UNCC QA: A Biomedical Question Answering System

Abhishek Bhandwaldar
Department of Computer Science
UNC Charlotte
abhandwa@uncc.edu

Wlodek Zadrozny
Department of Computer Science
UNC Charlotte
wzadrozn@uncc.edu

Abstract

In this paper, we detail our submission to the BioASQ competition's Biomedical Semantic Question and Answering task. Our system uses extractive summarization techniques to generate answers and has scored highest ROUGE-2 and Rogue-SU4 in all test batch sets.

Our contributions are named-entity based method for answering factoid and list questions, and an extractive summarization techniques for building paragraph-sized summaries, based on lexical chains. Our system got highest ROUGE-2 and ROUGE-SU4 scores for ideal-type answers in all test batch sets.

We also discuss the limitations of the described system, such lack of the evaluation on other criteria (e.g. manual). Also, for factoid- and list -type question our system got low accuracy (which suggests that our algorithm needs to improve in the ranking of entities).

1 Introduction

Most of the recent question answering (QA) systems produce either `factoid` type answers (typically, a phrase or a short sentence) or a summary (typically, returning a few sentences or passages from the text). Creating a natural language answer from relevant passages is still an open problem. Our paper presents is also about providing `factoid` and `summary` answers in BioASQ.

BioASQ is a research competition which is organized by tracks in the biomedical domain. Namely, large-scale online biomedical semantic indexing, biomedical semantic question answering, and information extraction from biomedical literature.

The biomedical QA task is organized in two phases. Phase A deals with retrieval of the relevant document, snippets, concepts, and RDF triples, and phase B deals with exact and `ideal` answer generations. Exact answer generation is required for `factoid`, `list`, and `yes/no` type question. `ideal` answer is required for all the question. An `ideal` answer is a paragraph-sized summary of snippets.

BioASQ competition provides the train and test dataset. The training dataset consists of questions, golden standard documents, concepts, and `ideal` answers. The test dataset is split between phase A and phase B. The phase A dataset consists of the questions, unique ids, question types. The phase B dataset consists of the questions, golden standard documents and snippets, unique ids, and question types. Exact answers for `factoid` type questions are evaluated using strict accuracy, lenient accuracy, and MRR (Mean Reciprocal Rank). Answers for the `list` type question are evaluated based on precision, recall, and F-measure. `ideal` answers are evaluated using automatic and manual scores. Automatic evaluation scores consist of ROUGE-2 and ROUGE-SU4 and manual evaluation is done by measuring readability, repetition, recall, and precision.

Summary of our results. In this paper, we present our submission for BioASQ competition. We describe two methods, evaluated on two BioASQ tasks: `ideal` answer and `factoid` type questions. Both methods use conceptual representations based on MetaMap and UMLS.

We compute answers by choosing sentences with the concept chains that are similar to concepts in the question. In `factoid` questions, additionally, our method selects the entities with the highest idf scores.

The first method obtains the best rank for test batch 2,3 and 5 of Phase B of Task 6B. The second method was evaluated on previous year tests with mediocre results.

Proceedings of the 2018 EMNLP Workshop BioASQ: Large-scale Biomedical Semantic Indexing and Question Answering, pages 66–71
Brussels, Belgium, November 1st, 2018. ©2018 Association for Computational Linguistics

2 Related Work

Previous submissions to BioASQ show different approaches by teams taken for answering `ideal`, `factoid`, `list` and `yes/no` questions. OAQA systems(Chandu et al., 2017) use extractive summarization technique for answering `ideal` questions. They have used agglomerative clustering algorithm for similar sentence selection and MMR (Maximal Marginal Relevance) as a sentence similarity measure. Olelo (Neves et al., 2017) proposes a system for getting `yes/no`, `factoid`, `list` and `summary` type question. For summary based questions, the system selects snippets with greatest semantic similarity to question. For `factoid` and `list` type questions, they select an answer, based on matching predicates, and for `yes/no` question, they do sentiment analysis. (Aliod, 2017) have submitted the system for `ideal` answers only. They propose an extractive summarization approach that does sentence segmentation, ranks the sentences based using a scoring function, and return the top n sentences as the answer.

(Sarrouti and Alaoui, 2017) describes a system which retrieves snippets from relevant documents, re-ranks using BM25 model and finally concatenates top two snippets. For `factoid` and `list` type, their system extracts biomedical entities from relevant snippets, ranks them based on their frequency, and return top n. For `yes/no` they use sentiment analysis.

