BioAMA: Towards an End to End BioMedical Question Answering System

Vasu Sharma*, Nitish Kulkarni*, Srividya Pranavi Potharaju*,
Gabriel Bayomi*, Eric Nyberg, Teruko Mitamura
Language Technologies Institute
School Of Computer Science
Carnegie Mellon University
[vasus, nitishkk, spothara, gbk, ehn, teruko] @cs.cmu.edu

Abstract

In this paper, we present a novel Biomedical Question Answering system, *BioAMA*: "Biomedical Ask Me Anything" on task 5b of the annual BioASQ challenge (Balikas et al., 2015). We focus on a wide variety of question types including factoid, list based, summary and yes/no type questions that generate both exact and well-formed 'ideal' answers. For summary-type questions, we combine effective IR-based techniques for retrieval and diversification of relevant snippets for a question to create an end-to-end system which achieves a ROUGE-2 score of 0.72 and a ROUGE-SU4 score of 0.71 on ideal answer questions (7% improvement over the previous best model). Additionally, we propose a novel Natural Language Inference (NLI) based framework to answer the yes/no questions. To train the NLI model, we also devise a transfer-learning technique by cross-domain projection of word embeddings. Finally, we present a two-stage approach to address the factoid and list type questions by first generating a candidate set using NER taggers and ranking them using both supervised and unsupervised techniques.

1 Introduction

In the era of ever advancing medical sciences and the age of the internet, a remarkable amount of medical literature is constantly being posted online. This has led to a need for an effective retrieval and indexing system which can allow us to extract meaningful information from these vast knowledge sources. One of the most effective and natural ways to leverage this huge amount of data in real life is to build a Question Answering (QA) system which will allow us to directly query this data and extract meaningful and structured information in a human readable form.

Our key novel contributions are as follows:

1. We achieve state of the art results in automatic evaluation measures for the ideal answer questions in Task 5b of the BioASQ dataset, yielding a 7% improvement over the previous state of the art system (Chandu et al., 2017).

2. We introduce a novel NLI-based approach for answering the yes/no style questions in the BioASQ dataset. We model this as a Textual Entailment (TE) problem and use Hierarchical Convolutional Neural Network based Infersent models (Conneau et al., 2017) to answer the question. To address the challenge of inadequate training data, we also introduce a novel embedding projection technique which allows for effective transfer learning from models trained on larger datasets with a different vocabulary to work well on the much smaller BioASQ dataset.

3. We present two-stage approach to answer factoid and list type questions. By using an ensemble of biomedical NER taggers to generate a candidate answer set, we devise unsupervised and supervised ranking algorithms to generate the final predictions.

4. We improve upon the MMR framework for relevant sentence selection from the chosen snippets that was introduced in the work of Chandu et al. (2017). We experiment with a number of more informative similarity metrics to replace and improve upon the baseline Jaccard similarity metric.

Proceedings of the BioNLP 2018 workshop, pages 109–117
Melbourne, Australia, July 19, 2018.

2 Relevant Literature

Biomedical Question answering has always been a hot topic of research among the QA community at large due to the relative significance of the problem and the challenge of dealing with a non standard vocabulary and vast knowledge sources. The BioASQ challenge has seen large scale participation from research groups across the world. One of the most prominent among such works is from Chandu et al. (2017) who experiment with different biomedical ontologies, agglomerative clustering, Maximum Marginal Relevance (MMR) and sentence compression. However, they only address the ideal answer generation with their model. Peng et al. (2015) in their BioASQ submission use a 3 step pipeline for generating the exact answers for the various question types. The first step is question analysis where they subdivide each question type into finer categories and classify each question into these subcategories using a rule based system. They then perform candidate answer generation using POS taggers and use a word frequency-based approach to rank the candidate entities. Wiese et al. (2017) propose a neural QA based approach to answer the factoid and list type questions where they use FastQA: a machine comprehension based model (Weissenborn et al., 2017) and pre-train it on the SquaD dataset (Rajpurkar et al., 2016) and then finetune it on the BioASQ dataset. They report state of the art results on the Factoid and List type questions on the BioASQ dataset. Another prominent work is from Sarrouti and Alaoui (2017) who handle the generation of the exact answer type questions. They use a sentiment analysis based approach to answer the yes/no type questions making use of SentiWordNet for the same. For the factoid and list type questions they use UMLS metathesaurus and term frequency metric for extracting the exact answers.

