Ontology-Based Retrieval & Neural Approaches for BioASQ Ideal Answer Generation

Ashwin Naresh Kumar,* Harini Kesavamoorthy,* Madhura Das,* Pramati Kalwad,*
Khyathi Raghavi Chandu, Teruko Mitamura and Eric Nyberg
Language Technologies Institute, Carnegie Mellon University
{anareshk,hkesavam,madhurad,pkalwad,kchandu,teruko,ehn}@cs.cmu.edu

Abstract

The ever-increasing magnitude of biomedical information sources makes it difficult and time-consuming for a human researcher to find the most relevant documents and pinpointed answers for a specific question or topic when using only a traditional search engine. Biomedical Question Answering systems automatically identify the most relevant documents and pinpointed answers, given an information need expressed as a natural language question. Generating a non-redundant, human-readable summary that satisfies the information need of a given biomedical question is the focus of the Ideal Answer Generation task, part of the BioASQ challenge. This paper presents a system for ideal answer generation (using ontology-based retrieval and a neural learning-to-rank approach, combined with extractive and abstractive summarization techniques) which achieved the highest ROUGE score of 0.659 on the BioASQ 5b batch 2 test.

1 Introduction

In this paper, we describe our attempts to address the Ideal Answer Generation task of the sixth edition of the BioASQ challenge,[1] which is a large-scale semantic indexing and question answering challenge in the biomedical domain. In particular, the sub-task of Phase B of this annual challenge is to develop a system for *query-oriented summarization*. Traditionally, there are two classes of summarization techniques, each having their own merits and pitfalls: (1) extractive and (2) abstractive. While extractive techniques patch relevant sentences together enabling them to generate grammatically robust summaries, they flounder on maintaining coherence and readability. On the contrary, abstractive techniques extract relevant information from the original text, which is then used to generate a novel natural language summary. While abstractive techniques are more succinct and coherent, automatic text generation is prone to grammatical error. This directly implies that extractive summarization techniques should perform well on automatic evaluation metrics (such as ROUGE), but do less well on human evaluation measures which account for precision, repetition and readability. We explore the hypothesis that a combination of these techniques will provide better overall performance on the ideal answer task, when compared with either approach used in isolation.

The dataset we use for development of the current work is released as a part of the sixth edition of the annual BioASQ challenge (Tsatsaronis et al., 2012). The main categories of answers in this data include *summary*, *factoid*, *list* and *yes/no*. There are a total of 2,251 questions, each of which is accompanied by a list of relevant documents and a list of relevant snippets extracted from each of these documents. Our model is an extension to the highest ROUGE scoring model in the final test batch of the fifth edition of the BioASQ challenge (Chandu et al., 2017), which is based on Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998). In addition, we attempted abstractive techniques that are scoped to improve the readability and coherence aspect of the problem. We made 4 submissions to the challenge.

The paper is organized as follows: Section 2 describes our overall system architecture and the implementation details. Experiments and results are discussed in Section 3 followed by conclusion and future work in 4.

2 System Architecture

The main components of the QA pipeline are outlined in Figure 1. As illustrated, the first step is pre-processing of the question to enrich it with

*denotes equal contribution

[1] bioasq.org

Proceedings of the 2018 EMNLP Workshop BioASQ: Large-scale Biomedical Semantic Indexing and Question Answering, pages 79–89
Brussels, Belgium, November 1st, 2018. ©2018 Association for Computational Linguistics

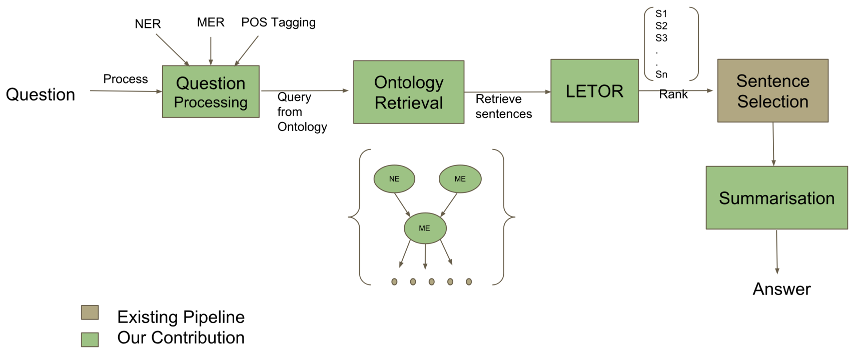


Figure 1: System Architecture

features derived from standard NLP techniques such as Part of Speech (POS) tagging, Named Entity Recognition (NER) and Medical Entity Recognition (MER). Subsequently an ontology-based retrieval system is used to retrieve relevant snippets for the question. The retrieved snippets are combined with the given BioASQ snippets for the question, and passed to the ranking module. The ranked snippets are then input to the sentence selection module from the existing OAQA pipeline (Chandu et al., 2017), which implements the CoreMMR (Zechner, 2002) and SoftMMR algorithms, which use similarity measures to select the most relevant and least redundant snippets. The selected sentences are then passed to the summarization module, which produces the final summary. Each of these modules is discussed in detail below.

