IIITH at BioASQ Challange 2015 Task 3b: Bio-Medical Question Answering System

Harish Yenala[1], Avinash Kamineni[1],
Manish Shrivastava[1], and Manoj Chinnakotla[2]

[1] International Institute of Information Technology Hyderabad, India
[2] Microsoft, India
harish.yenala@research.iiit.ac.in,avinash.kamineni@research.iiit.ac.in
m.shrivastava@iiit.ac.in,manojc@microsoft.com

Abstract. In this paper, we describe our participation in the 2015 BioASQ challenge on Bio-Medical Question Answering. For Question Answering task (Task 3b), teams were provided with natural language questions and asked to retrieve responses from PubMed corpus in the form of documents, snippets, concepts and RDF triplets (Phase A) and direct answers (Phase B). For Phase A, we took the support of PubMed search engine and our snippet extraction technique. In our QA system, apart from the standard techniques discussed in literature, we tried the following novel techniques to - a) leverage web search results for improving question processing and b) identify domain words and define a new answer ranking function based on number of common domain words. We scored an F-measure of 0.193 for document extraction and F-measure of 0.0717 in snippet generation.

Keywords: PubMed; Biomedical Question Answering; Information Retrieval

1 Introduction

The present innovations in the bio-medical domain is leading to the creation of large amounts of data. The bio-medical literature growth can be understood from the vast data in PubMed [2] database of National Library of Medicine (NLM) which contains more than 14 million articles and hundreds of thousands more are being added every year [3]. However, such a huge repository of data is useful only if it can be easily accessed and the contents retrieved as per the user requirements [14]. Question Answering (QA) systems enable the user to express their information need in the form of natural language questions and retrieves the precise answers to them.

QA has been a well studied research area [7, 15, 8, 18]. However, QA in Bio-Medical Domain has its own challenges like presence of complex technical terms, compound words and domain specific semantic ontologies [16]. BioASQ-QA (Task 3b) [17] is a Bio-Medical Question Answering task which uses benchmark datasets containing development and test questions, in English, along with

gold standard answers. The benchmark datasets contain at least 500 questions. The participants have to respond with relevant concepts (from designated terminologies and ontologies), relevant articles (in English, from designated article repositories), relevant snippets, relevant RDF triples, exact answers (e.g., named entities in the case of factoid questions) and Ideal (summary) answers.

In this paper, we describe the approach taken by the IIIT-Hyderabad team for Task 3b of BioASQ Challenge. For retrieving documents as answers, we used *chunking*, *stop word removal* and *search query formulation* techniques. Later, from the top documents, we filter the most relevant phrases as snippets. For extracting the exact and Ideal answers from the snippets, we used cosine similarity and noun chunk identification techniques.

The rest of the paper is organized as follows: Section 2, describes the previous work done in Bio-Medical QA. Section 3 describes our approach in more detail. Section 4 gives the experimental results. Finally, Section 5 concludes the paper with future work.

2 Related Work

The Bio-medical Question Answering has been a challenge from past few years. There has been not much progress since then. The major challenge for this was a very complex domain. So, only domain expert could understand the inner details for system to be built. Major concentration was done on the factoid based questions and yes/no questions.

MedQA [11] is a bio-medical question answering system which has information retrieval, extraction, and summarization techniques to automatically generate paragraph-level answers for definitional questions. However, it is still limited due to its ability to answer only definitional questions. BioinQA [14] uses the technique of entity recognition and matching. It is based on the search in context and utilizes syntactic information. BioinQA also answers the comparison type questions from multiple documents, a feature which contrasts sharply with the existing search engines, which merely return answers from single document or passage.

The jikitou [5] system's architecture is composed of four subsystems: knowledge base, question analysis, answer agents, and user interface. Multiple software agents find possible answers to questions, and the most relevant one is presented to the user. Additional relevant information is presented to the user establishing a kind of dialog with the user to obtain feedback to refine the query.

OHSU (Oregon Health & Science University) [6] does multiple iterations of basic QA with each iteration successively refining the original question such as synonymy expansion, ranked series of topic queries and a range of specificities. Finally, they retrieve all the likely relevant passages in ranked order.

