Towards Complex Biomedical Question Answering System

Submitted by:

Hasibur Rahman

ID: 2016-2-60-049

Mariam Nahar Moni

ID: 2016-2-60-016

Azmain Iktidar

ID: 2016-1-60-063

In partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

Department of Computer Science and Engineering

East West University

Dhaka-1212, Bangladesh

October 23th, 2020

Declaration

We, hereby, declare that the work presented in this thesis is our own work and the outcome of the investigation performed by us under the supervision of our supervisor Md. Mohsin Uddin Senior Lecturer, Department of Computer Science & Engineering, East West University. We also declare with our best knowledge and belief that this approach has done beforehand but we ensure that our work has been improved and upgraded than previous work. So, our must be better than previous one.

. . . . . .

(Md. Mohsin Uddin)
Supervisor

. . . . . .

(Hasibur Rahman)

…….

(Mariam Nahar Moni)

…….

(Azmain Iktidar)

ABSTRACT

Biomedical experts and bio-curators are unable to quickly find short and precise information using typical search engine as the amount of biomedical literature is increasing exponentially. The research community is focusing on biomedical question answering (QA) systems so that anyone can find precise information nuggets from the massive amount of biomedical literature. Generally, the user queries fall under different categories such as factoid, list, yes/no, or summary. The existing state-of-the-art question answering systems deal with most of these question types. However, there search to improve the performance of individual question types is also on the rise. To improve QA system performance, question classification plays a vital role for factoid and list type questions as it allows the answer processing stage to narrow down the candidate answer space and assigns a higher rank to the correct answers.

Till now the system 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. Also, for factoid and list -type question the previous system got low accuracy (which suggests that their algorithm needs to improve in the ranking of entities).

But we introduce a novel biomedical question answering (QA) dataset collected from MedQuAD_GHR_QA abstracts. The task of MedQuAD is to answer Medical related questions with five types of question type using the corresponding abstracts. In this paper, we propose a novel QA system and resources that we built and evaluated on real medical questions. First, we compare machine learning and deep learning methods using our of datasets, including textual inference, question similarity and entailment in both the open and clinical domains. Each MedQuAD instance is composed of (1) a question which is either an existing research article title or derived from one, (2) question type including- information, frequency, genetic changes, inheritance, treatment & (3) a long answer, which is the conclusion of the abstract and, presumably, answers the research question. To study the end-to-end QA approach, we built the MedQuAD dataset collection of 5034 question-answer pairs from trusted medical sources, we introduce and share in the scope of this paper. The evaluation results also support the relevance of question entailment for QA. Our findings also show that relying on a restricted set of reliable answer sources can bring a substantial improvement in medical QA.

Experimental analysis reveals that the proposed algorithm outperforms other traditional deep learning techniques. We have implemented convolutional neural network (CNN, or ConvNet), recurrent neural network (RNN), Long short-term memory (LSTM), bi-directional LSTM, Gated recurrent units (GRUs), sequence to sequence, one hot encoding, encoder decoder and attention. Our Proposed approach has 93.72% accuracy in whole Complex dataset.

We hope that this shared task will attract further research efforts in textual inference, question entailment, and question answering in the medical domain.

Towards Complex Biomedical Question Answering System----------1

Submitted by: ----------1

TABLE OF CONTENT ----------3

LIST OF FIGURES----------6

LIST OF TABLES ----------7

CHAPTER 1 ----------1

INTRODUCTION----------1

1.1 Introduction----------1

1.2 Motivation ----------2

1.3 Objectives ----------3

1.4 contribution----------4

1.5 Outline ----------5

CHAPTER 2 ----------6

LITERATURE REVIEW----------6

2.1 BACKGROUND----------6

2.1.1 Long Short-Term Memory (LSTM) ----------6

2.1.2Recurrent Neural Network (RNN) ----------7

2.1.3 Bidirectional LSTM---------- 9

2.1.4 Gated recurrent Unit ----------10

2.1.5 Convolutional Neural Network (CNN) ----------10

2.2 Related work----------12

CHAPTER 3----------14
PROPOSED METHOD ----------14
3.1 factoid type ---------- 15
3.2 Encoder-Decoder Sequence to Sequence Model----------16
3.3 Attention Model ----------16
CHAPTER 4----------18
DATASET OVERVIEW----------18
4.1 Attributes Description----------18
CHAPTER 5----------23
IMPLEMENTATION----------23
5.1 Factoid type question answering----------23
5.2 Complex type question answering----------25
CHAPTER 6----------29
RESULT ANALYSIS----------29
6.1 Performance analysis----------29
CHAPTER 7----------35
CONCLUSION AND FUTURE WORKS----------35
REFERENCES----------36

LIST OF FIGURES

Figure 2.1.1: Long short term memory ---------- 6

Figure 2.1.2: Recurrent Neutral Network ---------- 7

Figure 2.1.4: Gated recurrent Unit---------- 10

Figure 2.1.5: Array of RGB matrix ----------11

Figure 2.1.6: Neural Network with many convolutional layers ----------12

Figure 3.1: Factoid type proposed method----------15

Figure 3.2: Encoder-decoder sequence to sequence to model----------16

Figure 3.3: Attention Model ----------17

Figure 4.1: Total column in factoid type dataset---------- 18

Figure 4.2: Total Occurrence of each significance in dataset ----------20

Figure 4.3: Examples of QA pairs generated from an article ----------22

Figure 4.4: Total column in complex type dataset----------22

Figure 6.1: Accuracy, precision, recall comparison of the algorithms----------31

Figure 6.2: Comparison of proposed work and related paper work---------- 32

Figure 6.3: Accuracy, precision, recall comparison of all algorithm ---------- 33