2.1 Ontology-Based Information Retrieval

Although a large amount of biomedical text is available in resources such as NLM (NIH, 2018), it can be difficult to leverage in the absence of supervised or automatic labeling (annotation) of the unstructured text content. Our hypothesis is that an Ontology-based retrieval module which utilizes entity and relation extraction techniques to represent and compare the content of questions and candidate answers can improve the recall of answer-bearing documents from unstructured sources.

Our goal is to develop a graphical model that can represent the content of the question and each candidate answer. The nodes in the graph represent medical entities and the edges between them represent the relations between the entities. We extract relations from the text and index them into the graph based on previously-published work (Abacha and Zweigenbaum, 2015). The base architecture for the Ontology-based Retrieval module is shown in Figure 2.

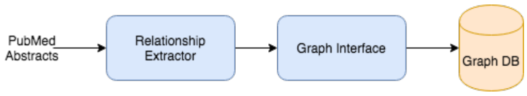


Figure 2: Graph Generation Pipeline

For every edge present in the graph, we store the ID of the source abstract, along with the ordinal index of the source sentence.

2.1.1 Relation Extraction

Relation Extraction (RE) is a technology used by an ontology-based retrieval system to capture the semantic relations which exist between the named entities mentioned in the text; both the entities and relations are considered instances of a given set of ontological types. In practice, various NLP toolkits are available for related tasks, such as dependency parsing, semantic role labeling, and subject-verb-object relation extraction.

However, common NLP tools aren't easily leveraged on biomedical text, due to dramatic differences in the structure and content of the sentences. There exist tools for relation extraction in sub-domains such as Bacteria (Duclos et al., 2007) and disease-cause ontologies (Schriml et al., 2011), but these methods heavily rely on the presence of specific words or features at the sentence level, and cannot be easily scaled to general biotext. Most neural methods for training relation extractors require a large ($O(10^6)$) corpus of labelled examples, which is not available for general biotext (Yih et al., 2015). In order to explore the use of ontology-based retrieval, we developed a novel

RE approach, which is described below. The base architecture for the RE module is depicted in Figure 3.

The following 4 steps are employed for extracting relations from a sentence:

1. Noun Phrase Chunking: The sentence is parsed using the TreeTagger POS tagger (Schmid, 1995) to obtain all the Noun Chunks that form the potential nodes of the graph. For our purpose, the nodes of the graph are all Medical and Named Entities. In order to perform this, the potential nodes are passed through a Medical Entity Recogniser (GRAM-CNN) (Zhu et al.) and the Stanford NER (Manning et al., 2014) discarding the chunks that are not recognized. For an example, let us consider the following sentence: *'Genomic microarrays have been used to assess DNA replication timing in a variety of eukaryotic organisms.'* which extract the following noun chunks: *'Genomic Microarrays'*, *'DNA Replication Timing'* and *'Eukaryotic Organisms'*.

2. Relation Extraction: This step comprises of 2 sub parts.

(2a) RE using Predicate Argument Structures: The Predicate Argument Structure (PAS) for the sentence, obtained using the Enju parser (Miyao et al., 2008), is further parsed in order to obtain possible relations for the graph. Possible relations are those that contain arguments related through a verb or a preposition.

(2b) RE transformation through transitivity: Transitivity is performed on relations obtained from the Enju parser in order to ensure that the arguments of the relations represent medical or named entities in the graph. The potential nodes are passed through the NER and MER. Nodes that are not tagged or recognized by either undergo a transitive transformation to give way to new relations. For the example mentioned, the following relations are formed post transitive formations: *'Microarrays assess Timing'*, *'Timing in Organisms'* and *'Microarrays in Organisms*.

3. Mapping to CUI: As the same medical entity can be represented in many forms, we employ a mapping to the Concept Unique Identifier (CUI) from the UMLS Metathesaurus (Bodenreider, 2004) using the python wrapper for MetaMap called pyMetamap (Aronson, 2001). For each of the Noun Chunks present, the UMLS Metathesaurus is queried to check if a CUI is present. If not, the individual CUIs are obtained for every word forming the noun chunk and the following rules are employed in order to form a hierarchical node structure. For the Noun chunks obtained in the example, CUIs are directly available for *'DNA Replication Timing'* and *'Eukaryotic Organism'* and not for *'Genomic Microarray'*. To build the tree structure for this node, the CUIs for *'Microarray'* and *'Genomic'* are individually obtained and since the latter is an adjective, the former becomes the child of the latter. The final CUIs obtained and the CUI node tree structure for *'Genomic Microarrays'* are depicted in Table 4 and Figure 6 respectively. For forming relations, the child nodes of all trees are used for connecting edges.

4. Relation Formation: As the arguments in the relations obtained through PAS are the base noun forms that do not represent the whole Noun Chunk, they are expanded to form the whole noun chunk. For the example, the relations obtained in 2b are expanded using the noun chunks to form the final relations as follows: *'Genomic Microarrays assess DNA Replication Timing'*, *'DNA Replication Timing in Eukaryotic Organisms'* and *'Genomic Microarrays in Eukaryotic Organisms'*. The entities in the relation are mapped to their CUI based representation to form a complete relation ready for insertion or retrieval from the graph. The mapped relations for the examples look as follows: *'C1709016, C0887950 assess C1257780'*, *'C1257780 in C0684063'* and *'C1709016, C0887950 in C0684063'*. Here, the root and children nodes are comma separated. The final graph structure for the relations is depicted in Figure 4.

