HPI Question Answering System in BioASQ 2016

Frederik Schulze, Ricarda Schüler, Tim Draeger, Daniel Dummer,
Alexander Ernst, Pedro Flemming, Cindy Perscheid, Mariana Neves
Hasso-Plattner Institute
August-Bebel-Str. 88
Potsdam, Brandenburg, 14482 Germany
mariana.neves@hpi.de

Abstract

Question answering (QA) systems are crucial when searching for exact answers for natural language questions in the biomedical domain. Answers to many of such questions can be extracted from the 26 millions biomedical publications currently included in MEDLINE when relying on appropriate natural language processing (NLP) tools. In this work we describe our participation in the task 4b of the BioASQ challenge using two QA systems that we developed for biomedicine. Preliminary results show that our systems achieved first and second positions in the snippet retrieval sub-task and for the generation of ideal answers.

1 Introduction

The deluge of scientific publication in biomedicine requires tools for processing and searching precise information in real time. Question answering (QA) comes as an alternative to standard search engines system, e.g. PubMed[1], and provides precise and short answers for questions in natural language (Athenikos and Han, 2010; Neves and Leser, 2015). One of the advantages of QA systems is that the user does not need to be proficient in formulating queries in a way that the system can understand. Instead, a user may simply enter a question as they would pose it to another person and receive a answer in return. Thus, no formal training is required to use QA systems.

QA is one of the more complex applications of natural language processing (NLP) (Jurafsky and Martin, 2013). This is usually achieved through a three-steps architecture: (1) the users question must be processed so that a query can be generated; (2) this query is then used to find all relevant text passages from a large document collection; and (3) finally, the system generates the exact answer to the users question and/or a summary of the facts from these passages. Some QA systems already exist for the biomedical domain (Bauer and Berleant, 2012). However, none of them are capable of answering questions in real time, in part due to the large collections of documents involved in the task.

We describe our participation in the fourth edition of the BioASQ challenge[2] (Tsatsaronis et al., 2015), a community-based shared task which aims to evaluate the current solutions for a variety of QA sub-tasks. We submitted runs from two QA systems which were specifically developed for the biomedical domain. One of the system (HPI1) successfully participated in the previous editions of the BioASQ challenge (Neves, 2015) and our second system (HPI2) is described in this work. We relied on existing NLP functionality from a in-memory database (IMDB) and we extend it with new procedures tailored specifically to QA. We participated in the task 4b (Biomedical Semantic QA) which is split in two phases: (a) phase A: concept mapping and document, passage and RDF triples retrieval; and (b) phase B: exact and ideal (short summary) answers.

The next section presents a short description of our the HPI2 system, followed by the preliminary results that we obtained in the challenge and a short discussion about our performance and methods.

2 Data

We relied on two main resources when developing our QA system: the MEDLINE and the Unified

[1] http://www.ncbi.nlm.nih.gov/pubmed

[2] http://bioasq.org/

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 38–44,
Berlin, Germany, August 7-12, 2016.

Medical Language System (UMLS). In this section, we give a short overview on both resources.

2.1 MEDLINE

MEDLINE[3] is the main source for biomedical publications and grows continuously. We downloaded the publications from MEDLINE and integrated them into our local database. For the purposes of our QA system, an article consists of a title, an abstract and the main text. In this paper we refer only to titles and abstracts, as full papers are not considered in the current edition of the BioASQ challenge.

2.2 Unified Medical Language System

Extracting meaning out of biomedical documents is usually supported by manually curated dictionaries. These dictionaries contain words and phrases which are common to the biomedical domain. such dictionaries are used to map synonyms and abbreviations of terms to a common base term. Often, they also contain information to assign categories to terms. There are various terminologies for the biomedical domain, such as UMLS, SNOMED CT or MeSH.

UMLS[4] is a comprehensive database that combine various sources into a single knowledge base. It includes vocabularies mapping words and phrases onto a set of concepts. Each concept has an associated semantic type and group, which classifies the category of the concept, such as gene or disease.