Figure 6.4: Comparison of proposed work and related paper work----------34

LIST OF TABLES

Table 4.1: Total column in factoid type dataset.....18

Table 6.1: Accuracies, precisions, recalls of the algorithms.....31

Table 6.2: Performance comparison of the proposed approach.....32

Table 6.3: Accuracies, precisions, recalls of the algorithm.....33

Table 6.4: Performance comparison of the proposed approach……34

ACKNOWLEDGEMENT

In the name of Allah, the most beneficent and merciful, we express our sincere gratitude towards the almighty Allah who gave us strength, patience and knowledge to complete this thesis work. After that, we would like to express our gratefulness towards our supervisor, Md. Mohsin Uddin, Senior Lecturer, Department of Computer Science & Engineering, East West University for giving us this opportunity to work into the field of machine learning. Throughout our work, he was always there to guide us and help us to improve more. He gave us moral support and guided in different matters regarding the work. Without his proper guidance, support and encouragement, we wouldn't able to complete this thesis work perfectly. His encouragements, visionaries and thoughtful comments and suggestions, unforgettable support at every stage of our B.Sc. study were simply appreciating and essential. We are also thankful to our parents who supported us mentally. Lastly we would like to thank other faculties of our department and our friends for their support and encouragement regarding our thesis work.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Every day new symptoms of different diseases along with their possible cures and new precautions are being discovered and published. In America and Europe, 30 million people are suffering from rare diseases. Fifty percent patients of these rare diseases are children, and it causes death in 35percent of newborn babies. Due to this reason, the biomedical information is exciting not only for biomedical experts but layman users also search for symptoms of different diseases. They also seem curious to know the symptom, overlook of various diseases. It is quite challenging to find such precise information from the massive amount of available literature.

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 question is still an open problem. Our paper presents is also about providing factoid and list type answers in Biomedical.

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 (VasuSharma*, 2018).

The Question Answering (QA) systems aim to solve the information overload problem by linguistically and semantically processing the user-posted query and providing exact answers to a user query or question. The question answering (QA) task, in which models learn how to answer questions, is often used as a benchmark for quantitatively measuring the reasoning and inferring abilities of such intelligent systems. With the abundance of information sources in the medical domain, consumers are more and more faced with a similar challenge, one that needs dedicated solutions that can adapt to the heterogeneity and specifics of health-related information.

In the context of QA, the goal of Recognizing Question Entailment (RQE) is to retrieve answers to a premise question (PQ) by retrieving inferred or entailed questions, called hypothesis questions (HQ) that already have associated answers. RQE is particularly relevant due to the increasing numbers of similar questions posted online and its ability to solve differently the challenging issues of question understanding and answer extraction (DinaDemner-Fushman, 2019). In addition to being used to find relevant answers, these resources can also be used in training models able to recognize inference relations and similarity between questions.

There are Factoid and list type questions expect single biomedical entity or a list of biomedical entities as answer respectively. Finally, summary type questions expect a summarized form of one or more documents as an answer. In the biomedical domain, QA systems may be designed to deal with a particular question type or it may handle more than one type of questions.

Our approach for implementing the system are Gated recurrent unit(GRUs), sequence to sequence , Long Short Term Memory cells (LSTMs), and Convolutional Neural Networks (CNNs), bi-directional LSTM, one hot encoding, encoder decoder and attention etc.

In this paper, we aim at building a biomedical QA dataset which (1) has substantial instances with some expert annotation and (2) require reasoning over the contexts to answer the questions. For this, we turn to the PubMed1, a search engine providing access to over 25 million references of biomedical articles. We found that around 760k articles in PubMed use questions as their titles. Among them, the abstracts of about 120k articles are written in a structured style – meaning they have subsections of "Introduction", "Results" etc. Conclusive parts of the abstracts, often in "Conclusions", are the authors' answers to the question title. Other abstract parts can be viewed as the contexts for giving such answers. This pattern perfectly fits the scheme of QA, but modeling it as abstractive QA, where models learn to generate the conclusions, will result in an extremely hard task due to the variability of writing styles (QiaoJin, 2019).

In case of better justification on our proposed approach we have implemented well known approaches comparatively. Moreover, manual statistical analysis has also been conducted on our dataset for making our model stronger to accept.

1.2 Motivation

Answering a question immediately produces a short or paragraph-level response to directly response a user's question; answering a question goes beyond retrieving information, where a large number of documents are normally retrieved by a user's query.

In the biomedical sector, physicians have found that the use of established medical companies (e.g. literature and online biomedical databases) is not responsible for several questions asked by physicians (DinaDemner-Fushman, 2019). Unanswerable questions include questions that do not discuss the particular area, patient-specific questions that involve knowledge from the record of a patient, and questions to which the responses are normally unknown. We are investigating whether certain unanswerable questions can be detected automatically; if such a question is found, we can prompt the user to reformulate the question.

By asking questions anytime, anywhere, mass people can get any disease-related response. This system can be very useful for them in a pandemic situation such as now when people can't go outside the house.

In order to produce a coherent paragraph-level response, BioQA further extracts responses at the sentence level and applies text summarization. The first part of BioQA is query analysis.

While a lot of research has been done to define relevant data to make the system correctly and timely, there is a lack of an automated framework for evaluating the system of question answering. We have therefore successfully developed the method from the available data sources with an accuracy of 97.31 percent for complex question answering, using machine learning and deep learning approaches.