In [13] and [19], similarity between the question and snippet was computed using cosine similarity and was also used for ranking. They also used domain specific tools like MetaMap [4] for identifying concepts. Only few [9] have worked on extracting triples, from different linked data domains like disease ontology [10]

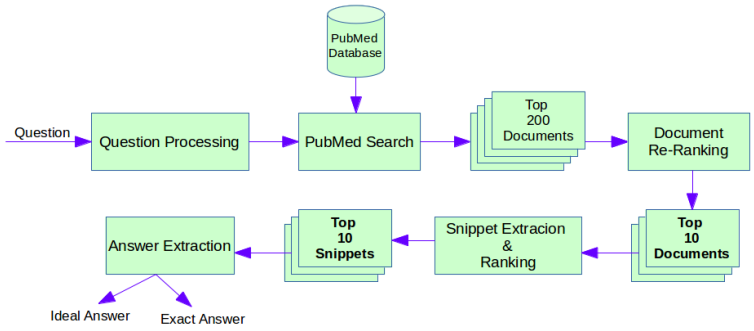

Fig. 1: System Architecture

and MeSH hierarchy [1]. While coming to document extraction, most systems uses PubMed Search, which has better ranking ability based on tf-idf [12] scores.

Our current approach differs with the above approaches in the following ways: a) We leverage web search results in question processing and b) We define new similarity metrics based on common domain words.

3 IIITH Bio-Medical QA System

We have designed an algorithm to extract the high informative documents for a given question from PubMed articles.

3.1 System Architecture

The architecture of the system is shown in Figure 1

3.2 Document Retrieval

This Algorithm takes a bio medical question 'Q' and outputs a set of 10 PubMed documents which are having high probability to contain answer 'A' for given Q. Detailed explanation of each step of algorithm is given below.

1. **Question Processing:** Let the given question be Q, we need to process the question to make it efficient and optimized for searching. For this we have sequence of sub steps which are explained below.
   (a) **Cleaning:** We clean the question for making the search efficient. In this step all unnecessary symbols like question mark(?) , dot(.) etc are removed. We have found that (-) makes the chunking task little difficult and wrong which is crucial task in this step. So Hyphens are replaced with some Named Entity Words.

Algorithm 1 Domain Word Extraction Algorithm

```
N ← Num_Of_Top_Results(20)
domainUrlPatterns ← {"nlm.nih.gov", "webmd.com", "medicine", "biocare", "drugs", ...}

function DomainWordIdentification(chunk, domainUrlPatterns)
    topResults ← SearchAPI(chunk, N)
    focusWord ← null
    for all result ∈ topResults do
        if result ∈ domainUrlPatterns then
            focusWord ← chunk
        else
            remove(chunk)
        end if
    end for
    return focusWord
end function

function SearchAPI(chunk, N)
    results ← GoogleAPI/BingAPI(chunk, N)
    return results
end function
```

(b) **Chunking:** We need to do chunking to get the phrases(chunks) to the modified Question Q. These chunks will be very useful to avoid removal of important(Focus) words which will be done in the next step. We used Annotator module from NLTK[3] Package for chunker.

(c) **Stop Word Removal:** The chunks with all the stop words will be removed. As they don't help at all in a keyword based search. This step makes the question more optimized. We have used NLTK corpus English StopWords list for this task.

(d) **Domain Word Identification:** In the processed query sentence Q, there will be generic words which may not be stop words. But, they contribute nothing in getting the relevant documents. Focus Word Identification step finds only the chunks which are Domain-words, Important Words. The pseudocode for identifying the focus word/chunk is shown at Algorithm 1.

We have observed the following list of url patterns that are most relevant to Bio-Medical domain[4].

After the question processing step, the question Q will be modified into set of Focus words, namely Set F_Q.

2. **PubMed Search:** The words in the Focus Word Set will be combined(concatenated) to make a single string. This string will be fired/searched in PubMed search engine and top 200 documents will be retrieved.

[3] Natural Language Toolkit `http://www.nltk.org/`

[4] Patterns in website urls which store bio-medical information "nlm.nih.gov", "webmd.com", "medicine", "biocare", "drugs"

Algorithm 2 Document Re-Ranking Algorithm

$Q \leftarrow Query$
$relDocs \leftarrow relaventDocuments$

function RANKALLDOCUMENTS($Q, relDocs$)
    $\overline{Q} \leftarrow$ REMOVESTOPWORDS(Q)
    $scores \leftarrow \{\}$
    for all $doc_i \in relDocs$ **do**
        $T_i \leftarrow doc_i.title$
        $\overline{T_i} \leftarrow$ REMOVESTOPWORDS(T_i)
        $scores[i] \leftarrow$ COSINESIM($\overline{Q}, \overline{T_i}$)
    end for
    $scores, topDocs \leftarrow$ SORTSCORES($scores, relDocs$)
    return $topDocs$
end function