(Wiese et al., 2017) proposes a deep learning based approach to answering the `factoid` and `list` type question. The system is based on FastQA (Weissenborn et al., 2017), which is trained on SQUAD dataset and fine-tuned on BioASQ dataset to select a substring in relevant snippets as the final answer.

Other non-BioASQ systems that are capable of question answering include IMB's Watson (Ferrucci, 2012). The Watson system is an open domain question answering system that won the TV game-show Jeopardy! in 2011. The system worked by pipelining different components like question decomposition, hypothesis generation, hypothesis and evidence scoring, and answer generation. A more recent approach using deep learning is dynamic memory networks(Kumar et al., 2015). It uses the SQUAD dataset and simulates episodic memory using recurrent neural networks; it can also answer questions that require transitive reasoning. The SQUAD dataset is a reading comprehension dataset that requires the system to find a segment of text as the answer for a given question. Most of the systems based on SQUAD dataset are factoid answering system, and do not generate natural language answers.

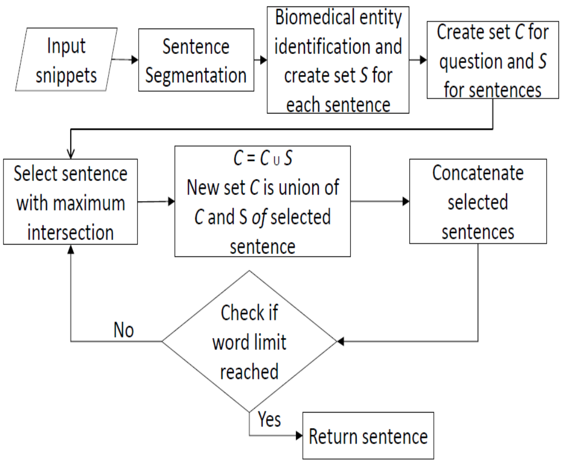
Input snippets
Sentence Segmentation
Biomedical entity identification and create set *S* for each sentence
Create set *C* for question and *S* for sentences
Select sentence with maximum intersection
C = *C* ∪ *S*
New set *C* is union of *C* and S *of* selected sentence
Concatenate selected sentences
Check if word limit reached
No
Yes
Return sentence

Figure 1: Our `summary` type question pipeline. The input is a list of snippets and the biomedical entity identification is done using MetaMap and UMLS.

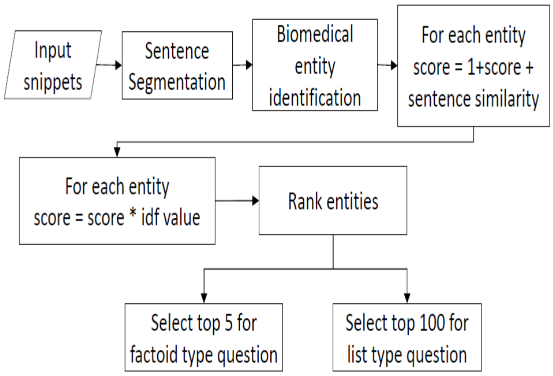
Input snippets
Sentence Segmentation
Biomedical entity identification
For each entity score = 1+score + sentence similarity
For each entity score = score * idf value
Rank entities
Select top 5 for factoid type question
Select top 100 for list type question

Figure 2: `factoid` and `list` type question pipeline. The input is a list of snippets and the biomedical entity identification is done using MetaMap and UMLS. The entities are scored using frequency, idf, and sentence similarity score.

3 Our Question Answering Pipeline

3.1 Ideal Answer

For `ideal` question type, we use extractive summarization to generate the answer. The word limit of the `ideal` answer is 200 words. Our extractive summarization pipeline is inspired by lexical chaining.

