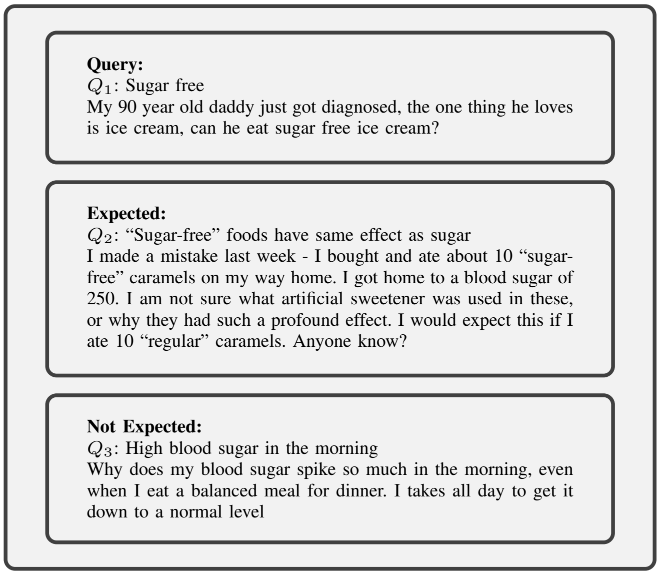
# Community Question Retrieval in Health Forums

Hamman Samuel[1], Mi-Young Kim[2], Sankalp Prabhakar[3], Mohomed Shazan Mohomed Jabbar[4], Osmar Zaïane[5]

*Abstract*— **Community Question Answering (CQA) has emerged as a popular type of service enabling users to ask and answer questions, and access the existing knowledge-base. CQA archives contain a lot of useful user-generated content and have been recognized as important information resources for the web. To improve accessibility to this body of knowledge in CQA archives, effective and efficient question retrieval is required. Question retrieval in a CQA archive aims to identify and retrieve existing questions that are relevant to new user questions. The objective of this study is to develop a question retrieval system that can sift through such forums and identify existing questions which are most similar to the user-provided question. We focus on health forums, and propose a CQA system using weighted TF-IDF, relevance heuristics, and term expansion. We compare our proposed algorithm against other well-known methods, and demonstrate that our method outperforms the Latent Dirichlet allocation (LDA) topic model, Latent Semantic Indexing (LSI), language model-based information retrieval, BM25, vector space, Word2Vec, and semantic similarity approaches. Our initial experiments use datasets from the IEEE Healthcare Data Analytics Challenge 2015, and we also present our efforts towards development of a Bronze Standard for question similarity evaluation using self-annotations and annotations provided by affiliates of Mayo Clinic.**

## I. Introduction

Online services have been used to build large repositories of questions and answers, moving from traditional Frequently Asked Questions (FAQ) services to community-based Question and Answer (Q&A) services such as Quora and Stack Exchange. To fully utilize these repositories, users need to be able to search existing answers by identifying similar questions that may have already been asked. This functionality typically implemented by first retrieving questions expected to have the same answers as a new question, and then returning the related answers [1]. For example, given a query $Q_1$ in Figure 1, question $Q_2$ can be returned as one of the similar questions to $Q_1$. In contrast, $Q_3$ is not as similar, and therefore, $Q_2$'s answer can then be used to address $Q_1$. This is what we term as "question retrieval", where returned questions are semantically equivalent or at least semantically nearer to a new incoming question.



**Query:**
$Q_1$: Sugar free
My 90 year old daddy just got diagnosed, the one thing he loves is ice cream, can he eat sugar free ice cream?

**Expected:**
$Q_2$: "Sugar-free" foods have same effect as sugar
I made a mistake last week - I bought and ate about 10 "sugar-free" caramels on my way home. I got home to a blood sugar of 250. I am not sure what artificial sweetener was used in these, or why they had such a profound effect. I would expect this if I ate 10 "regular" caramels. Anyone know?