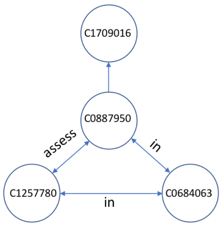


Figure 4: Graph obtained by relation extraction.

In order to index the graph, the relation extraction process specified above is utilized. The edges of the graph store additional information such as the abstract ID and the sentence offset in the abstract. In order to form a relation for a new query to retrieve information from the graph, a back-off mechanism is employed as follows:

1. Form relations using the process for indexing.
2. Query with all medical entities and obtain all

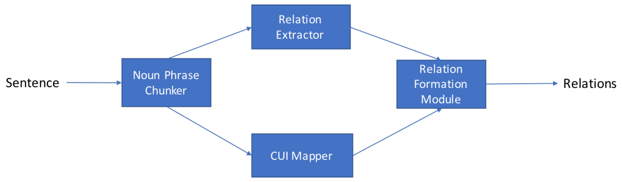


Figure 3: Architecture of Relation Extraction Module

abstracts between pairs of them.
3. Query the medical entities with the verb forms associated with them.
4. Query with just the medical entity and obtain all abstracts related to each of the medical entities.

2.1.2 Graph Creation

Graph Framework: All PubMed abstracts are tokenized and relations are extracted from them. These are added as relations in the graph. We create custom data structures for the Nodes and Edges(Relations) in Neo4j (Webber, 2012). Every relation has attributes which are comma separated values of PubMed ID, location within the abstract. This is stored in order to retrieve the exact sentence that was used to create a particular relation. We hypothesize that this can improve in getting relevant snippets across the abstracts.
Phase I: Ontology Creation with UMLS Concepts. Part of speech tagging is the most intuitive way of approaching the problem of extracting the relations from a given text. An initial strategy of forming the Subject Verb Object (SVO) triplets was formed based on a left-right parsing of the text. For this purpose, an off-the-shelf POS tagger (Schmid, 1994) was used. This is not an effective method to create a graph as it misses very important clauses and fails to recognize the Noun Chunks in sentences. In order to overcome the limitations of this, we form a UMLS based mapping of the noun phrases to medical concept thesaurus as in the UMLS Meta Thesaurus. Once the CUI ids are mapped using metamap as in Section 2.1.1, these are prepared to be indexed into the graph. These relations are added into the graph by following standard database methods, i.e.,

- If the relation is already present in the graph, then the relation attribute is appended with the PubMed ID, and offset in the abstract.
- Every node in the relation is first queried from the graph and then the relation is added between the nodes retrieved. If no node is retrieved, then new nodes are created and the relation is created.

Phase II: Adding transitivity to Relations. The limitations of Phase I was that despite accurate relations from the relation extractor, the mapping in the graph for different clauses joined together with different prepositions was not solved. We solve it with the following method. First, a key value pair is added as an attribute to the relation, where every key is a preposition and the value is the NodeID noun chunk that is associated with the preposition. For example, in the sentence "Genomic microarrays have been used to assess DNA replication timing in a variety of eukaryotic organisms", the clause "in a variety of eukaryotic organisms" would be missed in the phase II of ontology creation. But in Phase II, we convert such that the verb "assess" has an attribute "{in, $node_{xyz}$}" where $node_{xyz}$ is the node pertaining to the CUI of "eukaryotic organisms".

2.2 Ranking

Information Retrieval is one of the essential components of a Question Answering pipeline. It will help provide relevant information to the pipeline for more accurate answers. Ranking snippets based on relevance to the question will improve the answer selection process and in turn give more relevant answers. Employing Learning to Rank (LETOR)(Qin et al., 2010) methods to rank snippets should help rank snippets according to the questions.
The output of the Ontology based Information Retrieval is a set of relevant snippets. We combine the given BioASQ snippets along with the Ontology Retrieved snippets and rank them according to relevance to the question. The ranking algorithm finally gives a set of ranked snippets relevant to the question. There is a possibility that the Ontology based retrieved snippets may also have irrelevant snippets. To prevent the error from further propagating into the pipeline, we use a simple BM25(Robertson and Zaragoza, 2009) scoring threshold between the snippets and the question.

We discard the snippets which have a BM25 score lower than a certain threshold. LETOR is highly feature driven which necessitates a good amount of feature engineering. The explored features are listed in the next section. In this paper, we have explored 2 LETOR approaches: a) RankSVM and b) A Listwise Neural Approach.

2.2.1 Feature Engineering

Multiple features have been explored as inputs to the LETOR framework. The features can be divided into 3 major categories i.e. 1) Statistical Features 2) Semantic Features and 3) Syntactic Features. The statistical features included length of snippets, BM25 score between query and snippet, dot product of tf-idf between query and snippet, cosine similarity over the TF-IDF vectors (along with log space representation), number of bi-grams in the intersection of query and snippets and Jaccard similarity score. The semantic features include averaged word2vec representations across snippets. The syntactic features are number of medical entities and bag of words representation of medical entities.