3.1.1 Lexical Chaining

Lexical chaining is a technique for identifying semantically related words that represent the concept or the semantic meaning of a sentence. A lexical chain does not describe the grammatical structure of a sentence. This technique has been used for text summarization by ranking sentences with similar ideas. Top sentences are then combined to produce a final summary. They are also helpful in word sense disambiguation (Okumura and Honda, 1994) by providing context to a term and capturing concept represented by that term. (Xiong et al., 2013) describes a lexical chain based approach for machine translation. Implementations of the lexical chain approach vary based on their applications. For example, (Reeve et al., 2006) describes the use of lexical chains for biomedical document summarization. Their technique uses a chain of concepts found by mapping biomedical terms to concepts using UMLS(Unified Medical Language System). UMLS is a meta thesaurus for biomedical terms. Once the concepts are found the strongest chain is found by sorting the chains based on a scoring function that takes into account various factors like word frequency, distinct concepts, word distance, and homogeneity. Finally, top sentences are used to generate the final summary.

3.1.2 Our Approach

For `ideal` answers, we use extractive summarization technique on relevant snippets. Our extractive summarization pipeline uses lexical chaining for sentence similarity and ranking. We then select the top N sentences such that the total number of words doesn't exceed the 200-word limit, and concatenate them to form the summary or the final answer. In our algorithm, we first do sentence segmentation on relevant snippets, and pass each sentence through the MetaMap tool. The MetaMap tool identifies all biomedical entities contained in the statement and returns the preferred name and semantic type for every biomedical entity.

For every sentence, we create a set C containing the semantic types of all biomedical terms in the sentence. We also create a similar set S for the question text. Next, we find the intersection of set C of every sentence with the question set S and assign a score as the number of intersecting terms. We select the sentence with a maximum score and add it to the summary list. We also augment the question set S by doing the union of set C of the selected sentence with set S. We then use the new set S to find intersection with set C of remaining sentences. We repeat this procedure until we reach 200-word limit. Finally, to generate the summary, we concatenate the list of selected sentences to create the final answer.

In terms of tools, we used Stanford CoreNLP for snippet segmentation to get sentences. We use custom code using the Java API to MetaMap to get the concepts. MetaMap is also responsible for tokenizing, word sense disambiguation, connecting to UMLS and getting all the required mappings.

3.2 `factoid` and `list` type answers

For `factoid` type questions we are required to return a list of 5 entity names. The `list` type questions need to return a list of at most 100 entity names each of no more than 100 characters.

For answering the `factoid` type question, we use a similar technique as the `summary` generation pipeline, with additional scoring factors, and scoring at entity level, rather than sentence level. For each sentence, we get a list of biomedical entities using MetaMap. We score each entity using a scoring function that uses the entity frequency, the idf weight (the inverse document frequency of that entity), and the sentence similarity score found by the intersection of semantic set C of that sentence and question set S. Finally, we rank the list based on the scores and select top 5 entities as the answer for the `factoid` type question, and top 100 for the `list` type. We use the idf scores to eliminate common biomedical words or phrases. To get the idf score we downloaded and indexed Annual Baseline Medline repository, PubMed, using Lucene. We then use the Lucene indexes to get the term frequency and document count to calculate the idf score. For a multi-word entity, the idf score is the maximum of the idf scores of the individual tokenized words. This way a biomedical entity with even a single rare word will be ranked higher.

4 Results

We submitted results of our system for Phase B of task 6B. Phase B consisted of 5 test batches. We submitted our results for test batches 2,3 and 5 for `ideal` answers. We also evaluated our system on an older test batch sets using the BioASQ oracle. `ideal` answers are manually assessed using readability, repetition, recall, and precision and automatically by using ROUGE-2 and Rogue-SU4 scores. At the time of submission we did not have the manual scores; hence we only report the automatic scores. Table 1 and 5 shows our results on previous years dataset. As can be seen from Table 2 our system gave highest ROUGE-2 and ROUGE-SU4 scores among all systems on every test batch set.

Test Batch	ROUGE-2	ROUGE-SU4
Task 5B Batch 5	0.7188	0.7062
Task 5B Batch 4	0.7363	0.7258
Task 5B Batch 3	0.7802	0.7769
Task 5B Batch 2	0.6918	0.6903
Task 5B Batch 1	0.6716	0.6712
Task 4B Batch 5	0.7266	0.7250
Task 4B Batch 4	0.7196	0.7177
Task 4B Batch 3	0.6364	0.6527
Task 4B Batch 2	0.6777	0.6897
Task 4B Batch 1	0.6918	0.7024

Table 1: Results of `ideal` answers on task 4B and 5B test batch sets using BioASQ oracle. The results are arranged from most recent to least. The table shows ROUGE-2 and ROUGE-SU4 scores.