**Not Expected:**
$Q_3$: High blood sugar in the morning
Why does my blood sugar spike so much in the morning, even when I eat a balanced meal for dinner. I takes all day to get it down to a normal level

Fig. 1. Question Retrieval Examples

In recent years, Community-based Question Answering (CQA) has been studied extensively. Large numbers of questions and answers archived in health forums provide a valuable knowledge base to patients and caregivers. But a major challenge is that the users often ask the same or similar questions which are already existing in the knowledge-base, either because of not being able to spend time to search for similar questions, or because they lack domain knowledge. This creates a significant increase in repetitive questions. In this research work, we develop a system using a weighted TF-IDF methodology combined with relevance heuristics and term expansion approaches. Our method significantly outperforms several other existing popular techniques in the field of informal retrieval. In Section II, we explain the related work around information retrieval and CQA. In Section III, we describe our proposed method, and in Section IV, we present experiments comparing our method with others. Lastly, we conclude in Section V with potential future directions for research in question retrieval.

[1] Hamman Samuel is a PhD Candidate with the Department of Computing Science, University of Alberta, Canada `hwsamuel@ualberta.ca`

[2] Mi-Young Kim is a Researcher with the Alberta Machine Intelligence Institute, Canada `miyoung2@ualberta.ca`

[3] Sankalp Prabhakar was a Master's student with the Department of Computing Science, University of Alberta, Canada while conducting this research work `sankalp@ualberta.ca`

[4] Mohomed Shazan Mohomed Jabbar was a Master's student with the Department of Computing Science, University of Alberta, Canada while conducting this research work `mohomedj@ualberta.ca`

[5] Osmar R. Zaïane is a McCalla and Killam Professor at the Department of Computing Science, University of Alberta, Canada `zaiane@ualberta.ca`

## II. Related Work

The recent Healthcare Data Analytics Challenge (HDAC) 2015, hosted at the International Conference of Health Informatics (ICHI) 2015, looked at the research challenge of question retrieval. The approach of the challenge winners was based on TF-IDF to get important words as features, and then using Latent Semantic Analysis (LSA) for extracting other features [2]. The authors also used cosine similarity between words using vectors. However, there was no exploration on usefulness of synonyms or medical abbreviations.

In other related research work, [3] have worked on retrieval of semantically similar questions from frequently asked questions (FAQ) using a WordNet dictionary and a marker-passing algorithm. It is worth noting that most of the recent retrieval models are based on language models [4]. Translation-based language models were also proposed by using translation probabilities, including word-to-word [5] and phrase-to-phrase probabilities [6]. These probabilities are learned from question-question pairs [5], question-description pairs [7], and question-answer pairs [8].

Term Frequency and Inverse Document Frequency (TF-IDF) is a popular traditional model to determine the weights of terms in vector space models to detect document or text similarities. Once the weight vectors are derived, a similarity technique like cosine similarity could be used to determine the similarity between a query and a document [9]. Another related method is Okapi BM25 [10], which is a scoring function mainly used by search engines to rank documents according to the relevance to a given query. It is a bag of words retrieval function and computes the score based on the appearance of each of the terms in the query in each document being considered. BM25 is not a single scoring function, but rather a combination of different scoring components and parameters based on term frequency and inverse document frequency.

Bag of Words of TF-IDF approaches cannot account for the similarity distance between different words based on the context. To address this issue, a number of methods have been developed focusing on learning a latent low dimensional representation of documents. Latent Semantic Indexing (LSI) and Latent Dirichlet Allocation (LDA) are two such techniques [11]. Latent Semantic Indexing [12] eigen decomposes the bag of words features space by using singular value decomposition. On the other hand, Latent Dirichlet Allocation [13] uses probabilistic models to categorize similar words into topics and uses the distribution of such topics to represent a document.

## III. Methodology

Our approach approximates users' information seeking behaviour [14], which includes domain experts and non-experts. Given an incoming question, the expert searches for similar questions in the corpus by determining relevant keywords from the incoming question and the knowledge domain of such keywords. Corpus questions that contain these keywords or even other similar keywords but from the same knowledge domain are candidates for matches. The expert also has a sense of semantically related words, such as synonyms and abbreviations, that could be matched in the corpus. Finally, the expert has knowledge of the context of the incoming question and what exactly is being asked.

### A. Pre-Processing

As part of pre-processing, the corpus questions are indexed. Regular expression tokenization is applied to each corpus question and stop words are filtered using the modified Glasgow Stop Words List from the TAPoRware project [1]. Similarly, the new incoming question is also pre-processed by tokenization and stop words filtering. In the next step, the incoming question's tokenized keywords are then matched with the corpus index and scored.