2.2.2 Quasi Ground Truth Creation

An initial challenge while formulating the LETOR framework is the ground truth ranking of the snippets as that was not provided in the BioASQ training data. The primary purpose of the LETOR model was to rank more relevant snippets higher up to obtain a higher ROUGE score. Taking this into account, we decided to create the ground truth as the BM25 scores between the snippet and the ideal answer. The scores were calculated according to the formula in 1. The snippets were ranked according to the scores for each question.

$$\begin{aligned} &RelevanceScore \\ &\quad = score_{BM25}(idealanswer, snippet) \end{aligned} \tag{1}$$

2.2.3 RankSVM

RankSVM(Cao et al., 2006) is a pairwise LETOR approach towards ranking of documents. Each pair of snippets was taken for a question and was labeled as -1 if the second snippet was ranked higher and +1 if the second snippet was ranked lower. In a pairwise approach there is an overhead of maintaining the metadata as we need to know which set of snippets are going into the SVM as input for validation of the model. Consider $F(Q, S_1)$ as a feature representation of the question Q and snippet S_1. Similarly, $F(Q, S_2)$ is a feature representation of the question Q and snippet S_2. $F(Q, S_1)$ and $F(Q, S_2)$ are inputs to the SVM and the SVM predicts a -1 or +1 according to the relative ranking of S_1 and S_2.

2.2.4 Neural Ranking Approach

The second approach that we implemented was a list-wise ranking approach inspired from ListNet(Cao et al., 2007). Every data point is a feature representation between the question Q and n^{th} snippet S_n. The neural network is trained against the BM25 scores between the snippet and the ideal answer. The architecture of the network is a 2 layer MLP with ReLU activations. The final layer is a linear layer of size 1.

In an ideal scenario, where we would have had the ground truth rankings of the snippets, it would be intuitive to use a probabilistic loss. In our case, as we are using proxy golden ranks with the BM25 score, it would be more intuitive for the model to learn to estimate the scores instead of the relative ranking of the snippets. Hence, we use a RMSE loss as we want our model to estimate the BM25 scores. The RMSE loss is calculated per question as we would want to learn the distribution of snippets with respect to a single question and not across the complete dataset. The final ranking of the snippets are determined with respect to the scores the model predicts.

2.3 Summarization

Summarization is the final stage in the question answering pipeline. The ranked snippet sentences feed into the summarization module which finally outputs the ideal answers.

For the case of ideal answer generation, two types of summarization techniques can be employed; extractive and abstractive summarization. Extractive summarization works by selecting the most relevant sentences in a document to generate the summary (Allahyari et al., 2017). The summaries generated using this technique generally obtain high ROUGE scores (Lin, 2004) due to the high n-gram overlap between the generated summary and the ideal answer. Abstractive summarization on the other hand works by generating the summary word by word as opposed to picking sentences in the case of extractive summarization. Recent advances in abstractive summarization using Pointer Generator Coverage (PGC) networks (See et al., 2017) have shown that neural sequence to sequence models can generate abstractive summaries which are readable and have high ROUGE scores. Given that these networks can generate human readable summaries with high ROUGE scores, we decided to use this model as the ideal answer generation module in our question answering pipeline. We show that the neural sequence model is able to generate ideal answers in the biomedical domain. More concretely, we see that a pretrained PGC model can be transferred to the biomedical domain to generate ideal answers i.e., the model is able to handle new words (mainly biomedical words) and not generate any unknown tags (<UNK>) in the summaries which would hinder the readability. We also show that fine tuning the model on the BioASQ data generates better answers in terms of the ROUGE scores.

3 Experiments and Results

This section describes the experiments conducted to evaluate each of the components in the question answering pipeline. All the experiments have been conducted on batch 2 of the fifth edition of BioASQ and evaluated using the official *Oracle*[2] developed by the organizers of the task.

3.1 Ontology based Information Retrieval

Ontology based retrieval has been evaluated for providing summary answers to queries with zero snippets provided. For example, the snippets retrieved for the question, *'Does metformin interfere thyroxine absorption?'* has a ROUGE of 0.2044 compared with the ideal answer provided for it.

3.2 Ranking

3.2.1 RankSVM Feature Analysis

Ablation studies were carried out with respect to the features to determine which set of features give us the best results.

It is seen that the statistical features contributed the most to the model. Another interesting observation from the graph is that even though BM25 and log(BM25) were the top contributing features, the log(BM25) has a higher weight. This is mainly because the log scale is known to be more stable and therefore will help the model learn better.

3.2.2 Neural Ranking Approach Analysis

The Neural approach was evaluated against the RankSVM results. We also added the syntactic features and did a comparison study on them. From Table 1 it is seen that adding the syntactic features have contributed to an overall increase in the ROUGE scores for both the models. Also, it is noticed that the Neural model has performed better than the RankSVM. This is mostly due to the fact that the Neural approach is trying to estimate the BM25 scores between the snippet and the ideal answer rather than trying to mimic the quasi ranking. From the discussion of the results above, we can confirm the hypothesis that ranking snippets in an order of relevance will help improve the quality of answers generated by the pipeline.