3. **Document Re-Ranking:** As our question processing is not a standard one, we don't completely depend on PubMed ranking of documents. So we rank the obtained 200 documents again with our approaches.
   (a) **Cosine similarity [13]:**
   We take the original query Q, and Document Title T.
   (b) **Existence Test Score:** This measure tells number of common Focus words (Domain words) present in Question Q and title of the document T_i.

$$Scores[i] = ExistenceTestScore(\bar{Q}, T_i) \tag{1}$$

   (c) **Hybrid Approach:** Finally, we combined both the approaches giving them separate weights and interestingly the scores were better. In this approach the score[i] will be defined as below,

$$Score[i] = \alpha * CosSim(\bar{Q}, \bar{T_i}) + \beta * ExistTestScore(\bar{Q}, \bar{T_i}) \tag{2}$$

   α, β are normalization constants.
   After this step, high ranked which are most relevant 10 documents to the given query Q will be retrieved and output to the system.

3.3 Snippet Generation and Ranking

This section explains about retrieving top 10 snippets for given question Q. The algorithm for Snippet Generation and Ranking is shown in Figure 2

Initially we take the query Q and send in to the'Document Retrieval algorithm. From this step, we will get top 10 documents for the query Q.

After obtaining top 10 most relevant documents (D1...D10) we take abstracts of all those documents. From all the abstracts we extract all the sentences and make a set S.

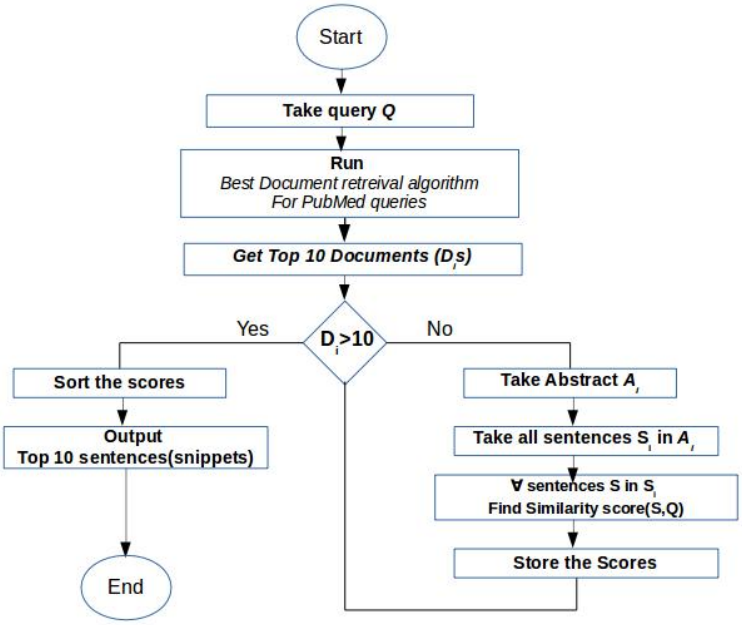

Fig. 2: Snippet Generation and Ranking Algorithm

For all the sentences s in S, we compute similarity scores with query Q. After finding similarity scores for all the sentences we sort and take top 10 sentences matching most with the query Q. We call those High-Matching sentences and snippets and output them to the system.

As we have parsed the PubMed web(search) page to get the abstracts and pmids, we have used Regular Expressions to identify abstracts and also to split sentences correctly. While finding similarity score between sentences s and query Q, we have used above explained 3 approaches and found *hybrid approach(Cosine similarity score+Existence Test score)* performance was good.

This algorithms gave good snippets in which always at least 4 of them contained the exact answer.

3.4 Ideal Answer Extraction

Here a Question and related snippets will be given to our algorithm and it should find Ideal answer i.e. a one or two line answer which perfectly answers the given query.

For this task we have taken all the sentences from the given snippets and calculated similarity scores with the approaches explained in section 3.2. Top 10 Ideal answers with highest similarity scores will be returned.