3 The BioASQ challenge

BioASQ challenge (Balikas et al., 2015) is a large scale biomedical question answering and semantic indexing challenge, which has been running as an annual competition since 2013. We deal with the Phase B of the challenge which deals with large scale biomedical question answering. The dataset provides a set of questions and snippets from PubMed, which are relevant to the specific question. It also provides users with a question type and urls of the relevant PubMed articles itself. The 5b version of this dataset consists of 1,799 questions in 3 distinct categories:

1. **Factoid type**: This question type has a single entity as the ground truth answer and expects the systems to output a set of entities ordered by relevance; systems are evaluated using the mean reciprocal rank (Radev et al., 2003) of the answer entities with reference to the ground truth answer entity.

2. **List type**: This answer type expects the system to return an unordered list of entities as answer and evaluates them using a F-score based metric against a list of reference answer entities which can vary in number.

3. **Yes/No type**: This question type asks the systems to answer a given question with a binary output namely yes or no. The questions typically require reasoning and inference over the evidence snippets to be able to answer the questions correctly.

The dataset expected the participants to generate two types of answers, namely, exact and ideal answers. In ideal answers, the systems are expected to generate a well formed paragraph for each of the question types which explains the answer to the question. They call these answers 'ideal' because it is what a human would expect as an answer by a peer biomedical scientist. In the exact answers the systems are expected to generate "yes" or "no" in the case of yes/no questions, named entities in the case of factoid questions and list of named entities in the case of list questions.

4 Ideal Answers

This section describes our efforts to address the ideal answer category on BioASQ.

Our pipeline for ideal answers has three stages. The first stage involves pre-processing of answer snippets and ranking of answer sentences by various retrieval models described in the following sections. The retrieval model scores form the soft positional component introduced in the MMR algorithm. We perform sentence selection next, where we select the top 10 best sentences for generating an ideal answer. The third and final stage involves tiling together the selected sentences to generate a coherent, non redundant, ideal answer for the given question as mentioned in (Chandu et al., 2017). The subsequent subsections explain

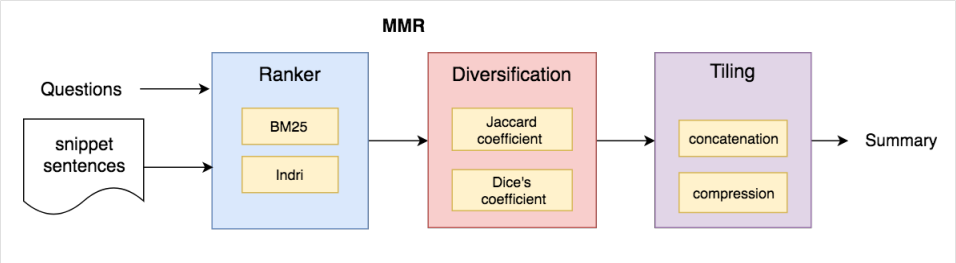

Figure 1: Pipeline for ideal answer generation

the pipeline for ideal answer type questions in detail (see Figure 1).

4.1 Question-Sentence Retrieval

In this section we describe various approaches which were adapted to improve the initial retrieval of candidate sentences. We used the standard BM25 algorithm with custom pre-processing of excluding medical entities from stop word removal.

4.1.1 Indri

Indri (Strohman et al., 2005) is a retrieval model based on the use of statistical language models and query likelihood. We employed a two-stage smoothing that considers characteristics of both the question and answer sentences.