1.3 Objective

The main objectives of our research are as follows:

I. 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.

II. To implement the High Performance biomedical question answering system.

III. Better understanding of biomedical question answering for experts.

IV. Allow directly query the data and extract meaningful information.

V. Avoid time consuming and extra effort.

1.4 Contribution

Contribution in our research are as follows:

We have suggested machine learning and deep learning approaches to respond to the concept of different human diseases, affected percentage worldwide, overlook, treatment after asking the question.

Deep learning methods are essentially computer algorithms of ensemble types that extract significant data within large datasets. Different techniques of deep learning as well as machine learning have a major influence on medical science today.

We focused on neural network optimization to find overall accuracy and convolution neural network efficiency depends largely on two parameters, such as question and answer.

Firstly we try to find out the factoid type information from our dataset which consist of 5430 row and 2 column. This is multiclass classification problem which has 5 class level for solving this problem we use word embedding (custom word2vec model). Then we define different types of model such as- : Gated recurrent unit (GRUs), bidirectional Long Short Term Memory cells (LSTMs), Long Short Term Memory cells (LSTMs) and Convolutional Neural Networks (CNNs), RNN etc. After that we evaluate accuracy, precision, recall for each and every model.

After that we focused on complex type question answering with two approach. First approach is sequence to sequence encoder decoder without attention and second approach is sequence to sequence encoder decoder with attention .Here we overall applied well-known approaches such as: sequence to sequence, encoder decoder and attention for the better performance of our proposed method. We applied it on the complex dataset and hence generated association rules from the dataset.

1.5 Outline

Chapter 1: Chapter 1 introduces our biomedical system, our motivation to make the system, the main objectives of our research, the contributions that we have made regarding the disease.

Chapter 2: This chapter illustrates the background of our proposed methods and the related works that have been done regarding our research so far.

Chapter 3: shows the architectural view of our proposed method.

Chapter 4: This chapter shows the analysis of our whole dataset.

Chapter 5: Chapter 5 describes the implementation process, algorithms that are used and codes.

Chapter 6: This chapter analyzes the results obtained from our proposed methods.

Chapter 7: The final chapter summarizes the overall work that we have done and also explains the future works that we need to focus on.

CHAPTER 2

LITERATURE REVIEW

2.1 Background

2.1.1 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition, and more. LSTMs are a complex area of deep learning. It can be hard to get your hands around what LSTMs are, and how terms like bidirectional and sequence-to-sequence relate to the field. There are few that are better at clearly and precisely articulating both the promise of LSTMs and how they work than the experts that developed them.

An LSTM has a similar control flow as a recurrent neural network. It processes data passing on information as it propagates forward. The differences are the operations within the LSTM's cells.

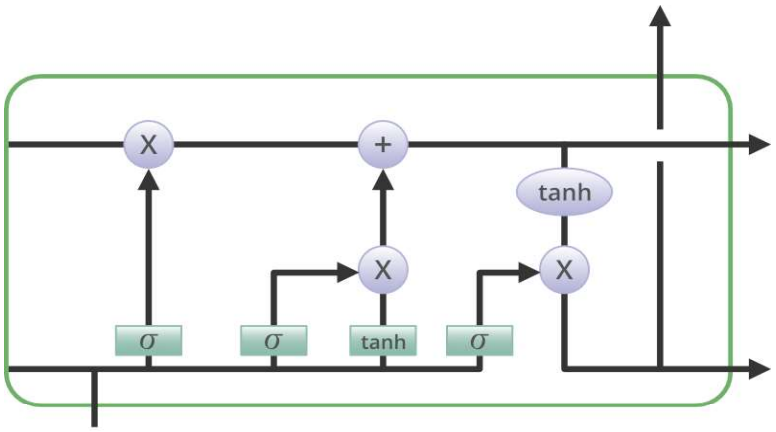


Figure 2.1.1: Long short term memory

2.1.2 Recurrent Neural Network (RNN)

RNN means Recurrent Neural Network. This is a type of neural network in which the output from the previous step are fed as input to the current step.

Naturally in neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words.

Then RNN came into existence for solving this issue with the help of a Hidden Layer. The main and most important feature of RNN is Hidden state, which remembers some information about a sequence. In RNN there is a part called memory in which it remembers all information about what has been calculated. It uses the same parameters are used as each input as it performs the same task on all the inputs or hidden layers to produce the output. The complexity of parameters are reduced by this processed, unlike other neural networks.

Suppose there is a deeper network with has one input layer, three hidden layers and one output layer. Each hidden layer will have its own set of weights and biases, let's say, for hidden layer 1 the weights and biases are (w1, b1), (w2, b2) for second hidden layer and (w3, b3) for third hidden layer. This means that each of these layers are independent of each other, they do not memorize the previous outputs.

The RNN will work on the following way:

Independent activations converted into dependent activations by providing the same weights and biases to all the layers, thus reducing the complexity of increasing parameters and memorizing each previous outputs by giving each output as input to the next hidden layer.

These three layers can be joined together such that the weights and bias of all the hidden layers is the same, into a single recurrent layer.

are used as each input as it performs the same task on all the inputs or hidden layers to produce the output. The complexity of parameters are reduced by this processed, unlike other neural networks.

Suppose there is a deeper network with has one input layer, three hidden layers and one output layer. Each hidden layer will have its own set of weights and biases, let's say, for hidden layer 1 the weights and biases are (w1, b1), (w2, b2) for second hidden layer and (w3, b3) for third hidden layer. This means that each of these layers are independent of each other, they do not memorize the previous outputs.