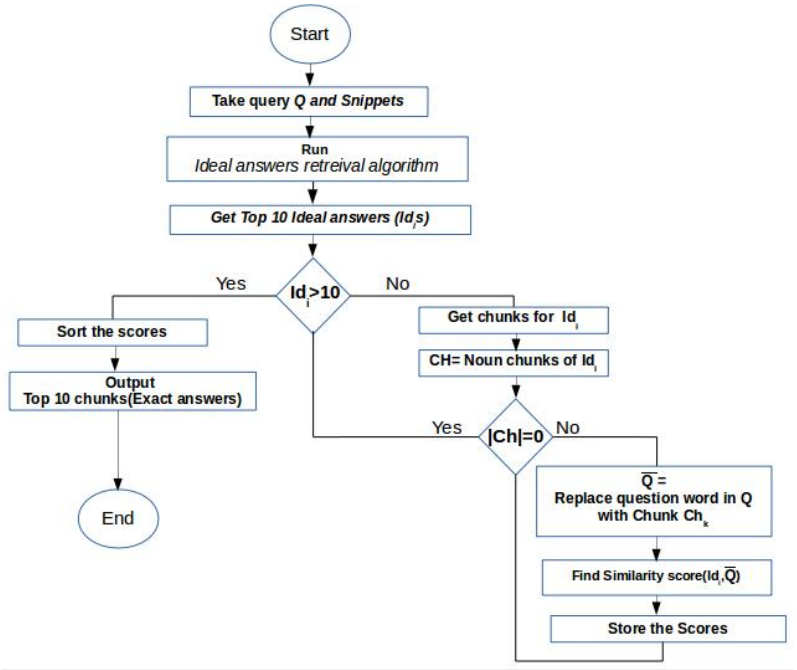

Fig. 3: Exact Answer Extraction Flow

3.5 Exact Answer Extraction

In this section we will be given with a query and set of snippets. We should result the exact one or two word answers. We have designed an algorithm for giving exact answers for Factoid questions. That approach is explained as a flow chart in Figure 3.

As shown in the Figure 3, we start with query Q and all snippets S. We apply, Phase A algorithm for finding top Ideal answers. For each, Ideal answer we find chunks and we consider only "Noun chunks" as we are dealing with only Factoid questions. Each time, we modify the given query Q by replacing question words (like what, when, who etc.) with the considered "Noun chunk", then we name that new query as $\bar{Q}$. Then, we find the similarity score between $\bar{Q}$ and Ideal answer I_{di}. Similarly, we repeat the same procedure for all ideal answers and their noun chunks. At the end, we sort the similarity scores and output top 10 related noun chunks as exact answers. In this algorithm, we used NLTK parser and Cosine similarity measurement. We got good results with proposed approaches.

4 Experiments and Results

As part of the BioASQ 3b challenge 2015, we have participated in the batch wise submissions. We could perform relatively better than the baseline System. These results of all submissions for documents and snippets are shown in Table 1 and Table 2.

4.1 Dataset Details

Task 3b of BioASQ 2015 released a training dataset [17] of 810 question-answer pairs and testing data comprising of 100 questions was released for five consecutive weeks. The average length of each question was 10 words. Out of 100 questions, on an average 30 factoid questions, 25 list type, 25 yesno type and 20 summary type of questions were there.

4.2 Evaluation Metrics

The evaluation metrics used in this task are mean precision, mean recall, and mean F -measure of the documents, snippets, exact and ideal answers returned by the system [17].

4.3 Experimental Setup

Different experiments have been conducted to improve the accuracy of system such as:

1. Increased the retrieved documents after *PubMed Search* step from 60 documents to 100 documents.
2. Defined new similarity measure called *Existance Similarity* and combined it with cosine similarity, which increased the accuracy of the system.

4.4 Results

Table 1: Document Extraction Results of All Batches

System Name	Mean precision	Recall	F-Measure	MAP	GMAP
batch1-qaiiit system 1	0.1957	0.1757	0.1559	0.1099	0.0006
batch2-qaiiit system 1	0.2379	0.2353	0.193	0.1092	0.0022
batch3-qaiiit system 1	0.1643	0.1719	0.1448	0.0569	0.0003
batch4-qaiiit system 1	0.2144	0.2376	0.1893	0.1057	0.0008

Our best results in snippet extraction for Task 3b Phase A has been achieved(see table 2, bold). Our system name for submission was "qaiiit system 1". In fact that was achieved in statistical approach. This approach was found better compared to baseline system.

Table 2: Snippet Generation and Ranking Results of All Batches

System Name	Mean precision	Recall	F-Measure	MAP	GMAP
batch1-qaiiit system 1	0.0616	0.0697	0.0511	0.0545	0.0002
batch2-qaiiit system 1	**0.0819**	**0.0889**	**0.0717**	**0.0709**	**0.0004**
batch4-qaiiit system 1	0.0976	0.0844	0.0816	0.0913	0.0003

Table 3: Ideal Answer Batch 2 Submission

	Automatic scores	
System Name	**Rouge-2**	**Rouge-SU4**
qaiiit system 1	0.3036	0.3299

In Table 3, from the snippets given by the BioASQ, we found most relevant snippet based on hybrid similarity match and returned respective snippet as the ideal answer. By using this method, rougue score was about 0.303, just below the baseline of 0.46.

