input the three variables and predicts the probability of a retrieved document being the target. In this work we also propose techniques to create artificial queries, where each query is created from a known document. This query set is essential for training the regression functions in the absence of manually annotated data.

2.1 Computing the query-document score

We represent PubMed documents by their bibliographic data including article title, author name(s), journal title, volume, issue, page, and date as features. Features from abstracts are not used as they are generally not as specific and less likely to be the source of a user's query terminology for a single citation. The seven fields of interest will be referred to as citation fields. We index the elements of citation fields by including all non-stop word single tokens and capitalized stop words, that are then lower cased. We also index all token pairs with the following exceptions: do not include first name or initials alone, do not include the last page of a page range alone, do not include the issue, except as paired with the volume.

The features are then weighted with the IDF weights approximating naïve Bayesian weights, and the resultant weighted features are added up for each element of a document matching the query. Using these IDF weights we compute the log odds score that the matching document is what the user was seeking.

To produce log odds scoring that is as close to the truth as possible we make some modifications to the weighting. The first problem is that IDF weighting is used for both word pairs and single words. To correct for this dependency, we modify the IDF weights of pairs as follows:

$$modIDF_{(w1,w2)} = IDF_{(w1,w2)} - IDF_{w1}$$

We also adjust the IDF weights to correct for the unevenness in the amount of dependency within fields in the bibliographic record. The unevenness is caused by terms in some fields being more independent then in others. For example, the terms in the author, page, volume, issue and date fields tend to be independent of each other. On the other hand, in fields such as article titles and journal titles terms are more dependent. Intuitively, it is significantly more difficult to predict author first name given the last name, or to predict issue given the volume, then to predict a word following another word in a title.

Query: "Strategies for assessing and fostering hope Penrod.J.& Morse.J.M"

Clicked Article: "Penrod, J., & Morse, J. M. (1997). Strategies for assessing and fostering hope: The Hope Assessment."

Derived Query Parse: Strategies for assessing and fostering hope [Title] Penrod.J. [Author] & Morse.J.M [Author].

Figure 1: Query annotation based on clicked article.

We use a machine annotated training set to optimize the weight modification. The machine annotated queries are created from NCBI PubMed logs by sampling navigational queries that are followed by a user clicked document. Given a query and a clicked pmid, we interpret the parts of the query by mapping them to citation fields of the clicked document (title, author, journal, volume, issue, page and date). This approach allows us to obtain an unlimited amount of citation query–pmid pairs. Figure 1 presents an example of such annotation. Using the machine annotated queries as a training set we now modify the IDF weights to improve the matching between the query elements and PubMed citation. To correct for the dependencies within the title fields, we upweight the IDF weights for terms coming from all the remaining fields by the factor of 1.4. The factor of 1.4 is empirically determined using a grid search.

Given a query, we can now score all PubMed records and retrieve top ten ranks. As users frequently submit queries with misspelled words (Behnert & Lewandowski, 2017) we have incorporated spell checking limited to a single edit correction per term into our processing. This is implemented by retrieving the top ten scoring records based on the original query and then applying spelling correction to the query one term at a time. This may increase the match score between the query and a record. If we have increased the difference between the top score and the next best score, the revised query is accepted as the preferred result. Otherwise the original scores are retained.

Now that we can compute the scores between a query and candidate pmids, the next step is how to interpret and combine the scores. To address that question we build regression functions to map the log odds scores and the fraction of the query matched, to the probability the top scoring document being the target. Since the training of a regression function requires labeled query-pmid pairs, we propose methods for producing artificial queries.

2.2 Artificial Queries

We propose techniques for creating an artificial dataset of annotated citation queries modelled upon user's actual queries. Simulating test collections for evaluating retrieval quality has been explored in the literature (Azzopardi & de Rijke, 2006; Azzopardi, de Rijke, & Balog, 2007) as it offers a viable alternative to manually annotating queries. Constructing simulated known-item queries present a particularly well-defined task; the retrieval goal is the document from which a query is constructed.