The RNN will work on the following way:

Independent activations converted into dependent activations by providing the same weights and biases to all the layers, thus reducing the complexity of increasing parameters and memorizing each previous outputs by giving each output as input to the next hidden layer.

These three layers can be joined together such that the weights and bias of all the hidden layers is the same, into a single recurrent layer.

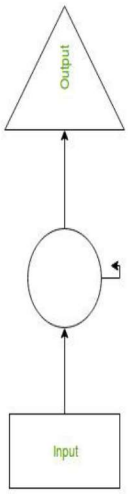


Figure 2.1.2: Recurrent Neural Network

2.1.3 Bidirectional LSTM

Bidirectional LSTMs are an extension of traditional LSTMs that can improve model performance on sequence classification problems.

In problems where all timestamps of the input sequence are available, Bidirectional LSTMs train two instead of one LSTMs on the input sequence. The first on the input sequence as-is and the second on a reversed copy of the input sequence. This can provide additional context to the network and result in faster and even fuller learning on the problem.

Bidirectional LSTMs are supported in Keras through the Bidirectional layer wrapper.

This wrapper takes a recurrent layer as an argument.

It also allows you to specify the merge mode, that is how the forward and backward outputs should be combined before being passed on to the next layer. The options are:

- '*sum*': The outputs are added together.
- '*mul*': The outputs are multiplied together.
- '*concat*': The outputs are concatenated together.
- '*ave*': The average of the outputs is taken.

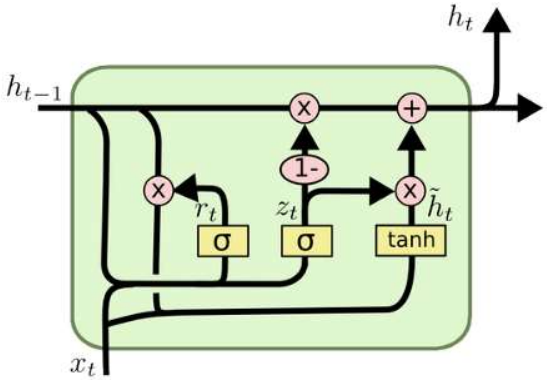
The default mode is to concatenate, and this is the method often used in studies of bidirectional LSTMs.

2.1.4 Gated recurrent units (GRUs)

Gated recurrent units (GRUs) are a gating mechanism in recurrent neural networks, introduced in 2014 by Kyunghyun. Cho et al. The GRU is like a long short-term memory (LSTM) with a forget gate, but has fewer parameters than LSTM, as it lacks an output gate. GRU is a variant of LSTM.GRU retains the vanishing gradient properties of LSTM is retained by GRU but it is internally simpler and faster than LSTM.

There are 3 gate in LSTM: input, output and forget gates. But in GRU we only have two gates an update gate z and a reset gate r. There is no persistent cell state distinct from the hidden state in GRU.

Here there is a equation for GRU gating mechanism:



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 2.1.4: Gated recurrent Unit

2.1.5 Convolutional Neural Network (CNN)

A convolutional neural network (CNN) is a type of artificial neural network which is mostly used in image processing and recognition which is specifically designed to process pixel data. Basically, CNNs are powerful image processing which is used in deep learning for performing both generative and descriptive tasks but sometimes using machine approaches that includes image and video recognition, and natural language processing (NLP). A CNN uses a framework much like a multi-layer perceptron that has been intended for diminished processing prerequisites.

The layers of a CNN comprise of an input layer, an output layer and a hidden layer that incorporates numerous convolutional layers, pooling layers, fully connected layers and normalization layers. The evacuation of constraints and increment in effectiveness for image handling results in a framework that is unquestionably effective and simpler to train for image processing and natural language processing (Rouse, 2018-2020).

In CNN image classifications takes an input image, then process it and classify it under certain categories. Based on the image resolution, it will see h x w x d (h = Height, w = Width, d = Dimension). For example, an image of 6 x 6 x 3 array of matrix of RGB (Raghav Prabhu, 2019).

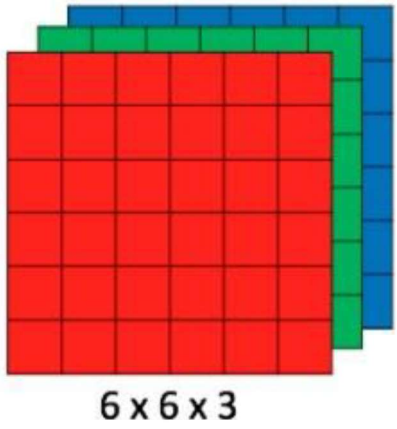


Figure 2.1.5: Array of RGB matrix

In deep learning CNN models, each input image will pass through a convolutional layers with pooling, kernels, and fully connected layers and then apply softmax function for classifying an object with probabilistic values between 0 and 1.