### B. Scoring

For keyword matching, we compute the TF-IDF for each keyword in the incoming question using the corpus index. The TF-IDF measure is well-known in literature as a weighting factor for word relevance in documents [15]. Given an incoming question, a matching score is determined for each corpus question as shown in Equation 1.

$$\mathcal{T}(w, C) = \frac{tf(w, C) \times log\frac{|D|}{df(w)}}{max_{tf}} \tag{1}$$

where $max_{tf} = max(tf(k, C))$, and $tf(k, C)$ is the term frequency of a keyword $k$ in the corpus question $C$, $|D|$ is the corpus size and $df(w)$ is document frequency of $w$ in the entire corpora. The best matches in the corpus are then determined by sorting the scores. The TF-IDF score is normalized by $max_{tf}$. Next, the score is adjusted by taking into account relevance heuristics and term expansions.

### C. Relevance Heuristics

We note that the title of the question is usually the summary of the question. Consequently, the keywords extracted from the titles of the incoming and corpus questions are given a higher score by a constant relevance weight as shown in Equation 2. For determining relevance within the question's body keywords, we observe that various non-stop words are also not pertinent. Consequently, we leverage domain knowledge and remove all non-medical terms from the body. The medical terms are determined by using Merriam-Webster's Medical Dictionary API [2]. We use the Lancaster Stemming algorithm [3] to account for word form variations, such as singularization and pluralization.

[1] TAPoRware Project, Glasgow Stop Words List, http://taporware.ualberta.ca/~taporware/cgi-bin/prototype/glasgowstoplist.txt, Retrieved June 30, 2015

[2] Merriam-Webster's Medical Dictionary with Audio, www.dictionaryapi.com/products/api-medical-dictionary.htm, Retrieved May 22, 2016

[3] NLTK 3.0 Documentation - Lancaster Stemmer, www.nltk.org/modules/nltk/stem/lancaster.html, Retrieved Jan 12, 2016

$$\begin{aligned}\texttt{Score}(Q, D) \\ &= \sum_{t \in Q.title, t \in D.title} w_i w_c \times \mathcal{T}(t, D) \\ &+ \sum_{t \in Q.title, t \notin D.title} w_i \times \mathcal{T}(t, D) \\ &+ \sum_{t \notin Q.title, t \in D.title} w_c \times \mathcal{T}(t, D) \\ &+ \sum_{t \notin Q.title, t \notin D.title} \mathcal{T}(t, D)\end{aligned} \tag{2}$$

In Equation 2, $Q.title$ refers to the title of the incoming question $Q$, and $D.title$ is the title of the corpus document $D$. $w_i$ is the weight for a word which is included in the title of the incoming question, but not included in the title of the corpus document. This implies that the term occurs in the body of the corpus document. $w_c$ is the weight for a word which is not included in the title of the incoming question, but included in the title of the corpus document. For words that occur in both of the titles of the incoming and corpus questions, we assign a weight $w_i \times w_c$. These weight parameters can be tuned, where $w_i, w_c \geq 1$.

### D. Term Expansion

*1) Semantic Relatedness:* Some words are semantically related but not similar, such as *libido* and *impotence*, or *sugar* and *glucose*. Most dictionaries, ontologies and thesauri we investigated do not group such words together. We ultimately used Moby Thesaurus to look up synonyms in situations where no matches are found for an incoming question's keyword. The TF-IDF for the synonyms is used as a score for the original keyword. The number of synonyms, `ns`, that match corpus questions is used as an offsetting weight to balance the possible inflation of scores for a keyword having many synonyms, i.e. $\texttt{Score}(C_j)$ = $\texttt{Score}(C_j)/\texttt{ns}$

*2) Abbreviations Expansion:* We expand any keywords that are medical abbreviations to their full forms. To get abbreviations, we use a subset of the list of medical abbreviations from Wikipedia "List of Medical Abbreviations"[4].

## IV. Results

### A. Evaluation Criteria

We follow the evaluation measure used at the HDAC 2015. In the challenge, domain experts identified up to three most similar queries for each incoming question. Systems were evaluated using the percentage of expert-identified results that were included in the system-identified results. As an example, suppose the domain expert identified $Q_6$ and $Q_{37}$ as the most similar queries to an incoming query $Q_1$, and a system identified $Q_{37}$, $Q_{52}$ and $Q_{74}$ as the most relevant queries. This system be scored and receive 50 points. It is exactly the same as the evaluation measure recall, which is a ratio of the number of correctly identified answers against the number of true answers.