For BioASQ task 6B, we submitted `ideal` answers with summary created by selecting the only top sentence and with 200 word limit. The system that created the summary with 200 word limit gave highest ROUGE-2 and ROUGE-SU4 scores on every test batch set (2,3, and 5) that we submitted. Table 2 details this results.

Test Batch	Test Batch	ROUGE-2	ROUGE-SU4
UNCC System 1	Task 6B Batch 2	0.5833	0.6015
UNCC System 1	Task 6B Batch 3	0.6184	0.6290
UNCC System 2	Task 6B Batch 3	0.1973	0.1947
UNCC System 1	Task 6B Batch 5	0.7250	0.7122
UNCC System 2	Task 6B Batch 5	0.3846	0.3759

Table 2: Results of `ideal` answers on task 6B test batch sets using BioASQ oracle. The table shows ROUGE-2 and ROUGE-SU4 scores of UNCC System 1 and 2. UNCC System 1 submitted the summary created with 200 word limit. UNCC System 2 submitted summary created by selecting only top sentence.

For `factoid` and `list` type questions we did not submit results for task 6B, and we only report results from previous year's test batch sets using BioASQ oracle. Table 3 shows scores of `factoid` type question and Table 4 shows scores of `list` type questions.

Test Batch	Factoid SAcc	Factoid LAcc	Factoid MRR
Task 5B Test Batch 1	0.1200	0.1600	0.1333
Task 5B Test Batch 2	0.0323	0.1290	0.0575
Task 5B Test Batch 3	0.0385	0.0769	0.0462
Task 5B Test Batch 4	0.0303	0.0909	0.0455
Task 5B Test Batch 5	0.0571	0.1429	0.0786

Table 3: Result of `factoid` type question on task 5B of BioASQ. The scores include the Lenient accuracy LAcc, strict accuracy SAcc, and MRR(Mean Reciprocal Rank).

Test Batch	List Mean Precision	List Recall	List F-measure
Task 5B Test Batch 1	0.0241	0.3252	0.0441
Task 5B Test Batch 2	0.0353	0.2700	0.0600
Task 5B Test Batch 3	0.0195	0.3673	0.0367
Task 5B Test Batch 4	0.0250	0.2051	0.0389
Task 5B Test Batch 5	0.0391	0.2867	0.0630

Table 4: Result of `list` type question on task 5B of BioASQ. The results are evaluated on Mean Precision, Recall, and F-measure.

Test Batch	ROUGE-2	ROUGE-SU4
Task 3B Batch 5	0.5651	0.5672
Task 3B Batch 4	0.5848	0.5950
Task 3B Batch 3	0.5994	0.6128
Task 3B Batch 2	0.5451	0.5674
Task 3B Batch 1	0.5240	0.5368
Task 2B Batch 5	0.3967	0.4180
Task 2B Batch 4	0.4201	0.4458
Task 2B Batch 3	0.4731	0.4754
Task 2B Batch 2	0.4075	0.4258
Task 2B Batch 1	0.5313	0.5326
Task 1B Batch 2	0.3319	0.3596
Task 1B Batch 1	0.3032	0.3276

Table 5: Results of `ideal` answers on task 3B, 2B and 1B test batch sets using BioASQ oracle. The results are arranged from most recent to least. The table shows ROUGE-2 and ROUGE-SU4 scores.

5 Discussion

In this section we discuss the limitations of our work and address the reviewers comments not included in the revision of previous sections.

Our system was only evaluated by the ROUGE scores. However, high ROUGE results do not always imply good manual scores. In our approach we use the sentences of the passages without any changes. Thus, as observed by one of the reviewers, the manual score Repetition might be high; neither it is clear what is the impact of this approach on Readability. Furthermore, high ROUGE scores could be a side effect of a situation when the total number of words in all passages is less than 200.

Our evaluation was only done on ROUGE, because while building the system we only had access to old batch sets, and it was our first attempt to participate in Phase B of this competition. This said, our system tries to find the best candidate answers, and then concatenates them. So, further work needs to be done to convert the information from candidate snippets to a natural language answer that makes sense, and does not include any irrelevant information. We hope to address in the next year competition.

Regarding Repetition, our system first does sentence segmentation to get a list of snippets. Sometimes the snippets are overlapping and can have common sentences. Our system takes care of not repeating these sentences. What the system lacks is detecting sentences that are semantically similar and only consider one of them. Again – future work.