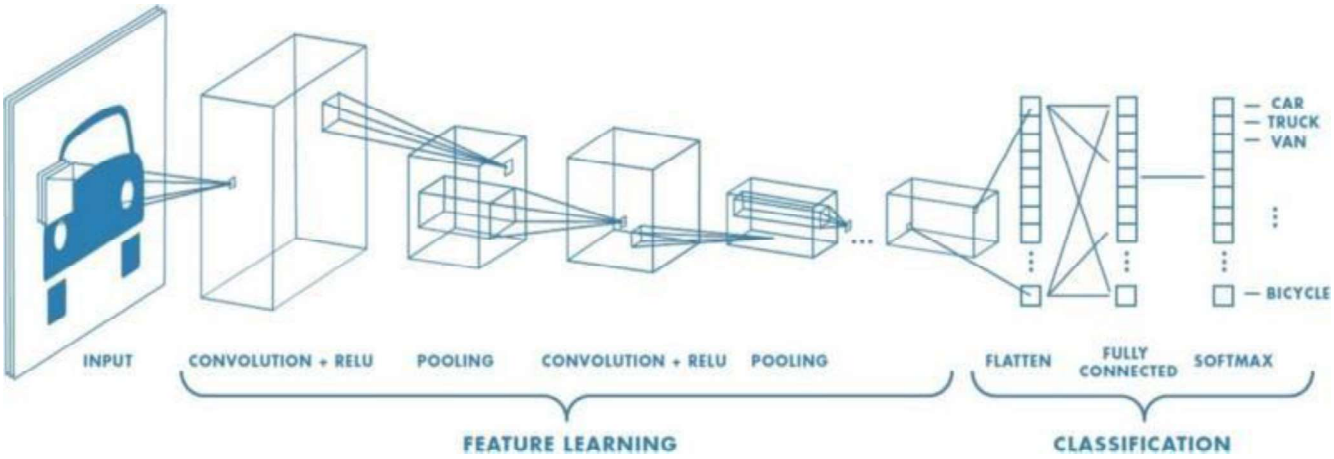


Figure 2.1.6: Neural Network with many convolutional layers

2.2 Related Work

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. Thousands of contributions regarding medical question answering have been conducted by lots of researchers and academics but most of the works have been conducted in a traditional manner.

In this era of science, technology and big data, the traditional way of determination of medical information are not feasible enough. Various works have been conducted to analyze it, however very few have been found where machine learning have been applied effectively. 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.

An alternative approach consists in finding similar questions or FAQs that are already answered] (V. Jijkounand M. de Rijke , 2005) One of the earliest question answering systems based on finding similar questions and re-using the existing answers was FAQ FINDER (R. D. Burke, K. J. Hammond, V. Kulyukin, S. L. Lytinen, N. Tomuro, and S. Schoenberg, 1997). Another system that complements the existing Q&A services of NetWellness3 is SimQ (J. Luo, G.-Q. Zhang, S. Wentz, L. Cui, and R. Xu, 2015), which allows retrieval of similar web-based consumer health questions. SimQ uses syntactic and semantic features to compute similarity between questions, and UMLS (D. A. Lindberg, B. L. Humphreys, and A. T. McCray, 1993)as a standardized semantic knowledge source. The system achieves 72.2% precision, 78.0% recall and 75.0% F-score on Net Wellness questions. However, the method was evaluated only on one question similarity dataset, and the retrieved answers were not evaluated.

The aim of the medical task at TREC 2017 Live QA was to develop techniques for answering complex questions such as consumer health questions, as well as to identify relevant answer sources that can comply with the sensitivity of medical information retrieval.

The CMU-OAQA system (D. Wang and E. Nyberg, 2017) achieved the best performance of 0.637 average score on the medical task by using an attentional encoder-decoder model for paraphrase identification and answer ranking. The Quora question-similarity dataset was used for training. The PRNA system (V. V. Datla, T. R. Arora, J. Liu, V. Adduru, S. A. Hasan, K. Lee, A. Qadir, Y. Ling, A. Prakash, and O. Farri, 2017) achieved the second best performance in the medical task with 0.49 average score using Wikipedia as the first answer source and Yahoo and Google searches as secondary answer sources.To extract the answer from the selected text passage, a bi-directional attention model trained on the SQUAD dataset was used.

Deep neural network models have been pushing the limits of performance achieved in QA related tasks using large training datasets. The results obtained by CMU-OAQA and PRNA showed that large open-domain datasets were beneficial for the medical domain. However, the best system (CMU-OAQA) relying on the same training data obtained a score of 1.139 on the Live QA open-domain task.

While this gap in performance can be explained in part by the discrepancies between the medical test questions and the open-domain questions, it also highlights the need for larger medical datasets to support deep learning approaches in dealing with the linguistic complexity of consumer health questions and the challenge of finding correct and complete answers.

Another technique was used by ECNU-ICA team (W. An, Q. Chen, W. Tao, J. Zhang, J. Yu, Y. Yang, Q. Hu, L. He, and B. Li. ECNU, 2017) based on learning question similarity via two long short-term memory (LSTM) networks applied to obtain the semantic representations of the questions. To construct a collection of similar question pairs, they searched community question answering sites such as Yahoo! and Answers.com. In contrast, the ECNU-ICA system achieved the best performance of 1.895 in the open-domain task but an average score of only 0.402 in the medical task. As the ECNU-ICA approach also relied on a neural network for question matching, this result shows that training attention-based decoder-encoder networks on the Quora dataset generalized better to the medical domain than training LSTMs on similar questions from Yahoo! and Answers.com.

The CMU-Live Med QA team (Kai Wang, 2009) designed a specific system for the medical task. Using only the provided training datasets and the assumption that each question contains only one focus, the CMU-Live Med QA system obtained an average score of 0.353. They used a convolutional neural network (CNN) model to classify a question into a restricted set of 10 question types and crawled "relevant" online web pages to find the answers. However, the results were lower than those achieved by the systems relying on finding similar answered questions. These results support the relevance of similar question matching for the end-to-end QA task as a new way of approaching QA instead of the classical QA approaches based on Question Analysis and Answer Retrieval.