3.3 Summarization

Ranked snippet sentences from the ranking pipeline are fed into the summarization module. The following experiments were carried out:

1. Using the PGC network pretrained on CNN/Daily Mail to generate the ideal answers

2. Fine tuning the pretrained PGC network on BioASQ data

Table 1 gives the ROUGE scores obtained by both the models on the BioASQ dataset. For the model fine tuning, the pretrained model is further trained on BioASQ 5b training data. We see that the fine tuned model obtains much higher ROUGE-2 and ROUGE-SU4 scores when compared to the pretrained model. This shows that the fine tuned model generates better answers than the pretrained model in terms of ROUGE score. On closely analyzing the answers generated by the PGC models, we see that there are no <UNK>s generated by both the pretrained and fine tuned models. The model is also able to effectively copy the unknown words from the biomedical source text. A detailed error analysis for the answers generated by the model is discussed in the upcoming subsection 3.3.1.

3.3.1 Error Analysis

This subsection discusses the analysis on the ideal answers generated by both the pretrained model and fine tuned model for different question types in the BioASQ dataset (Table 2). All the readability judgments made in this subsection are an indicator of the subjective judgments made by the authors of this paper:

1. **Yes/No type:**

[2] BioASQ Oracle

RankSVM LETOR Framework	**ROUGE-2**	**ROUGE-SU4**
BM25, Cosine Similarity, TF-IDF dot product	0.6442	0.6453
BM25, Cosine Similarity, TF-IDF dot product, word2vec dot product	0.6439	0.6460
BM25, Cosine Similarity, TF-IDF dot product, word2vec dot product, Snippet Length, log(sum(TF-IDF))	**0.6471**	**0.6481**
Neural LETOR Framework		
Neural Approach	0.6565	0.6552
Neural Approach + MER Bag-of-words	**0.6590**	**0.6581**
Abstractive Summarization		
Pretrained PGC	0.29	0.3
Fine tuned PGC	**0.4640**	**0.4659**

Table 1: Results on 5b batch 2

Question	**Ideal Answer**	**Pre-trained model answer**	**Fine tuned model answer**
Do proton pump inhibitors affect thyroxine absorption? (Yes-No type)	***Proton-pump inhibitors, antacids and a long list of drugs may decrease thyroxine absorption***. Patients with hypothyroidism and normal TSH values during LT4 replacement therapy may need additional thyroid function testing after treatment with PPIs and may need adjustment of their LT4 dose.	***Ppis and a long list of drugs may decrease thyroxine absorption*** . Many commonly used drugs , such as bile acid sequestrants , ferrous , phosphate binders , raloxifene and ***proton-pump inhibitors , have also been shown to interfere with the absorption of levothyroxine*** . Patients with hypothyroidism and normal tsh values during short-term treatment .	***Antacids inhibitors , antacids and a long list of drugs may decrease thyroxine absorption*** . Many commonly used drugs , such as bile acid sequestrants , ferrous sulphate , calcium carbonate , aluminium-containing antacids , phosphate binders , raloxifene and ***proton-pump inhibitors , have also been shown to interfere with the absorption of levothyroxine*** . Pantoprazole did not influence endocrine function in healthy male volunteers during short-term treatment .
What is the ubiquitin proteome? (Factoid Type)	The ***ubiquitin proteome is the entire set ubiquitinated proteins and of their respective ubiquitination sites***.	Tandem repeated ubiquitin binding entities tubes under non-denaturing conditions followed by mass spectrometry analysis to study global ubiquitylation events that may lead to the identification of potential drug targets . To study the ubiquitin proteome we have established an immunoaffinity purification method for the proteomic analysis of endogenously ubiquitinated protein complexes .	Mass spectrometry now allows high throughput approaches for the identification of the thousands of ubiquitinated proteins and of their ubiquitination sites . We used tandem repeated ubiquitin binding entities tubes under non-denaturing conditions followed by mass spectrometry analysis to study global ubiquitylation events that may lead to the identification of potential drug targets .
What memory problems are reported in the "Gulf war syndrome"? (List Type)	***Loss of memory and dysmnesia are memory problems*** reported in the Gulf war syndrome. Patients suffering from this syndrome often have ***other nonspecific symptoms such as fatigue, skin rash, headache, muscle and joint pain and sexual dysfunction***.	The strongest associations were for ***mood swings or 20.9***. The symptoms include ***fincapacitating fatigue , musculoskeletel and joint pains , headaches , muscle and joint pain*** . His history revealed several front-line operations in 1990 and 1991 during the gulf war .	The gulf war syndrome gws is the beginning of what was to be identified as the gulf war syndrome . In early 1992 , u.s. troops returning from the gulf war began reporting a variety of nonspecific symptoms such as ***fatigue , skin rash , muscle and joint pain , and loss of memory*** .
What is the treatment of acute pericarditis? (Summary Type)	A multidisciplinary approach is frequently necessary to treat acute pericarditis; the most frequent treatments are: ***antiinflammatory steroid and non-steroid drugs, antibiotic therapy, pericardial drainage and, less frequently ,intrapericardial irrigation of fibrinolytics; antituberculous chemotherapy in presence of Tuberculous Agent***	The incidence of purulent pericarditis has decreased since the antibiotic era . it is typically an acute and potentially lethal disease , necessitating rapid diagnosis and adequate therapy to improve prognosis . ***standard treatment combines appropriate antibiotic therapy with surgical drainage.***	Purulent pericarditis has decreased considerably since the antibiotic era . It is typically an acute and potentially lethal disease , necessitating rapid diagnosis and adequate therapy to improve prognosis . ***Standard treatment combines appropriate antibiotic therapy with surgical drainage*** .
What causes erucism? (Summary Type)	Erucism is defined as urtication by Lepidoptera larvae. It is a skin reaction to envenomation from certain poisonous caterpillar bristles. The hair on the dorsum of the last instar larvae of the moth may cause urticarial reactions (erucism) as well as eye problems and temporary blindness.	Tropical insects carry infectious diseases , cutaneous injury can occur by other mechanisms . the hair on the dorsum of the moth may cause urticarial reactions . ***erucism is defined as urtication by lepidoptera larvae . erucism is defined as urtication by lepidoptera larvae*** .	Tropical insects carry infectious diseases , cutaneous injury can occur by other mechanisms . The hair on the dorsum of the last instar larvae of the moth may cause urticarial reactions . ***Urticating is defined as urtication by lepidoptera larvae . erucism is defined as urtication by lepidoptera larvae*** .