5 Conclusion and Future Work

In this paper, we describe the approach taken by IIIT-H team for the Bio-Medical Question Answering task of BioASQ task 3B. Apart from the standard QA techniques, our current approach differs in the following ways - a) We leverage web search results in question processing and b) We define new similarity metrics

based on common domain words. By applying our approach, we obtained F-measure of 0.193 for document extraction and F-measure of 0.0717 in snippet generation.

As part of the future work, we would be working on the Triples extraction, which is still under progress, as no one has even attempted it. We will also be using more domain specific entity recognition tools, for more cleaner way of identifying the exact answer. For the ideal answer or summary type answers, we are working on answer generation techniques from snippets of multiple documents.

References

1. Medical Subject Headings https://www.nlm.nih.gov/mesh/.
2. Search Engine of Medline database http://www.ncbi.nlm.nih.gov/pubmed.
3. U.S. National Library of Medicine https://www.nlm.nih.gov/.
4. A. R. Aronson. Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The Metamap Program. *Proc AMIA Symp*, pages 17–21, 2001.
5. Michael Anton Bauer. The jikitou Biomedical Question Answering System: Facilitating the Next Stage in the Evolution of Information Retrieval, 2008.
6. Aaron M. Cohen, Jianji Yang, Seeger Fisher, Brian Roark, and William R. Hersh. The OHSU Biomedical Question Answering System Framework. In Ellen M. Voorhees and Lori P. Buckland, editors, *TREC*, volume Special Publication 500-274. National Institute of Standards and Technology (NIST), 2007.
7. Richard J Cooper and Stefan M Rüger. A Simple Question Answering System. In *TREC*, 2000.
8. David A. Ferrucci. Ibm's Watson/Deepqa. *SIGARCH Comput. Archit. News*, 39(3):–, June 2011.
9. Konrad Höffner and Jens Lehmann. Towards Question Answering on Statistical Linked Data. In *Proceedings of the 10th International Conference on Semantic Systems*, SEM '14, pages 61–64, New York, NY, USA, 2014. ACM.
10. Warren A. Kibbe, Cesar Arze, Victor Felix, Elvira Mitraka, Evan Bolton, Gang Fu, Christopher J. Mungall, Janos X. Binder, James Malone, Drashtti Vasant, Helen Parkinson, and Lynn M. Schriml. Disease Ontology 2015 update: An Expanded and Updated Database of Human Diseases for Linking Biomedical Knowledge through Disease Data. *Nucleic Acids Research*, 43(D1):D1071–D1078, January 2015.
11. Minsuk Lee, James Cimino, Hai Ran Zhu, Carl Sable, Vijay Shanker, John Ely, and Hong Yu. Beyond Information RetrievalMedical Question Answering. In *AMIA annual symposium proceedings*, volume 2006, page 469. American Medical Informatics Association, 2006.
12. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.
13. Yuqing Mao, Chih-Hsuan Wei, and Zhiyong Lu. NCBI at the 2014 BioASQ challenge task: Large-scale Biomedical Semantic Indexing and Question Answering. In *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014.*, pages 1319–1327, 2014.
14. Sparsh Mittal, Saket Gupta, Ankush Mittal, and Sumit Bhatia. Bioinqa: Addressing Bottlenecks of Biomedical Domain through Biomedical Question Answering System. In *The International Conference on Systemics, Cybernetics and Informatics (ICSCI-2008)*, pages 98–103, 2008.

15. Jun-Ping Ng and Min-Yen Kan. QANUS: An Open-Source Question-Answering Platform. *arXiv preprint arXiv:1501.00311*, 2015.
16. Stefan Schulz, Martin Honeck, and Udo Hahn. Biomedical Text Retrieval in Languages with a Complex Morphology. In *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain-Volume 3*, pages 61–68. Association for Computational Linguistics, 2002.
17. George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. An Overview of the BioASQ Large-Scale Biomedical Semantic Indexing and Question Answering Competition. *BMC bioinformatics*, 16(1):138, 2015.
18. Menno van Zaanen. Multi-Lingual Question Answering using OpenEphyra. In *Working Notes of CLEF 2008*, 2008.
19. Dirk Weissenborn, George Tsatsaronis, and Michael Schroeder. Answering Factoid Questions in the Biomedical Domain. In Axel-Cyrille Ngonga Ngomo and George Paliouras, editors, *BioASQ@CLEF*, volume 1094 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.