### B. Towards a Bronze Standard for Question Retrieval

The HDAC 2015 dataset consists of 10 incoming questions as queries, and 95 questions as corpus. In order to determine the efficiency of our proposed methods, the experts' benchmark is needed to evaluate the output from the algorithm. To this end, we requested the HDAC 2015 organizers for their annotations of expert-identified results, but received no response. Consequently, we used the HDAC dataset to develop self-annotations for the top 3 corpus matches for each of the incoming questions. We also were grateful to receive annotations developed by another team of participants at the HDAC 2015 who are affiliated with Mayo Clinic. These two annotations are an initial step towards a benchmark for evaluation of question retrieval tasks. The average Jaccard/Tanimoto coefficient between these two annotations is 0.56.

### C. Optimization of Title Weights

Table I shows the recall results for different settings of the title weighing parameters. The first row represents the weights for corpus document titles, $w_c$, while the first column represents weights for incoming question titles, $w_i$. We tried a range of values, [0.5, 1.0, 1.5, ..., 10.0], for $w_i$ and $w_c$ and Table I shows the recall results for the range band of [0.5, 2.5] which includes the best question retrieval performance.

| ⇓ $w_i$ / $w_c$ ⇒ | 0.5 | 1.0 | 1.5 | 2.0 | 2.5 |
|---|---|---|---|---|---|
| 0.5 | 0.233 | 0.450 | 0.500 | 0.450 | 0.433 |
| 1.0 | 0.333 | 0.533 | 0.667 | 0.667 | 0.651 |
| 1.5 | 0.650 | 0.701 | 0.735 | **0.818** | 0.768 |
| 2.0 | 0.533 | 0.768 | **0.801** | 0.751 | 0.751 |
| 2.5 | 0.567 | 0.768 | 0.751 | 0.751 | 0.718 |

TABLE I
Optimization of Title Weights

Using self-annotations as the benchmark, the highest recall value is 0.818 for $w_i = 1.5$ and $w_c = 2.0$, while the next best value is 0.801 for $w_i = 2.0$ and $w_c = 1.5$. However, when we investigated the corresponding recall values for the Mayo Clinic annotations, we observed that the results were 0.584 and 0.601 respectively. In order to avoid bias towards our self-annotations, we ultimately used $w_i = 2.0$ and $w_c = 1.5$.

### D. Comparative Analysis

We evaluated various modeling techniques such as BM25, LDA, language model, LSI, vector space, Word2Vec, and NLTK/semantic similarity. Based on the Bronze Standard used, we obtain the best performance with our proposed algorithm. We get an average recall of 80.1% on our self-annotations and an average recall of 60.1% for the Mayo Clinic annotations. A recall comparison graph for all these methods using both annotations is shown in Figure 2.

### E. Discussion

*1) Contribution of Features to Performance:* Various parameters used in our proposed algorithm are compared against the recall in order to determine which features contribute to the observed performance, as shown in Table II.

[4] List of Medical Abbreviations, `www.wikipedia.org/wiki/List_of_medical_abbreviations`, Retrieved June 1, 2016

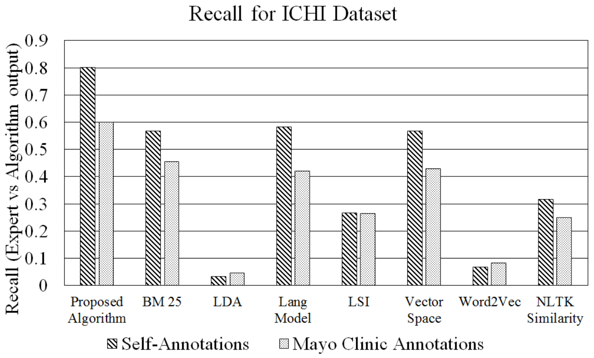


Fig. 2. ICHI Dataset Recall Results (Annotations vs Algorithm Outputs)

The reported recall is against the self-annotated benchmark. When the Moby thesaurus is excluded, performance drops because the terms with similar meaning but different lexical forms are ignored. A similar observation can be made for abbreviations and filtering of medical words. When these features are removed, the average recall decreases. The most significant decrease is observed when the weight heuristic feature is relaxed, leading to a sharp decline in the recall.

| Method | Recall |
|---|---|
| All features | 0.801 |
| Without thesaurus | 0.768 |
| Without abbreviations | 0.718 |
| Without medical dictionary | 0.684 |
| Without title weighting | 0.433 |