The Indri score for a candidate sentence is estimated in a collection (C) of snippets as follows:

$$p(q_i|d) = (1-\lambda)p_{mle}(q_i|d) + \lambda p_{mle}(q_i|C) \quad (1)$$

$$p_{mle}(q_i|d) = \frac{tf + \mu p_{mle}(q_i|C)}{length(d) + C} \quad (2)$$

$$p_{mle}(q_i|C) = \frac{ctf}{length(C)} \quad (3)$$

where, λ is the coefficient for linear interpolation based smoothing that accounts for question length smoothing and also compensates for differences in the word importance (gives idf-effects). Since the questions are of moderate length, after tuning, the best value of λ is attained at 0.75

In equation 2, μ is parameter for Bayesian smoothing using Dirichlet priors used for sentence length normalization, improving the estimates of the sentence sample.Since sentences of snippets can be of varying lengths, after tuning, the best value of μ is attained at 5000.

4.2 Sentence Selection

Once the top most relevant snippets have been chosen, we want to choose sentences from these snippets which are most relevant to a specific question. In this section we demonstrate how this selection is done.

4.2.1 MMR

We use the Maximum Marginal Relevance (MMR) algorithm (Forst et al., 2009) as the baseline for sentence selection. In contrast to the basic Jaccard similarity metric used in previous work (Chandu et al., 2017), we experimented with other similarity measures which consistently perform better than the Jaccard baseline. MMR ensures the selected set contains non-redundant yet complete information. The sentences are selected based on two aspects, the sentence's relevance to the question and how different it is to the already selected sentences. At each step we select a sentence to append to the ranking based on the equation below.

$$s_i = \arg\max_{s_j \in R \setminus S} (\lambda \cdot sim(q, s_i) - (1-\lambda) \cdot max_{s \in \mathcal{S}}(sim_{sent}(s_i, s_j))) \quad (4)$$

We define a custom similarity metric between sentences which uses positional values of sentences from the initial ranking as follows:

$$sim_{sent}(s_i, s_j) = (1-\beta) \cdot (1 - \frac{rank(d_i)}{n}) + \beta \cdot sim(s_i, s_j) \quad (5)$$

Here, $sim_{sent}(s_i, s_j)$ is the sentence to sentence similarity, $sim(q, s_i)$ is the question - sentence similarity, $rank(d_i)$ is the rank of the snippet d_i, which contains the sentence s_i, S are Sentences already selected for summary i.e. which are ranked above this position. In the above equation, we tried various metrics to account for the sentence to sentence similarity. In cases where β is non-zero, equation 4 is identified as our SoftMMR which includes soft scoring based on sentence position.

β	Configuration	Rouge-2	Rouge-SU4
-	baseline	0.7064	0.6962
0.5	BM25, Jaccard	0.7175	0.7110
0.5	BM25, Dice	0.7193	0.7106
0.6	BM25, Dice	0.7133	0.7053
0.6	BM25, Jaccard	0.7133	0.7053
0.5	**Indri, Jaccard**	**0.7206**	**0.7135**
0.5	Indri, Dice	0.7113	0.7052

Table 1: ROUGE scores for different experiments on similarity metrics for extractive summarization

4.2.2 Dice's similarity Coefficient (DSC)

Dice's similarity Coefficient (DSC) (Srensen, 1948) is a quotient of similarity between two samples and ranges between 0 and 1 calculated as

$$dsc = (2 * n_t)/(n_x + n_y)$$

where n_t is the number of character bigrams found in both strings, n_x is number of bigrams in string x and n_y is the number of bigrams in string y. We used Dice coefficient as a similarity metric between two sentences in 5

4.3 Evaluation

The pipeline described above is primarily designed to improve the ROUGE evaluation metric (Lin, 2004). Although a higher ROUGE score does not necessarily reflect improved human readability, MMR can improve readability by reducing redundancy in generated answers. Results for ideal answers for Task 5 phase b are shown in Table 1. We also compare our results with other state of the art approaches in Table 4.