In our QA system, UMLS was mainly used for named-entity recognition (NER), i.e., for extracting named-entities both in the question and in the document collection. Also in the context of NER, we used the UMLS semantic types to map the named-entities to their corresponding types. Finally, we also rely on UMLS to resolve synonyms, thus avoiding to miss important passages which include synonyms to the words in the questions. Abbreviations, in particular, are very frequent in biomedical documents.

3 Methods

Our QA is composed of many components (cf. Figure 1) which are included in three main steps, i.e., question processing, document retrieval, and answer processing. The later includes a two-step phase: exact answer extraction (not included in this paper) and summarization. Details for each component are described below.

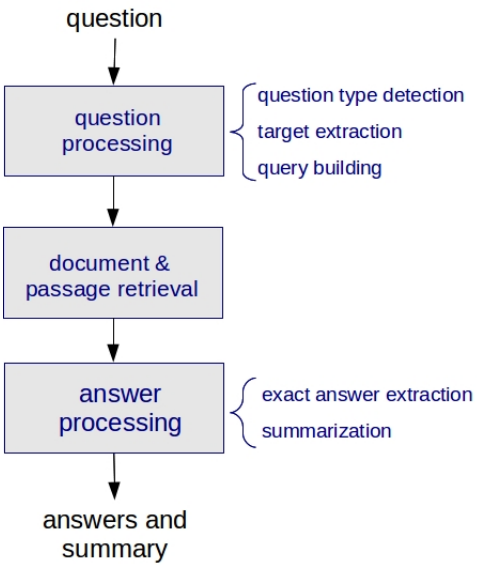

Figure 1: Work-flow of our question answering system.

3.1 Architecture

Our system was developed on top of a IMDB (SAP HANA database) (Plattner, 2013), which allows fast access of data directly from main memory, in contrast to processing data from files that reside on disk space, thus requiring loading data into main memory. The IMDB we used comes with built-in text analysis features, such as language detection, sentence splitting, tokenization, stemming, part-of-speech (POS) tagging, NER based on pre-compiled dictionaries, information extraction based on manually crafted rules, document indexing, approximate searching and sentiment analysis.

All textual resources (documents and questions) were added to the database and dictionaries of biomedical terms were created based on the UMLS terminology. Then we created the so-called full text index (FTI), i.e., an additional table which can be created for columns which contain text. Such an index can be created in many ways, we opted for two of them, namely: (a) a linguistic index, which contains all words from the original documents, as well as corresponding POS tags; and (b) a NER index, which contains all entities that were found based on the dictionary that was

[3] https://www.nlm.nih.gov/pubs/factsheets/medline.html

[4] https://www.nlm.nih.gov/research/umls/

previously built. In summary, from the linguistic FTI it is possible to retrieve information about sentence splitting, tokenization, stemming and POS tags, while the NER provides the named-entities.

3.2 Question processing

The first step in a question answering system is to analyze the input question. This step is composed of three components in our QA system: (a) question type detection, (b) target extraction, and (c) query building.

Question Type Detection. The question type can be either "yes/no", "factoid", "list" or "summary". It defines which kind of the answer the system needs to return. In this step, we split the question into words and and apply special rules to find the correct type, by considering question words and the structure (POS tags) of the question. Our approach is based on regular expressions, for instance, a questions beginning with an auxiliary verb is classified as yes/no-question. Although our QA system includes a component for detecting the question type, this step is not necessary in the BioASQ challenge because all question types are given.