Table 2: Examples of error types observed in the qualitative analysis

Here, we see that the model generated answers address the question and also gives out extra facts not described in the ideal answer, but pertaining to the question. We see that the answer given by the fine tuned model seems more complete than that of the pretrained model as it mentions that antacids (which contain PPI) decrease thyroxine absorption and also that they interfere with a specific type of thyroxine, namely levothyroxine.

2. **Factoid type:**

   Here, the generated summaries miss the answer as exact answer is not present even in the snippets. Fine tuned model on the other hand, generates an answer more readable than the pretrained model generated answer.

   For most of the other factoid questions, we saw that the question was answered correctly, but the answer describes the facts very differently and also gives different extra facts compared to the ideal answer. Another observation was that presence of several acronyms and abbreviations reduced the ROUGE score.

3. **List type:** Here, we see that the answer generated by the fine tuned model is more readable as there is a seamless flow in the answer where the answer starts off by explaining what the Gulf War Syndrome is and later goes on to list the problems reported in the Gulf War Syndrome. We also notice that the pretrained model misses the symptom 'loss of memory' mentioned in the ideal answer which is however picked up by the fine tuned model.

4. **Summary type:** Here, the generated answers partially answer the question as both

Test Batch	System	ROUGE 2	ROUGE SU4
Batch 1	RankSVM	0.6372	0.6456
Batch 3	Neural Ranking + Extractive Summarization	**0.5743**	**0.5883**
Batch 4	Neural Ranking + Abstractive Summarization	0.4183	0.4281
Batch 5	Relation extraction + Ontology + Neural Ranking + Abstractive Summarization	0.4573	0.4637

Table 3: BioASQ Results for Task 6B

of them mention the surgical drainage treatment which is a super set of the pericardial drainage treatment. Other treatment types are however missed by both the models in their answers. This is mainly due to the fact that there is no direct mapping between the ideal answer and the snippets.

5. **Summary type:** Another error which occasionally surfaces is repetition. In the pre-trained model answer, we see that the last sentence is repeated. In the fine tuned model however, there is no repetition with respect to the entire sentence. Although, a majority of the last sentence is repeated in the fine tuned model answer, we see that the term 'urtication' is defined in terms of its own verb form and urtication is later used to define erucism.

Table 3 comprises our results over the test batches of the sixth edition. We believe the model gave the best ROUGE scores for test batch 3 as we have seen from the previous section that the neural model along with domain specific features performed the best among all the models.

4 Conclusion and Future Work

This paper discusses our system for summary type answer generation using a knowledge graph and a neural learning to rank approach. The ranked snippets are further used to generate the answers using extractive and abstractive summarization techniques. We also show that we can transfer the abstractive summarization knowledge from the CNN/Daily-Mail summarization task to the task of biomedical summarization.

From a brief manual inspection of the generated summaries and their relevant documents, we believe that from an NLP standpoint the following are some of the promising directions to explore. Anaphora resolution would help provide better relations. We also plan to use the ontology indexing and retrieval system for factoid and list types of questions. Incorporating the question type as contextual information while generating summaries could lead to improving precision. Instead of a dual step of transfer learning with training and fine tuning on PGC network, the abstracts of the PUBMED articles and the entire document could potentially be leveraged to train the end to end network.