5 Exact answers

Exact answers represent the subset of the BioASQ task where the responses are not structured paragraphs, but instead either a single entity (*yes/no* types) or a combination of named entities (*factoid* or *list* types) that compose the correct reply to the given query. The main idea refers to evaluating if a response is able to capture the most important components of an answer. For factoid or list types of questions, we must return a list of the most likely entities to compose the answer. The main difference between them is that ground truth for *factoid* questions is composed of only one correct answer and the evaluation method is Mean Reciprocal Rank (MRR). However, the ground truth for *list* is an actual list of correct answers with varying length, which uses F-measure as an evaluation metric. The BioASQ submission format allows everyone to submit 5 ranked answers for *factoid* and 1 to 10 answers for *list*. For *yes/no* questions, the ground truth is simply the yes or no label, using F-measure as an evaluation metric.

5.1 Yes/No type questions

Although yes/no questions require a simple binary response, calculating yes/no responses for the BioASQ question can be challenging:

1. There is an inherent class-bias towards the questions answered by `yes` in the dataset;
2. The dataset is quite small for training a complex semantic classifier;
3. An effective model must perform reasoning and inference using the limited information it has available, which is extremely difficult even for non-expert humans.

Due to the nature of the question type, these questions can not be simply classified by using word-level features. Learning the semantic relationship between the question and the sentences in the documents is quite elemental to solving this task. Hence, we present a Natural Language Inference (NLI)-based system that learns if the assertions made by the questions are true in the context of the documents. As a part of this system, we first generate assertions from questions and evaluate the entailment or contradiction of these assertions using a Recognizing Textual Entailment (RTE) model. We then use these entailment scores for all the sentences in the snippets or documents to heuristically evaluate if the answer to the yes/no question.

5.1.1 Assertion Extraction

The first step towards answering the question is to identify the assertions made by the question. For this, we use a statistical natural language parser to identify the syntactical structure in the question. We, then, heuristically generate assertions from the questions.

Consider the following example question:

Is the monoclonal antibody Trastuzumab (Herceptin) of potential use in the treatment of prostate cancer?

Upon parsing of this question, we have the phase constituents of the question. Almost all

yes/no questions have a standard format that begins with an auxiliary verb followed by a noun phrase. In this example, we can toggle the question word with the first noun phrase to generate the assertion:

The monoclonal antibody Trastuzumab (Herceptin) is of potential use in the treatment of prostate cancer.

In a similar manner, we then create positive assertions for all *yes/no* questions. As a simple extension to this, we can also create negative assertions by using *not* along with the auxiliary verbs.

5.1.2 Recognizing Textual Entailment

The primary goal of our NLI module is to infer if any of the sentences among the answer snippets entails or contradicts the assertion posed by the question. We segmented the answer snippets for each question to produce a set of assertion-sentence pairs. To then evaluate if these assertions can be inferred or refuted from the sentences, we built a Recognizing Textual Entailment (RTE) model using the *InferSent* model (Conneau et al., 2017), which computes sentence embeddings for every sentence and has been shown to work well on NLI tasks. In training *InferSent*, we experienced two major challenges:

1. The number of assertion-sentence pairs in BioASQ is too few to train the textual entailment model effectively.

2. The models that are pre-trained on SNLI (Bowman et al., 2015) datasets use GLOVE (Pennington et al., 2014) embeddings that cannot be used for biomedical corpora which have quite different characteristics and vocabulary compared to the corpora that GLOVE was trained on.

However, we have pre-trained embeddings available that were trained on PubMed and PMC texts along with Wikipedia articles (Pyysalo et al., 2013). To leverage these embeddings, we implemented an embedding-transformation methodology to projecting the PubMed embeddings to GLOVE embedding space and then fine tune the pre-trained *InferSent* on the BioASQ dataset for textual entailment. The hypothesis is that, since both the embeddings had a significant fraction of documents in common (Wikipedia corpus), by transforming the embeddings from one space to another, the sentence embeddings from the model would still represent a lot of the semantic features of the input sentences that can subsequently used for classifying textual entailment. For this task, we explore both linear and non-linear methods of embedding transformation.