Target Extraction. The second component of our question processing step extracts the target of the question, in case of factoid questions, and classifies it according to the UMLS semantic types[5], e.g, whether the question asks for a disease or a gene. This is an important information for the answer extraction step. We extract the headword using simple rules, for instance, the first noun after the question word (e.g., "what", "which"). For classifying the headwords according to the many UMLS semantic types, and inspired by (Huang et al., 2008), we relied on a machine learning (ML) approach based on the implementation of the Support Vector Machine (SVM) algorithm in the IMDB database. The features that we use were the headwords and the questions words. All headwords in the factoid questions were manually classified into the semantic types by one of the authors (MN) and this is the training data that was used in our experiments. During the process, several different features were evaluated, but they did not improved our results.

[5] https://metamap.nlm.nih.gov/SemanticTypesAndGroups.shtml

Query Building. Good query terms are important features when relying on a keyword-based search to find relevant documents for the question. For this purpose, we use all words, except for stopwords and question words (e.g., "what", "which").

3.3 Document and Passage Retrieval

The query that was built in the previous step was used in this step to find relevant documents and passages within the millions abstracts. We relied on the tf-idf method (Manning et al., 2008) as a basis and we adjusted it by various means to better fit the biomedical domain. We opted for the weighted tf-idf approach since our experiments showed that it provided up to 10% more recall than an equally weighted approach. We used a proximity measure to boosts a documents relevancy rating when it contains words from the query which appear close together. This measure searches for each possible word pair that appears in the query and applies a fixed rating increase for each such pair that is separated by a maximum of two words anywhere in the document.

We also consider the documents title in our approach. A documents titles relevancy was added to the documents relevancy in a weighted sum, thereby increasing the relevancy of documents with relevant titles. We also utilized a Jaccard-based word overlap measure between sentences in the document and in the question for the passage retrieval step. Our system first retrieves the 100.000 most relevant documents and then checks their sentences. This way we achieve a significant speed-up compared to calculating relevancy scores for all sentences in all documents. The document's total proximity score and the best sentence's word overlap score are then used to boost the initial tf-idf score. Their influence was tuned empirically on a test set of BioASQ questions and answers. Finally, our document and passage retrieval algorithms return a list of documents or passages, sorted by their relevance score.

3.4 Answer extraction

We only submitted ideal answers, i.e. short summaries, for the BioASQ challenge. Our approach is described in details below.

For the generation of summaries, we used an algorithm that is based on LexRank (Erkan and Radev, 2004), but that solely used the named-entities for the similarity function. In other words,

instead of using tf/idf values to rate the importance of each word, we use the named-entities instead.

The first step was to build a sentence graph. Therefore we calculated the cosine similarity of each sentence with each other sentences, i.e., a vector representation of each sentence. However, instead of using each word as dimension for the vector, we only use the named-entities. After the construction of the vectors, we calculate the cosine similarity (cf. equation 1) between each two of these:

$$\text{cosine} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

where A_i and B_i are the dimensions of the vectors representing the sentences. Afterwards, we create the sentence graph by adding a vertex for each sentence. Then we create edges between those vertices whose corresponding sentences have a similarity score above 0.2.

For calculating the ranking, we used the exact round based formula (cf. equation 2) that is used in LexRank and that originates from PageRank (Page et al., 1999):

$$\text{score}(s_i) = \frac{d}{N} + (1-d) \sum_{s_j \in adj[s_i]} \frac{\text{score}(s_j)}{deg(s_j)} \quad (2)$$

where N is the total number of vertices in the graph, $adj[s]$ are all adjacent vertices of the vertex s. Additionally, we have the parameter d, a 'damping factor', which is typically set to 0.2 (Page et al., 1999).

Subsequently, we ranked all sentences according to their centrality in the set of related abstracts. We need a last step to generate a summary by removing redundant sentences and we follow the following process:

1. Initialize two sets: (a) an empty set A and a set B that contains all extracted sentences.
2. Order the sentences in set B by decreasing order of their score.
3. Move the top sentence s_i from set B to set A. Then penalize all sentences s_j whose similarity to s_i is greater than a threshold of 0.3 by multiplying their score with the penalty factor of 0.5.
4. Repeat the steps 2 and 3 until enough sentences are in set A.