As an extension, we intend to pursue the following tasks for BioASQ:

- The current pipeline of the work includes the MMR algorithm while selecting sentences. Experimentation with other diversification algorithms like xQuAD(Santos et al., 2010) and PM-2(Dang and Croft, 2012) can be used for sentence selection.
- Exploration of more language model based features in the LETOR pipeline like Pointwise Mutual Information.

References

Asma Ben Abacha and Pierre Zweigenbaum. 2015. Means: A medical question-answering system combining nlp techniques and semantic web technologies. *Information processing & management*, 51(5):570–594.

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. Text summarization techniques: A brief survey. *arXiv preprint arXiv:1707.02268*.

Alan R Aronson. 2001. Effective mapping of biomedical text to the umls metathesaurus: the metamap program. In *Proceedings of the AMIA Symposium*, page 17. American Medical Informatics Association.

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon. 2006. Adapting ranking svm to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193. ACM.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136. ACM.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM.

Khyathi Chandu, Aakanksha Naik, Aditya Chandrasekar, Zi Yang, Niloy Gupta, and Eric Nyberg. 2017. Tackling biomedical text summarization: Oaqa at bioasq 5b. *BioNLP 2017*, pages 58–66.

Van Dang and W. Bruce Croft. 2012. Diversity by proportionality: An election-based approach to search result diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 65–74, New York, NY, USA. ACM.

Catherine Duclos, Jérome Nobécourt, Gian Luigi Cartolano, Anis Ellini, and Alain Venot. 2007. An ontology of bacteria to help physicians to compare antibacterial spectra. In *AMIA Annual Symposium Proceedings*, volume 2007, page 196. American Medical Informatics Association.

Tadayoshi Hara, Yusuke Miyao, and Jun-ichi Tsujii. 2010. Evaluating the impact of re-training a lexical disambiguation model on domain adaptation of an hpsg parser. In *Trends in Parsing Technology*, pages 257–275. Springer.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Yusuke Miyao, Rune Sætre, Kenji Sagae, Takuya Matsuzaki, and Jun'ichi Tsujii. 2008. Task-oriented evaluation of syntactic parsers and their representations. *Proceedings of ACL-08: HLT*, pages 46–54.

NIH. 2018. *National Library of Medicine*.

Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. 2010. Letor: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13(4):346–374.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Rodrygo LT Santos, Jie Peng, Craig Macdonald, and Iadh Ounis. 2010. Explicit search result diversification through sub-queries. In *European Conference on Information Retrieval*, pages 87–99. Springer.

Helmut Schmid. 1994. Probabilistic pos tagging using decision trees. In *Proceedings of International Conference on New methods in Language Processing*.

Helmut Schmid. 1995. Treetaggerl a language independent part-of-speech tagger. *Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart*, 43:28.

Lynn Marie Schriml, Cesar Arze, Suvarna Nadendla, Yu-Wei Wayne Chang, Mark Mazaitis, Victor Felix, Gang Feng, and Warren Alden Kibbe. 2011. Disease ontology: a backbone for disease semantic integration. *Nucleic acids research*, 40(D1):D940–D946.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

George Tsatsaronis, Michael Schroeder, Georgios Paliouras, Yannis Almirantis, Ion Androutsopoulos, Eric Gaussier, Patrick Gallinari, Thierry Artieres, Michael R Alvers, Matthias Zschunke, et al. 2012. Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. In *AAAI fall symposium: Information retrieval and knowledge discovery in biomedical text*.

Jim Webber. 2012. A programmatic introduction to neo4j. In *Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity*, pages 217–218. ACM.

Scott Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base.

Klaus Zechner. 2002. Automatic summarization of open-domain multiparty dialogues in diverse genres. *Computational Linguistics*, 28(4):447–485.

Qile Zhu, Xiaolin Li, and Ana Conesa. Gram-cnn: a deep learning approach with local context for named entity recognition in biomedical text. *Bioinformatics*, page btx815.

Supplementary Material

For the example provided in section 2.1.1, the Predicate Argument Structure provided by Enju is shown in Figure 5, which forms the following relations: *'Microarrays assess Timing'*, *'Assess in variety'* and *'variety of organisms'*.

Table 4: CUI Mapping

Concept	CUI
Microarray	C1709016
Genomic	C0887950
DNA Replication Timing	C1257780
Eukaryotic Organism	C0684063

Figure 6 shows the node structure that is built from the CUIs for Genomic Microarray.

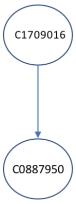


Figure 6: CUI node structure for Genomic Microarray

The efficacy of the Relation Extraction module depends on the tools it utilizes. The Enju parser trained using the GENIA corpus has an F-score of 90.15 on the same (Hara et al., 2010). GRAM-CNN has an F1-score of 87.26% on the Biocreative II dataset, 87.26% on the NCBI dataset and 72.57% on the JNLPBA dataset (Zhu et al.). In addition to the hierarchical node structure, these can also lead to the existence of incorrect nodes and edges in the ontology. Every node is associated with all abstracts containing a mention of them. This results in the possibility of the retrieval logic returning all abstracts containing just a mention of the medical/named entity present in a query, rather than only the relevant abstracts for that particular query. The Ranking module filters these abstracts to obtain those most relevant to the query.