We have already shown how to get an unlimited supply of query-document pairs. From each such pair, we can take the annotated query as a model describing the fields from which the query is composed and the length of each such piece. We then randomly sample a PubMed document. Using the pattern of the annotated query, we generate a synthetic query from the reference PubMed document mimicking the structure of the annotated query. For example, if the annotated query contains an author name, we extract an author name from the document that is closest in length to the author name element in the model query. The same technique holds for all the fields found in the model query. A second technique randomly selects a PubMed document, creates its citation as a text string, and then randomly splits it into two strings. Each of these strings then simulates a cut-and-paste query.

The advantage of such queries is that we know the target document the query is intended to retrieve. However, we have no guarantee the query will retrieve the document on which it is based. Using these two techniques, we created a set of one million queries.

2.3 Training the Regression Functions using Artificial Queries

Based on the synthetic queries which have known target documents in PubMed, the goal is to build a regression model for estimating the relationship between the three dependent variables and the predictor. Predictor in this model is label of a query document pair, 1 if the document is identified correctly, and 0 otherwise. For each query we carry out retrieval using our system and record the top scoring documents from PubMed; x and y represent values determined by the retrieval as

x = score1; y = (score1-score2)/score1.

To be kept, a score had to be greater than a certain lower bound and we only record at most scores for the top three documents. The first stage of our computation is to estimate the probability $p(t \in PubMed|x,y)$, where t represents the target document of the query. We construct the first regression function which estimates that probability given $x\ and\ y$. The second stage of the computation is to estimate $p(d_1 = t|x,y,t \in PubMed)$, the probability that the document at rank 1 is the target document. Again, we use all the artificial queries and their retrieved documents as long as at least two scores were above the threshold to directly estimate $p(d_1 = t|x, y, t \in PM)$. This is obtained by a straightforward application of the two-dimensional isotonic regression algorithm (Spouge, Wan, & Wilbur, 2003). Consequently, we can combine this probability with the previously estimated $p(t \in PubMed|x, y)$ and obtain:

$$p(d_1 = t|x, y) = \\ p(d_1 = t|x, y, t \in PM)p(t \in PM|x, y).$$

Given retrieval results from our system for a query, $p(d_1 = t|x, y)$ provides an estimate of how likely the user was looking for document d_1.

The final step in the model constructs a third regression by taking as input $p(d_1 = t|x, y)$ and the query fraction matched. We hypothesize that if the query is a sufficiently good match to a PubMed record and there is a reasonable gap to the next best score, the top scoring record may be of interest even if not exactly what the user was seeking. We conjecture this to depend on the quality of match and how much of the query is involved in the match. The difficulty however is that we do not have a way to simulate this problem with known answers. Instead, we compare our system output to the output of a legacy system (known to have high precision) possessed and currently used by NCBI to processes single citation queries. A total of 343,731 unique queries were collected from PubMed logs on October 12, 2016. These were the queries that triggered the single citation matching system in PubMed. The existing system produced a presumed high-quality answer for 58,375 queries. SingleCite produced probabilistic output of variable quality for 232,256 of these queries. For the 51,472 queries where the existing and the new system both made predictions, we counted predictions as correct when the two systems agreed on the retrieved pmid (45,713) and incorrect otherwise (5,759). Using this data, we build the regression function that combines the probability of top scoring document being the target obtained from previous step and the fraction of the query matched for the 51,472 queries. We empirically chose a threshold of 0.98 and accept predictions from the third regression function that are above or at that value.

3 Evaluation

We ran SingleCite on the 343,731 query set mentioned above, and predicted high probability answers on 26,892 queries (with the 0.98 threshold) where the legacy system made no predictions. To evaluate the accuracy of our algorithm, we randomly sampled 500 queries from the set where we alone made predictions and examined the quality of the answers. We found 7 answers clearly wrong and 5 probably wrong but potentially useful. Wrong answers were mostly seen with the shorter queries. These results are consistent with a 98% accuracy level. We further randomly sampled 200 queries from the set of 11,688 queries where the legacy system alone made the prediction. There we found 22% of answers clearly wrong and 8% probably wrong, but potentially useful. The remaining 70% of queries produced a single citation match that we thought was correct. On close examination of queries missed by SingleCite, we identified a few opportunities for improvement, including enriching the index with journal name abbreviations (currently index contains only full journal names), and better handling of hyphenated last names (for example, query containing *Shiloh* did not retrieve the target document containing *Shiloh-Malawsky* as an author).