While simple, a linear projection of embeddings from one space to another has shown to be quite effective for a lot of multi-domain tasks. By imposing an orthogonality constraint on the project matrix, we model this problem as an orthogonal Procrustes problem:
Let d_p and d_g be the embedding dimensions of PubMed embeddings and GLOVE embeddings respectively. If E_p and E_g are the matrices of PubMed embeddings ($N \times d_p$) and their corresponding GLOVE embeddings ($N \times d_g$) for the words that both the embeddings have in common (N), the projection matrix ($d_g \times d_p$) can be computed as,

$$W^* = \arg\min_W \|W E_p^\intercal - E_g^\intercal\|$$

subject to the constraint that W is orthogonal. The solution to this optimization problem is given by using the singular value decomposition of $E_g^\intercal E_p$, i.e.$W^* = UV^\intercal$ where $E_g^\intercal E_p = U\Sigma V^\intercal$ With this simple linear transformation, we then computed the transformed embeddings for all the words in the PubMed embeddings that are not present in the GLOVE embeddings.

We also explore a non-linear transformation using a feed-forward neural network where the the objective is to learn function f such that, $f(e_p; \theta) = e_g$ where, e_p and e_g are PubMed and GLOVE embeddings respectively. We model f using a deep neural network with parameters θ, and train using the common words in both the embeddings.

The transformed embeddings from these models were used in conjunction with the pre-trained *InferSent* model to encode the semantic features of the biomedical sentences as sentence embeddings. Subsequently, we employ these sentence embeddings of the assertion-sentence pairs for a particular question to train a three-way neural classifier to predict if the relationship between the two is entailment, contradiction or neither.

It is worth noting here that the embedding transformation techniques that we implemented are not specific to the NLI tasks and, in fact, enable transfer learning of a much broader set of tasks on smaller datasets like BioASQ by using the pre-

trained models on large datasets of other domains and fine-tuning on the smaller dataset.

5.1.3 Classification

As a final step, we use the textual entailment results for each assertion-sentence pair generated to heuristically classify the answer as *yes* or *no*. Since our system comprises multiple stages with the errors of each cascading to the final stage, we do not get perfect entailment results for the pairs. However, since we have a lot of pairs, we aggregate these entailment scores to compute the overall entailment or contraction scores to reduce the effect of accumulated errors for individual pairs on classification.

We used a simple unsupervised approach for classification by just comparing the overall entailment and contradiction scores, i.e. if the total number of snippet sentences that entail the assertion are N_e and the total number of snippet sentences that contradict are N_c, then,

$$\text{answer}_\text{q} = \begin{cases} \text{yes} & \text{if } N_e \geq N_c \\ \text{no} & \text{otherwise} \end{cases}$$

The end-to-end architecture of our system from the input questions and snippets to the answer is shown Figure 2.

5.1.4 Experimental Details

For parsing the questions, we used BLLIP reranking parser (Charniak and Johnson, 2005) (Charniak-Johnson parser) and used the model `GENIA+PubMed` for biomedical text. For training the textual entailment classifier using *InferSent*'s sentence embeddings, we used Stanford's SNLI dataset (Bowman et al., 2015) to achieve a test-set accuracy of 84.7%.

5.1.5 Results

The performance of the system on yes/no questions on the training set of phase 5b has been tabulated in table 2. While the accuracies are better than a random classifier, the task is far from being solved. Nonetheless, the classifier does handle the class bias in the training data and performance similarly on both the categories of answers. Moreover, this classifier achieved the second best test accuracy of 65.6% on phase 5 of BioASQ 5b (Table 4). While we implemented a simple heuristic based answer-classifier, we believe that a supervised classifier using the sentence embeddings as well as fine-tuning of the textual entailment classifier on BioASQ dataset would considerably enhance the overall performance of the system.