Regarding Readability, our concatenation is done such that each concatenated sentence is separated by period, hence usually making a coherent passage. Still the sentences might not follow a particular flow, and this might affect the readability score.

Another issue worth discussing is our approach to scoring the candidate answers. First, while scoring on the basis of term frequency is common, we use it (like other systems do), but we combine it with a `summary` pipeline score and the idf score. Second, we would have gone for machine learning techniques, but we felt we did not have enough labeled data.

One can argue that MetaMap doesn't always capture the all biomedical entities. However, we didn't face this problem. Although expanding candidate answers to include noun phrases could possibly improve the recall in generating candidate answers.

The OAQA system (Chandu et al., 2017) uses extractive summarization techniques like our system and the difference lies in sentence similarity. Our extractive summarization algorithm also shares similarity with Maximal Margin Relevance (MMR) in that both get sentence relevance score by comparing with question and other selected sentences. Our extractive summarization technique gives us higher ROUGE score than OAQA. Olelo system (Neves et al., 2017)and our system have similar pipeline in generation of summary and the only difference is in the way we do sentence similarity.

For `factoid` and `list` type questions our system does not perform well and the system can be improved by introducing better ranking algorithm, improved entity identification and filtering (at this time we use idf score to find out very common entities), and better relevance score between entity and the question.

6 Conclusion

In this paper, we showed our system's extractive summarization technique using lexical chains, or, more accurately, conceptual chains). We introduced an extractive summarization techniques for building paragraph-sized summaries. We have seen that use of the set of semantic type has proved very capable in ranking candidate answer sentences. Our system got highest ROUGE-2 and ROUGE-SU4 scores for `ideal` answers in all test batch sets.

We also showed a method to answer factoid and `list` type question. For these type of questions our system got low accuracy, which suggests that our algorithm needs to improve in ranking the entities. We plan to address these and other issues in future experiments.

Acknowledgment. We would like to thank the referees for their comments and suggestions. All the remaining faults are ours.

References

Diego Mollá Aliod. 2017. Macquarie university at bioasq 5b - query-based summarisation techniques for selecting the ideal answers. In (Cohen et al., 2017), pages 67–75.

Khyathi Chandu, Aakanksha Naik, Aditya Chandrasekar, Zi Yang, Niloy Gupta, and Eric Nyberg. 2017. Tackling biomedical text summarization: OAQA at bioasq 5b. In (Cohen et al., 2017), pages 58–66.

Kevin Bretonnel Cohen, Dina Demner-Fushman, Sophia Ananiadou, and Junichi Tsujii, editors. 2017. *BioNLP 2017, Vancouver, Canada, August 4, 2017*. Association for Computational Linguistics.

D.A. Ferrucci. 2012. Introduction to “this is watson”. 56:1:1–1:15.

Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2015. Ask me anything: Dynamic memory networks for natural language processing. *CoRR*, abs/1506.07285.

Mariana L. Neves, Fabian Eckert, Hendrik Folkerts, and Matthias Uflacker. 2017. Assessing the performance of olelo, a real-time biomedical question answering application. In (Cohen et al., 2017), pages 342–350.

Manabu Okumura and Takeo Honda. 1994. Word sense disambiguation and text segmentation based on lexical cohesion. In *15th International Conference on Computational Linguistics, COLING 1994, Kyoto, Japan, August 5-9, 1994*, pages 755–761.

Lawrence H. Reeve, Hyoil Han, and Ari D. Brooks. 2006. Biochain: lexical chaining methods for biomedical text summarization. In *Proceedings of the 2006 ACM Symposium on Applied Computing (SAC), Dijon, France, April 23-27, 2006*, pages 180–184. ACM.

Mourad Sarrouti and Said Ouatik El Alaoui. 2017. A biomedical question answering system in bioasq 2017. In (Cohen et al., 2017), pages 296–301.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Making neural QA as simple as possible but not simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pages 271–280. Association for Computational Linguistics.

Georg Wiese, Dirk Weissenborn, and Mariana L. Neves. 2017. Neural question answering at bioasq 5b. In (Cohen et al., 2017), pages 76–79.

Deyi Xiong, Yang Ding, Min Zhang, and Chew Lim Tan. 2013. Lexical chain based cohesion models for document-level statistical machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1563–1573. ACL.