We carefully analyze the pipeline of QA system for factoid and list type questions. To deal with the multi-label nature of biomedical LATs, the QA system uses copy transformation technique. Once transformed, a one-vs-all multi-class classification approach is used to rank and select top k LATs. We find that the system uses an automated corpus generation process to train the LAT prediction model. For all the entities for which the system is unable to identify a class label, a null type is assigned to the questions. Such assignment for multiple questions confuses the classifier as many questions are labeled as null. Out of 899 question in the training data, 119 questions were assigned only with null type. We use an improved dataset to increase the performance of both factoid and list type questions. Furthermore, we introduce binary transformation to further boost the performance for factoid type questions.

CHAPTER 3

PROPOSED METHOD

Since, the number of disease databases are increasing day by day and so only clinical intervention is not enough to detect definition, genetic changes, symptom, overlook, treatment of all the disease within a short time. Till now for ideal answers, researcher use extractive summarization technique on relevant snippets. They used extractive summarization pipeline uses lexical chaining for sentence similarity and ranking.

For complex type question answering we select the top N sentences such that the total number of words doesn't exceed the 120-word limit, and concentrate on the main focus of final answer. To construct trusted medical question-answer pairs, we crawled websites from the National Institutes of Health7. We also annotated each question with the associated focus (topic of the web page) as well as the question type identified with the designed patterns.

Overall our proposed method is we firstly construct two types of dataset for factoid and complex type question answering. Then we construct code for this two approach. Factoid types give us the question type as output for different types of biomedical question. Whereas complex type gives us the proper answer for those biomedical question.

However, proposed of this work consists of the following segments.

- Extracting data from dataset file and splitting them into train (80%), test (20%) and cross_ validation set.
- For factoid type we used word embedding which is custom word2vec model.
- Further applying several types of model: Gated recurrent unit (GRUs), bidirectional Long Short Term Memory cells (LSTMs), Long Short Term Memory cells (LSTMs) and Convolutional Neural Networks (CNNs), Recurrent Neural Network (RNN).
- Constructing this we evaluate every model's accuracy, precision, recall.
- Finally, performances of all the classification approaches have been analyzed, compared and demonstrated based on accuracy, precision, recall for the approach have been illustrated.
- For complex type we applied two approach. First approach is sequence to sequence encoder decoder without attention and second approach is sequence to sequence encoder decoder with attention. We use embedding vector for representing the input as some sort of integer
- Here we overall applied well-known approaches such as: sequence to sequence, encoder decoder, word embedding and attention for the better performance of our proposed method.

Thus we constitute our whole thesis work for the best performance throughout the biomedical question answering dataset.

3.1 Proposed Work: factoid type

Factoid type step:

- We Convert XML into Data Frame.
- In the preprocessing steps we construct Basic data cleaning
- We use Data cleaning for NLP
- Dataset Train-test split
- We implement Custom word2vec model
- We create LSTM, Bidirectional LSTM, GRU, simple RNN, CNN model
- We Train the model
- At last our code make predictions from test dataset

However, complete infrastructure of our proposed approach has been represented graphically in Figure 3.1.

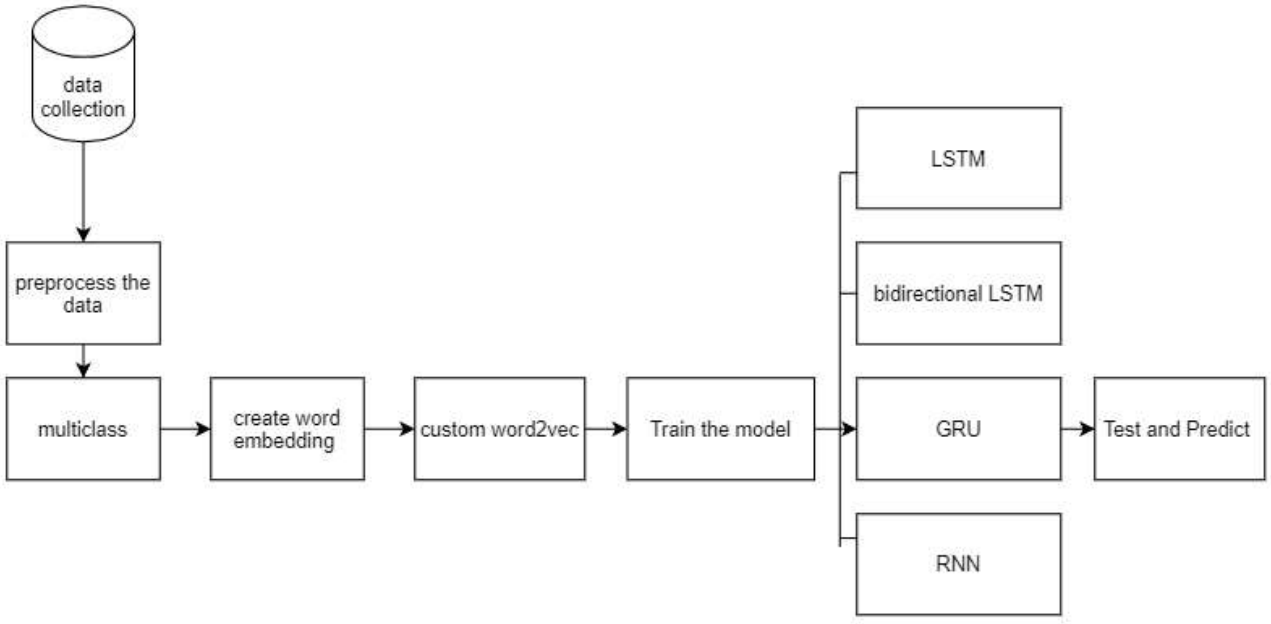


Figure 3.1: Factoid type proposed method

3.2 Proposed work: Encoder-Decoder Sequence to Sequence Model

The encoder-decoder architecture is a neural network design pattern. The encoder's role is to encode the inputs into state, which often contains several tensors. Then the state is passed into the decoder to generate the outputs. In machine translation, the encoder transforms a source sentence, like example "Hello world." into state, a vector, that captures its semantic information, The decoder then uses this state to generate the translated target sentence.