In a final step, we order the sentences from set A according to their occurrence in the original documents. Thus, we tried to roughly keep the sentence at the position that the author intended.

4 Results and Discussion

In this section, we present the preliminary results we obtained in the fourth edition of the BioASQ challenge. We introduce the details of the BioASQ challenge and then present our results for the two systems with which we participated this year.

4.1 BioASQ challenge

We participated on the Task 4b, which is composed of two phases: A and B. During phase A, the participating teams received a test set of 100 questions along with their question type, i.e., whether yes/no, factoid, list or summary, and had 24 hours to submit their predictions for concepts, documents, passages and RDF triples. When phase A was over, the organizers released the the test set for phase B which contained the same questions previously released for phase A along with gold-standard annotations. During phase B, the participating teams had 24 hours to submit their predictions for exact and ideal answers.

The BioASQ organizers released five bathes of around 100 questions every two weeks. Although our QA systems are capable to output results for most of the tasks covered in BioASQ, we did not submit runs for every sub-task due to problems with the systems, which are still under development.

4.2 Systems

We participated this year with two QA systems, as identified by their run names:

1. HPI1: our previous system that participated in the BioASQ challenge last year (Neves, 2015);
2. HPI2: our new QA system, which is described in this work.

HPI1 is exactly the same system that participated in the BioASQ 2015 and that was one of the winners systems[6]. We made no changes in the

[6] http://www.bioasq.org/participate/third-challenge-winners

system and details on the methods can be found in our previous publication (Neves, 2015). This system was used this year for concept matching and for document and snippet retrieval. The only change made to this system was on the dictionaries which are used in the concept matching task of Phase A. The dictionaries were re-created based on newer versions of the five terminologies specified in the guidelines of the BioASQ challenge: DO, MeSH, Jochem, GO and Uniprot. We downloaded the original files from the respective web sites and compiled dictionaries for each of the terminologies. The dictionaries include various names and synonyms for each concept and was used by the built-in NER functionality of the database to match concepts to the questions.

The document and passage retrieval of the HPI1 system did not make use of our local copy of MEDLINE but it queries PubMed instead. For each question, we generate two queries based on its tokens: (1) by using the "OR" operator and words in the question, except stopwords, and (2) by using the "AND" operator and using all words in the question, except stopwords and words in list of common English words (cf. (Neves, 2015)). We retrieve up to 200 PubMed documents for each of the queries and index these in the IMDB. We rank the sentences for each question based on an approximate similarity between the words in the question and the ones in the document, while a score is automatically calculate between those. Finally, we rank the sentences according to the sum of scores of the matching words and select the top 10 sentences. The list of up to 10 documents is derived from the list to top 10 sentences, i.e., the corresponding documents of these sentences, in the same order.

4.3 Evaluation

Currently, only preliminary results are available for some of the tasks of the BioASQ challenge. We summarize them in Table 1. More details on the results can be found in the BioASQ web site [7].

We present in this section a discussion on the preliminary results that we obtained in the BioASQ challenge, on the limitation of our methods and improvements for future versions of our QA system.

[7] http://participants-area.bioasq.org/results/4b/phaseA/ and http://participants-area.bioasq.org/results/4b/phaseB/

	HPI1	**HPI2**
Concepts	**MAP**	**MAP**
batch1	na	-
batch2	-	-
batch3	na	-
batch4	na	-
batch5	na	-
Documents	**MAP**	**MAP**
batch1	0.0474 (12/15)	0.0028 (15/15)
batch2	-	-
batch3	0.0674 (16/18)	0.0006 (18/18)
batch4	-	-
batch5	0.434 (16/21)	-
Snippets	**MAP**	**MAP**
batch1	0.0481 (1/7)	-
batch2	-	-
batch3	0.0715 (4/14)	-
batch4	-	-
batch5	0.0510 (5/16)	-
Ideal Answ.	**Rouge-2**	**Rouge-2**
batch1	-	0.2231 (1/2)
batch2	-	0.2240 (6/7)
batch3	-	0.2559 (6/7)
batch4	-	0.2280 (4/4)
batch5	-	0.3233 (6/7)

Table 1: Preliminary results in the BioASQ task 4b. Scores for concepts, documents and snippets are in terms of MAP (Mean Average Precision). "na" indicated that results are still not available for this task, while "-" indicated that we did not submit any run for the task. The values inside parameters indicate our current rank and the total number of submissions for the task.