As a second experiment, we ran SingleCite on one million queries randomly sampled from queries submitted to PubMed in 2017 that produced no results using the legacy system. We found a single citation match for 3.34% of them.

4 Conclusion

Here we present our preliminary work on the single citation matching tool aimed to facilitate user's search for a specific document in PubMed. The method depends on good feature engineering combined with novel approaches for adjusting feature weights when combining elements from different fields. We also describe how we create one million synthetic queries, each along with the PMID of the document used as the source. SingleCite shows promising results compared to the existing system for finding single citations. The tool can also be used as part of NLP pipeline for identifying citations in text, abstract or full text, and mapping them to corresponding PMIDs. The tool can further be useful for citation management systems and portfolio analysis.

Acknowledgments

This research was supported by the Intramural Research Program of the NIH, National Library of Medicine.

References

Anzaroot, S., & McCallum, A. (2013). *A New Dataset for Fine-Grained Citation Field Extraction*. Paper presented at the Proceedings of the 30 th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

Azzopardi, L., & de Rijke, M. (2006). *Automatic construction of known-item finding test beds*. Paper presented at the SIGIR '06.

Azzopardi, L., de Rijke, M., & Balog, K. (2007). *Building Simulated Queries for Known-Item Topics*. Paper presented at the SIGIR'07, Amsterdam, The Netherlands.

Behnert, C., & Lewandowski, D. (2017). Known-item searches resulting in zero hits: Considerations for discovery systems. *The Journal of Academic Librarianship, 43*(2), 128-134.

Falagas, M., Pitsouni, E., Malietzis, G., & Pappas, G. (2008). Comparison of PubMed, Scopus, Web of Science, and Google Scholar: strengths and weaknesses. *The FASEB Journal, 22*(2), 338-342.

Islamaj, R., Murray, C., Névéol, A., & Lu, Z. (2009). Understanding PubMed user search behavior through log analysis. *Database*.

Jansen, B. J., Booth, D. L., & Spink, A. (2007). *Determining the User Intent of Web Search Engine Queries*. Paper presented at the WWW 2007, Banff, Alberta, Canada.

Kim, J., Le, D., & Thoma, G. (2008). *Naive Bayes Classifier for Extracting Bibliographic Information From Biomedical Online Articles*. Paper presented at the Proc 2008 International Conference on Data Mining. Las Vegas, Nevada, USA. July 2008;II:373-8.

Lu, Z. (2011). PubMed and beyond: a survey of web tools for searching biomedical literature. Database: the journal of biological databases and curation. *Database (Oxford), 2011.*

Ogilvie, P., & Callan, J. (2003). *Combining Document Representations for Known-Item Search.* Paper presented at the SIGIR, Toronto, Canada.

Spouge, J., Wan, H., & Wilbur, W. J. (2003). Least Squares Isotonic Regression in Two Dimensions. *Journal of Optimization Theory and Applications, 117*(3), 585-605.

Wildgaard, L. E., & Lund, H. (2016). Advancing PubMed? A comparison of 3rd-party PubMed/MEDLINE tools. *Library Hi Tech, 34*(4), 669-684. doi: https://doi.org/10.1108/LHT-06-2016-0066

Yeganova, L., Kim, W., Comeau, D. C., Wilbur, W. J., & Lu, Z. (2018). A Field Sensor: Computing the composition and intent of PubMed queries. *DATABASE.*

Zhang, Q., Cao, Y.-G., & Yu, H. (2011). Parsing Citations in Biomedical Articles Using Conditional Random Fields. *Comput Biol Med, 41*(4), 190-194.