The decoder has an additional method initial state to parse the outputs of the encoder with possible additional information, the valid lengths of inputs, to return the state it needs.

In the forward method, the decoder takes both inputs, a target sentence and the state. It returns outputs, with potentially modified state if the encoder contains RNN layers.

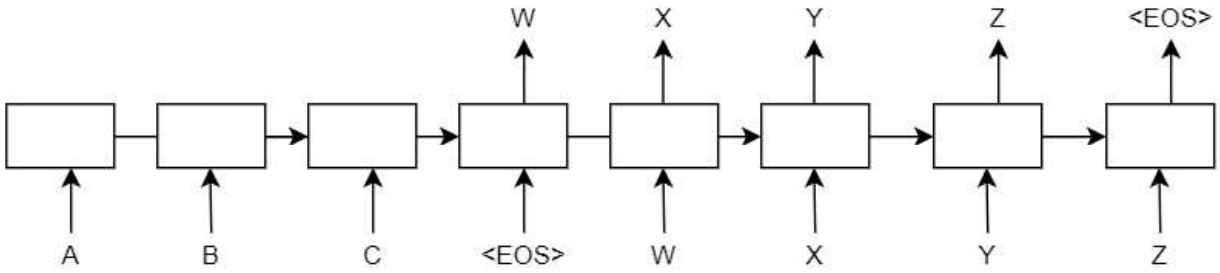


Figure 3.2: Encoder-decoder sequence to sequence to model

3.3 Proposed work: Attention Model

Attention is proposed as a solution to the limitation of the Encoder-Decoder model encoding the input sequence to one fixed length vector from which to decode each output time step. This issue is believed to be more of a problem when decoding long sequences.

A potential issue with this encoder–decoder approach is that a neural network needs to be able to compress all the necessary information of a source sentence into a fixed-length vector.

Attention is proposed as a method to both align and translate. Alignment is the problem in machine translation that identifies which parts of the input sequence are relevant to each word in the output, whereas translation is the process of using the relevant information to select the appropriate output. We introduce an extension to the encoder–decoder model which learns to align and translate jointly. Each time the proposed model generates a word in a translation, it (soft-) searches for a set of positions in a source sentence where the most relevant information is concentrated.

The model then predicts a target word based on the context vectors associated with these source positions and all the previous generated target words. Instead of encoding the input sequence into a single fixed context vector, the attention model develops a context vector that is filtered specifically for each output time step.

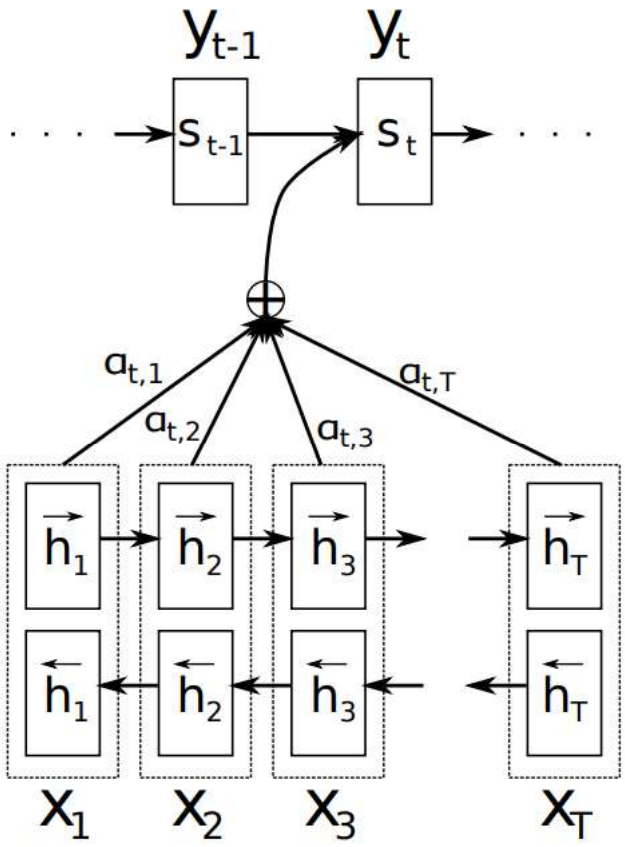


Figure 3.3: Attention Model

This is the diagram of the Attention model shown in [20]. The Bidirectional LSTM used here generates a sequence of annotat*ions (h1, h2,….., hTx)* for each input sentence. All the vectors h1,h2.., etc., used in their work are basically the concatenation of forward and backward hidden states in the encoder. $h_j = \left[\overrightarrow{h}_j^\top ; \overleftarrow{h}_j^\top\right]^\top$

To put it in simple terms, all the vectors h1,h2,h3…., hTx are representations of Tx number of words in the input sentence. In the simple encoder and decoder model, only the last state of the encoder LSTM was used (hTx in this case) as the context vector.