We also graphed out the top contributing features for our RankSVM model. Figure 7 displays the contribution of the top 6 features which were used in the model. The factors by which the top 6 features contributed were then normalized and plotted. The top contributing features are depicted in Figure 7.

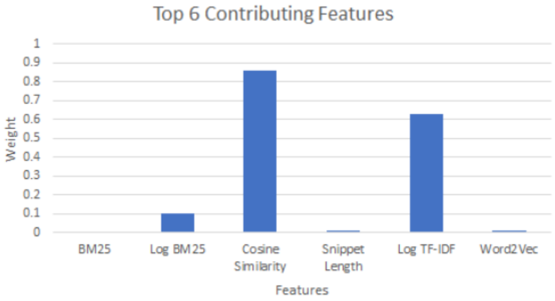
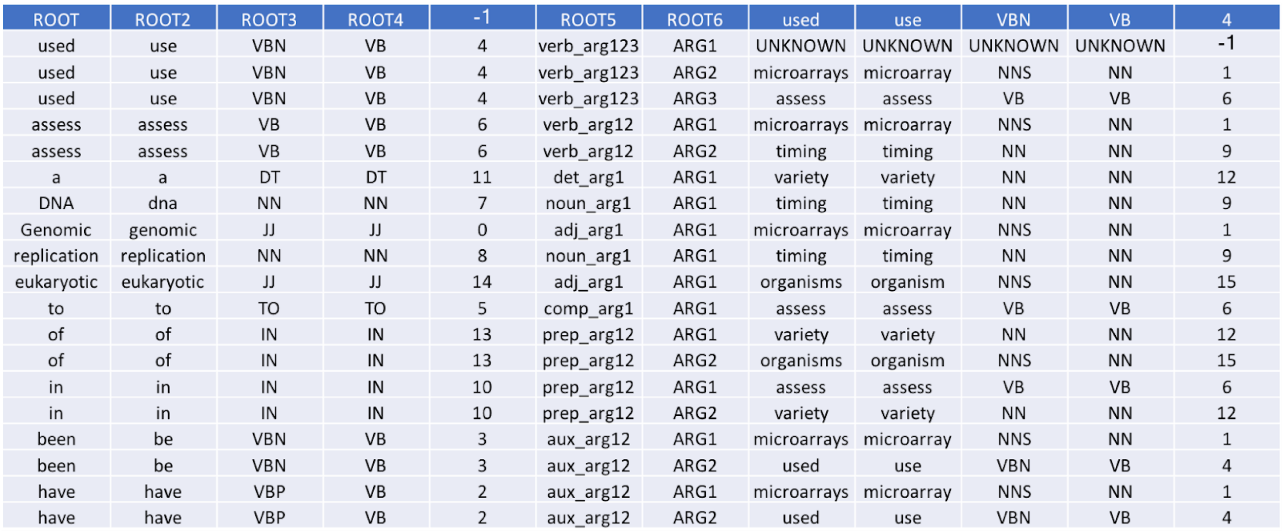


Figure 7: RankSVM Top Contributing Features



ROOT	ROOT2	ROOT3	ROOT4	-1	ROOT5	ROOT6	used	use	VBN	VB	4
used	use	VBN	VB	4	verb_arg123	ARG1	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	-1
used	use	VBN	VB	4	verb_arg123	ARG2	microarrays	microarray	NNS	NN	1
used	use	VBN	VB	4	verb_arg123	ARG3	assess	assess	VB	VB	6
assess	assess	VB	VB	6	verb_arg12	ARG1	microarrays	microarray	NNS	NN	1
assess	assess	VB	VB	6	verb_arg12	ARG2	timing	timing	NN	NN	9
a	a	DT	DT	11	det_arg1	ARG1	variety	variety	NN	NN	12
DNA	dna	NN	NN	7	noun_arg1	ARG1	timing	timing	NN	NN	9
Genomic	genomic	JJ	JJ	0	adj_arg1	ARG1	microarrays	microarray	NNS	NN	1
replication	replication	NN	NN	8	noun_arg1	ARG1	timing	timing	NN	NN	9
eukaryotic	eukaryotic	JJ	JJ	14	adj_arg1	ARG1	organisms	organism	NNS	NN	15
to	to	TO	TO	5	comp_arg1	ARG1	assess	assess	VB	VB	6
of	of	IN	IN	13	prep_arg12	ARG1	variety	variety	NN	NN	12
of	of	IN	IN	13	prep_arg12	ARG2	organisms	organism	NNS	NN	15
in	in	IN	IN	10	prep_arg12	ARG1	assess	assess	VB	VB	6
in	in	IN	IN	10	prep_arg12	ARG2	variety	variety	NN	NN	12
been	be	VBN	VB	3	aux_arg12	ARG1	microarrays	microarray	NNS	NN	1
been	be	VBN	VB	3	aux_arg12	ARG2	used	use	VBN	VB	4
have	have	VBP	VB	2	aux_arg12	ARG1	microarrays	microarray	NNS	NN	1
have	have	VBP	VB	2	aux_arg12	ARG2	used	use	VBN	VB	4

Figure 5: Predicate Argument Structure