Documents. Curiously, although the strategy used for the document retrieval is exactly the same one used for the snippet retrieval, we obtained much better results for the later, in term of position in the ranking, also in previous editions of the BioASQ challenge. As gold-standard and not available, we can only try to guess the reasons for our performance. When comparing our two systems, HPI2 performed much worse than HPI1, which proves that we still have to need to be improved to deal with large document collections, while HPI1 rely on up to 200 previously retrieved from PubMed.

Snippets. Our system HPI1 performed well again and it a good candidate for obtaining first and second position in the challenge. This proves that the IMDB could effectively match the keywords in the queries to the documents and rank the sentences. However, we see much room for improvement in our approach as named-entities are still not being used in this component, a step which can certainly improve both document and passage retrieval.

Ideal Answers. Our results for ideal answers, i.e., short summaries, provided by system HPI2 also obtained either first or second positions in the all of the batches, when considering results by teams, instead of each individual run.

5 Conclusions and Future Work

In this work we present our results for our two QA systems that participated in task 4b of the BioASQ challenge. The preliminary results show that our approaches are obtained top positions for the snippet retrieval and for the ideal answers. Regarding future work, we envisage much room for improvement for our HPI2 system, the one which is currently under development in our group:

- Both the document and snippet retrieval steps performed much worse than the HPI1 system, which rely on PubMed API. Future work should aim at improving our current ranking algorithms.

- We did not submit runs for factoid and list questions because our system could not return any answer for most of the answers. We did submit one run for yes/no questions but MAP value was of only 25%, while other system are close to 100%. We should perform a comprehensive evaluation of the question processing step, specially the target identification step, and properly integrate further components which can potentially boost our results, such as NER, chunking and semantic role labeling.

Finally, we should perform a comprehensive evaluation on biomedical corpora for the many built-in NLP components of the IMDB, such as NER and POS tagging, as mistakes returned by these can be propagated throughout the system.

Acknowledgments

We thank David Heller, Thomas Hille and Fabian Eckert for the interesting discussions during the project and the HPI Future Soc Lab Team for providing us the access to the IMDB. Finally, we also would like to thank technical support from the students of our current Bachelor Project: Maximilian Goetz, Marcel Jankrift, Julian Niedermeier, Toni Stachewicz and Soeren Tietboehl.

References

Sofia J. Athenikos and Hyoil Han. 2010. Biomedical question answering: A survey. *Computer Methods and Programs in Biomedicine*, 99(1):1 – 24.

MichaelA Bauer and Daniel Berleant. 2012. Usability survey of biomedical question answering systems. *Human Genomics*, 6(1):1–4.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, pages 457–479.

Zhiheng Huang, Marcus Thint, and Zengchang Qin. 2008. Question classification using head words and their hypernyms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 927–936, Stroudsburg, PA, USA. Association for Computational Linguistics.

Daniel Jurafsky and James H. Martin. 2013. *Speech and Language Processing*. Prentice Hall International, 2 revised edition.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.

Mariana Neves and Ulf Leser. 2015. Question answering for biology. *Methods*, 74:36 – 46.

Mariana Neves. 2015. HPI question answering system in the bioasq 2015 challenge. In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015.*

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: bringing order to the web.

Hasso Plattner. 2013. *A Course in In-Memory Data Management: The Inner Mechanics of In-Memory Databases*. Springer, 1st edition.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):138.