Category	Accuracy (%)
Yes	56.5 (252/444)
No	58.9 (33/56)
Overall	57.0 (285/500)

Table 2: Class-wise accuracies on yes/no questions in training set of BioASQ Phase 5b

5.2 Factoid & List Type Questions

Most of the state-of-the-art models for this task involve training end-to-end deep neural architectures to identify a subset of entities (or phrases) from the relevant snippets that are most likely to answer the question. But, owing to the small size of the dataset, we cannot effectively train such models on the BioASQ dataset. Hence, we adopted a two-stage approach that first finds a set of entities that could potentially answer the question and a supervised classifier to rank the entities on the basis of their likelihood of answering the question.

For devising the model and evaluation, we primarily focused on factoid type questions since the methodology for the list-type question would be largely similar and different only in the number of top entities returned.

5.2.1 Candidate Selection

We found that the most critical step in the answer generation process is to identify the set of potential answer candidates that can be fed into a classifier or ranker to identify the best candidates. At first, in order to accomplish this, we used Named Entity Recognition (NER) taggers to form a set of candidate answers. The taggers that we used include Gram-CNN (Zhu et al., 2017), LingPipe(Carpenter, 2007) and PubTator (Wei et al., 2013). To analyze the effectiveness of these taggers, we performed an analysis on BioASQ training set 5b by evaluating the fraction of questions whose answers are included in the candidate entity set by the taggers.

Table 3 shows the relative performances of the three taggers, their union as well as intersection on train dataset of BioASQ 5b factoid type questions. A question is exactly answered if a tagger tags an entity that matches an answer exactly, and it is partially answered if there is a non-zero over-

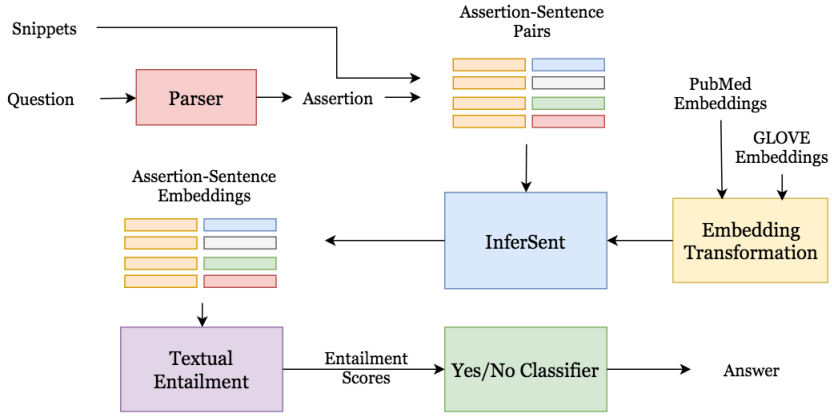

Figure 2: The complete system for yes/no answer classification using a question and relevant snippets

lap with an entity tagged and an answer for the question. We can notice that PubTator and LingPipe have a good recall with relatively low precision, while Gram CNN has high recall but low precision. However, the final results with the Named Entity Taggers were not aligned with our expectations. This is mostly because the answers for BioASQ are usually a combination of BioNERs and complementary words, making it hard to define a pruning method that is able to yield satisfactory results. Surprisingly, a group of candidates formed of the 100 most frequent n-grams (*n* from 1 to 4) from the snippets' sentences were a better candidate group than the NER approach for our supervised ranking method (with NER taggers used as features instead of candidate entities).