TABLE II

CONTRIBUTION OF EACH FEATURE TO PROPOSED ALGORITHM

*2) Error Analysis:* From unsuccessful instances, where the algorithm gave incorrect matches in the top 3 similar questions, we classified the error types as shown in Table III.

| Error Type | Proportion |
|---|---|
| Missing related terms | 0.17 |
| Missing abbreviations | 0.17 |
| Incorrect term weights | 0.17 |
| Others | 0.16 |

TABLE III

ERROR TYPES

The main errors are from missing related terms, i.e. synonyms and hyponyms, such as "exercise advice" and "yogic postures". Some missing abbreviations also were discovered. We also found some errors from incorrect weighting of terms. For example, given a question about "natural juice", the algorithm gave "natural supplements" as a relevant question due to the match for "natural", but "fruit juices" should be given more weight.

## V. CONCLUSION

We proposed a method for matching similar questions in health forums using weighted TF-IDF, relevance heuristics, and term expansions based on semantic relatedness and abbreviation expansions. Our experiments used the dataset from the HDAC 2015 challenge to demonstrate that our proposed algorithm outperformed other well-known methods. We also commenced preliminary work on gathering Bronze Standard data for evaluation of question retrieval systems. For future work, we will investigate giving different weights to terms based on their grammatical class, such as assigning higher weights to nouns. Other applications of question retrieval that we plan to explore include automatic generation of FAQs.

## ACKNOWLEDGEMENTS

The authors wish to thank Yanshan Wang, Research Fellow, Dept. of Health Sciences Research, Mayo Clinic, for providing his team's self-annotations from the HDAC 2015 challenge. The authors also thank the Alberta Machine Intelligence Institute for funding this research.

## REFERENCES

[1] H. Duan, Y. Cao, C.-Y. Lin, and Y. Yu, "Searching Questions by Identifying Question Topic and Question Focus," in *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL): Human Language Technologies (HLT)*, 2008, pp. 156–164.

[2] Z. Yu, B. C. Wallace, and T. R. Johnson, "Healthcare Data Analytics Challenge," in *Proceedings of IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE, 2015, pp. 515–516.

[3] R. D. Burke, K. J. Hammond, V. Kulyukin, S. L. Lytinen, N. Tomuro, and S. Schoenberg, "Question Answering from Frequently Asked Question Files: Experiences with the FAQ Finder System," *AI Magazine*, vol. 18, no. 2, p. 57, 1997.

[4] J. M. Ponte and W. B. Croft, "A Language Modeling Approach to Information Retrieval," in *Proceedings of ACM International SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1998, pp. 275–281.

[5] J. Jeon, W. B. Croft, and J. H. Lee, "Finding Similar Questions in Large Question and Answer Archives," in *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, 2005, pp. 84–90.

[6] G. Zhou, L. Cai, J. Zhao, and K. Liu, "Phrase-Based Translation Model for Question Retrieval in Community Question Answer Archives," in *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL): Human Language Technologies (HLT)*. Association for Computational Linguistics (ACL), 2011, pp. 653–662.

[7] S. Li and S. Manandhar, "Improving Question Recommendation by Exploiting Information Need," in *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL): Human Language Technologies (HLT)*. Association for Computational Linguistics (ACL), 2011, pp. 1425–1434.

[8] X. Xue, J. Jeon, and W. B. Croft, "Retrieval Models for Question and Answer Archives," in *Proceedings of ACM International SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2008, pp. 475–482.

[9] G. Salton, A. Wong, and C.-S. Yang, "A Vector Space Model for Automatic Indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[10] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford *et al.*, "Okapi at TREC-3," *NIST Special Publication*, vol. 109, p. 109, 1995.

[11] M. Kusner, Y. Sun, N. Kolkin, and K. Q. Weinberger, "From Word Embeddings To Document Distances," in *Proceedings of International Conference on Machine Learning (ICML-15)*, 2015, pp. 957–966.

[12] S. T. Dumais, "Latent Semantic Analysis," *Annual Review of Information Science and Technology*, vol. 38, no. 1, pp. 188–230, 2004.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[14] D. Ellis and M. Haugan, "Modelling the Information Seeking Patterns of Engineers and Research Scientists in an Industrial Environment," *Journal of Documentation*, vol. 53, no. 4, pp. 384–403, 1997.

[15] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," in *Proceedings of Instructional Conference on Machine Learning*, 2003.