5.2.2 Classification Features

Upon computing the set of candidate answers, we use the question q, set of relevant snippet sentences $\mathcal{S}$ and entity type t_i to devise a feature vector for each individual entity e_i that comprises the following features:

- BM25 Score: The BM25 scores for all the sentences are computed with the question as the query. Then, the scores of the sentence that contain the entity are aggregated to compute the BM25 score for the entity, i.e.

$$\text{Score}_{BM25}(e_i) = \sum_{s \in \mathcal{S}} \text{Score}_{BM25}(e_i) \cdot \mathbb{1}(s, e_i)$$

  where $\mathbb{1}(s, e_i)$ is 1 iff sentence s has entity e_i.
- Indri Score: Computed in the same manner as BM25 score in (i)
- Number of Sentences: Number of sentences $s \in \mathcal{S}$ that contain the entity e_i
- NER Tagger: A multinomial feature that represents which tagger among PubTator, LingPipe and GramCNN the entity was extracted with. This feature is included to identify the relative strengths of the different taggers.
- Tf Idf: The aggregate Tf-Idf scores of the entity with $\mathcal{S}$ as the set of documents
- Entity Type: Is a boolean feature that is 1 if the type of the entity (for example, *gene*) is present in the question, and 0 otherwise.
- Relative Frequency: The amount of times the entity appears on the snippets' sentences divided by the total appearance of all of the relevant entities.
- Query Presence: Is a boolean feature that is 1 if the query contains the entity completely and 0 otherwise.

NER Tags	% of questions Exactly Answered	% of questions Partially Answered	% of tokens extracted
PubTator	32.05	72.15	52.27
Gram CNN	34.90	99.03	94.97
LingPipe	26.67	76.75	11.06
Union	49.04	99.65	99.25
Intersection	16.29	38.00	3.33

Table 3: Baseline recall of different NER Taggers measured by the fraction of questions that can be answered by an ideal classifier if the candidates are chosen using the tagger. We also measure precision as the fraction of total unique tokens from the documents that are tagged.

5.2.3 Unsupervised Ranking

As a baseline, we first present an unsupervised ranking system for the candidate answers. In this

Model	Exact Answers Yes/No type Accuracy (%)	Exact Answers Factoid type MRR	Exact Answers List type F1 score	Ideal Answers All types ROUGE-2
(Chandu et al., 2017)	-	-	-	0.653
(Peng et al., 2015)	**0.714**	0.272	0.187	-
(Wiese et al., 2017)	-	**0.392**	**0.361**	-
Sarrouti and Alaoui (2017)	0.461	0.207	0.243	0.577
BioAMA(Ours)	0.653	0.195	0.234	**0.721**

Table 4: Comparison of our model with other state of the art approaches

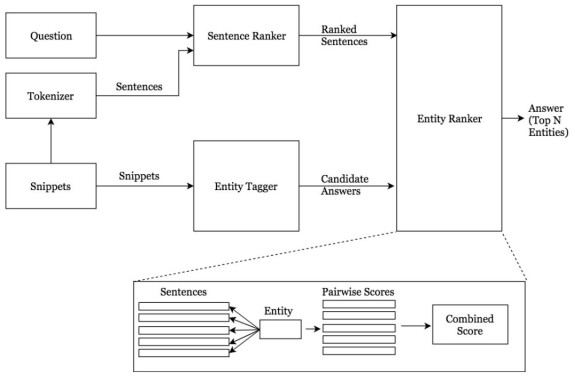

Figure 3: Unsupervised generation of factoid/list type answers using NER taggers and BM25 retrieval model

system, the snippet sentences are first ranked using the BM25 model. Then, for each entity, a score is computed by aggregating the BM25 scores of the sentences in which the entity is present. The rationale for this is that the entities in the top ranked sentences are more likely to be the answers. This entity score (which is equivalent to the BM25 score described in 5.2.2) is then used to rank the entities and return the top k entities as answers to the question. The overall unsupervised system is shown in Figure 3.

5.2.4 Learning To Rank

In order to rank the candidate entities in a supervised way, we use a ranking classifier based on the features described in 5.2.2. For ranking, we choose point-wise ranking classifiers over pairwise and list-wise, because it yields similar results to ranking methods with a less time-consuming and computationally expensive approach. We use a traditional SVM-Light (Joachims, 1998) implementation for point-wise ranking. The data for supervision is derived from the actual answers and candidate entities are ranked based on their overlap with the actual answers.

Once we rank the entities, we use a naive approach of merely taking top 5 entities as answers for factoid type and top 10 for list-type. One could, however, devise a separate model for identifying the number of top entities to return as answers for the list-type answers.

We found that using just the NER entities as the answer candidates, the classifier could achieve an MRR of 0.06 on factoid type questions and an F-measure of 0.18 for list type questions. However, by having all the n-grams ($n = 1, 2, 3, 4$) from the snippets as candidate answers and using NER tags as LeToR features, the performance was improved to an MRR of 0.195 on factoid type questions and an F1 score of 0.234 on list type questions. The results are summarized in Table 4.

6 Conclusion and Future Work

In this paper, we present a framework for tackling both ideal and exact answer type questions and obtain state of the art results on the ideal answer type questions on the BioASQ dataset. For exact answers, we incorporate neural entailment models along with a novel embedding transformation technique for answering yes/no questions, and employ LeToR ranking models to answer factoid/list based questions. For ideal answers, we improve the IR component of extractive summarization. Although this improves ROUGE scores considerably, the human readability aspect of the generated summary answer is not greatly improved. As future directions, we believe that effective abstractive summarization based approaches like Pointer Generator Networks (See et al., 2017) and Reinforcement Learning based techniques (Paulus et al., 2017) would improve the human readability of ideal answers. We aim to continue our research in this direction to achieve a good balance between ROUGE score and human readability.

References

Georgios Balikas, Anastasia Krithara, Ioannis Partalas, and George Paliouras. 2015. Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. In *Revised Selected Papers from the First International Workshop on Multimodal Retrieval in the Medical Domain - Volume 9059*, pages 26–39, New York, NY, USA. Springer-Verlag New York, Inc.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Bob Carpenter. 2007. Lingpipe for 99.99% recall of gene mentions. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, volume 23, pages 307–309.

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

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pages 173–180.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *CoRR*, abs/1705.02364.

Jan Frederik Forst, Anastasios Tombros, and Thomas Roelleke. 2009. Less is more: Maximal marginal relevance as a summarisation feature. In *Advances in Information Retrieval Theory*, pages 350–353, Berlin, Heidelberg. Springer Berlin Heidelberg.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proc. ACL workshop on Text Summarization Branches Out*, page 10.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *CoRR*, abs/1705.04304.

Shengwen Peng, Ronghui You, Zhikai Xie, Beichen Wang, Yanchun Zhang, and Shanfeng Zhu. 2015. The fudan participation in the 2015 bioasq challenge: Large-scale biomedical semantic indexing and question answering. In *CLEF*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

S. Pyysalo, F. Ginter, H. Moen, T. Salakoski, and S. Ananiadou. 2013. Distributional semantics resources for biomedical text processing. In *Proceedings of LBM 2013*, pages 39–44.

Dragomir Radev, Y Hong Qi, Harris Wu, and Weiguo Fan. 2003. Evaluating web-based question answering systems.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

Mourad Sarrouti and Said Ouatik El Alaoui. 2017. A biomedical question answering system in bioasq 2017. In *BioNLP 2017*, pages 296–301. Association for Computational Linguistics.

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

Trevor Strohman, Donald Metzler, Howard Turtle, and W. Bruce Croft. 2005. Indri: a language-model based search engine for complex queries.

T. Srensen. 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. In *Kongelige Danske Videnskabernes Selskab*, pages 1–34.

Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. 2013. Pubtator: a web-based text mining tool for assisting biocuration. *Nucleic acids research*, 41(W1):W518–W522.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Fastqa: A simple and efficient neural architecture for question answering. *CoRR*, abs/1703.04816.

Georg Wiese, Dirk Weissenborn, and Mariana L. Neves. 2017. Neural question answering at bioasq 5b. *CoRR*, abs/1706.08568.

Qile Zhu, Xiaolin Li, Ana Conesa, and Ccile Pereira. 2017. Gram-cnn: a deep learning approach with local context for named entity recognition in biomedical text. *Bioinformatics*, page btx815.