CHAPTER 4

DATASET OVERVIEW

4.1 Attributes Description

To construct trusted medical question-answer pairs, we crawled websites from the National Institutes of Health[1]. Each web page describes a specific topic (e.g. name of a disease or a drug), and often includes synonyms of the main topic that we extracted during the crawl. We constructed hand-crafted patterns for each website to automatically generate the question-answer pairs based on the document structure and the section titles. We also annotated each question with the associated focus (topic of the web page) as well as the question type identified with the designed patterns. To provide additional information about the questions that could be used for diverse IR and NLP tasks, we automatically annotated the questions with the focus, its UMLS Concept Unique Identifier (CUI) and Semantic Type. We combined two methods to recognize named entities from the titles of the crawled articles and their associated UMLS CUIs: (i) exact string matching to the UMLS Metathesaurus[2], and (ii) Meta Map Lite[3]. We then used the UMLS Semantic Network to retrieve the associated semantic types and groups.

In factoid type question answering the dataset contains total 5430 number of disease related information.

Among all the attributes collected, there are 5 categorical questions all over and we converted it to numerical. The 5 categorical questions are:

information	What is (are) Aarskog-Scott syndrome?
frequency	How many people are affected by Aarskog-Scott syndrome?
genetic changes	What are the genetic changes related to Aarskog-Scott syndrome?
inheritance	Is Aarskog-Scott syndrome inherited?
treatment	What are the treatments for Aarskog-Scott syndrome?

Table 4.1: Total column in factoid type dataset

[1] www.nih.gov

[2] We used the umls-2017AA version

[3] https://metamap.nlm.nih.gov/MetaMapLite.shtml

And the numerical encode we uses in the code are:

Genetic changes: 0,
Inheritance: 1,
Frequency: 2,
Information: 3,
Treatment: 4.

The question types were derived after the manual evaluation of 1,721 consumer health questions. Our taxonomy includes several types about Diseases, 20 types about Drugs and one type (Information) for the other named entities such as Procedures, Medical exams and Treatments.

Overall we describe below the considered question types and examples of associated question patterns. 1. Question Types about biomedical : Information, Research (or Clinical Trial), Causes, Treatment, Prevention, Diagnosis (Exams and Tests), Prognosis, Complications, Symptoms, Inheritance, Susceptibility, Genetic changes, Frequency, Considerations, Contact a medical professional, Support Groups etc.

Examples:

- What is the outlook for DISEASE?
- How many people are affected by DISEASE?
- When to contact a medical professional about DISEASE?
- Who is at risk for DISEASE?
- Where to find support for people with DISEASE?
- What are the genetic changes related to that DISEASE?
- Is the DISEASE inherited?
- What research (or clinical trial) is being done for DISEASE?
- What are the treatments for that DISEASE?

Here is a bar chart of the dataset for factoid type question answering. The chart is for question type and we can see the lines for every question type is equal in number. Because here we used a balanced dataset.

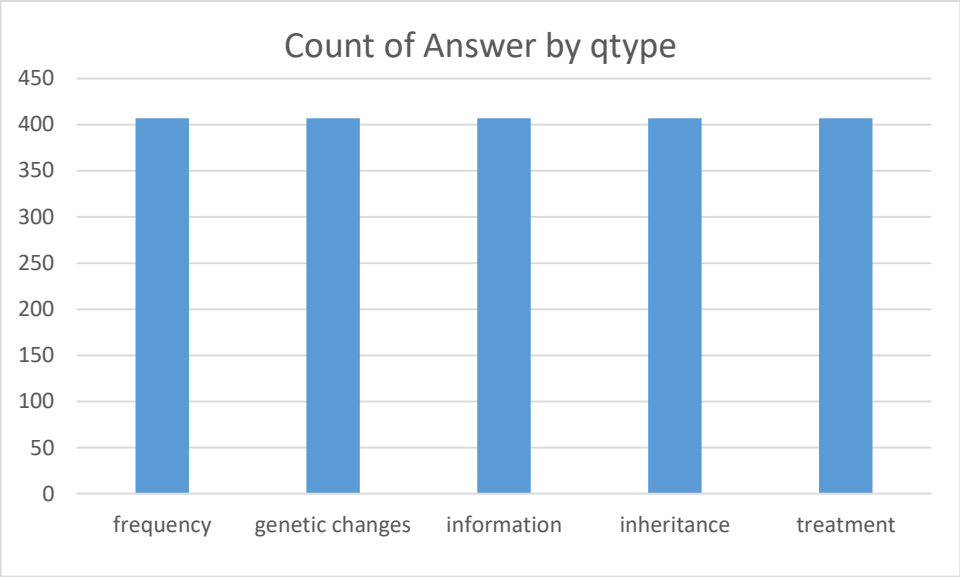


Figure 4.2: Total Occurrence of each significance in dataset

To create a set of question-answer pairs, we used 12 trusted websites. The list includes 37 categories of questions (e.g. Treatment, Diagnosis, Side Effects) related to illnesses, medications, examinations, and other medical institutions. In the 4 MedlinePlus sets, we introduced the query target category (Disease, Medication or Other). Any other collection is about illnesses. Concerning the management of the disease, the symptoms of the disease and the diagnosis of the disease, there are various kinds of questions. The following paper explains the set, the construction process, as well as its use and evaluation within a system of answering medical questions.

For each website, we extracted the free text of each article as well as the synonyms of the article focus (topic). These resources and their brief descriptions are provided below:

1. National Cancer Institute (NCI)[4]: We extracted free text from 116 articles on various cancer types (729 QA pairs). We manually restructured the content of the articles to generate complete answers (e.g. a full answer about the treatment of all stages of a specific type of cancer). Figure 2 presents examples of QA pairs generated from a NCI article.

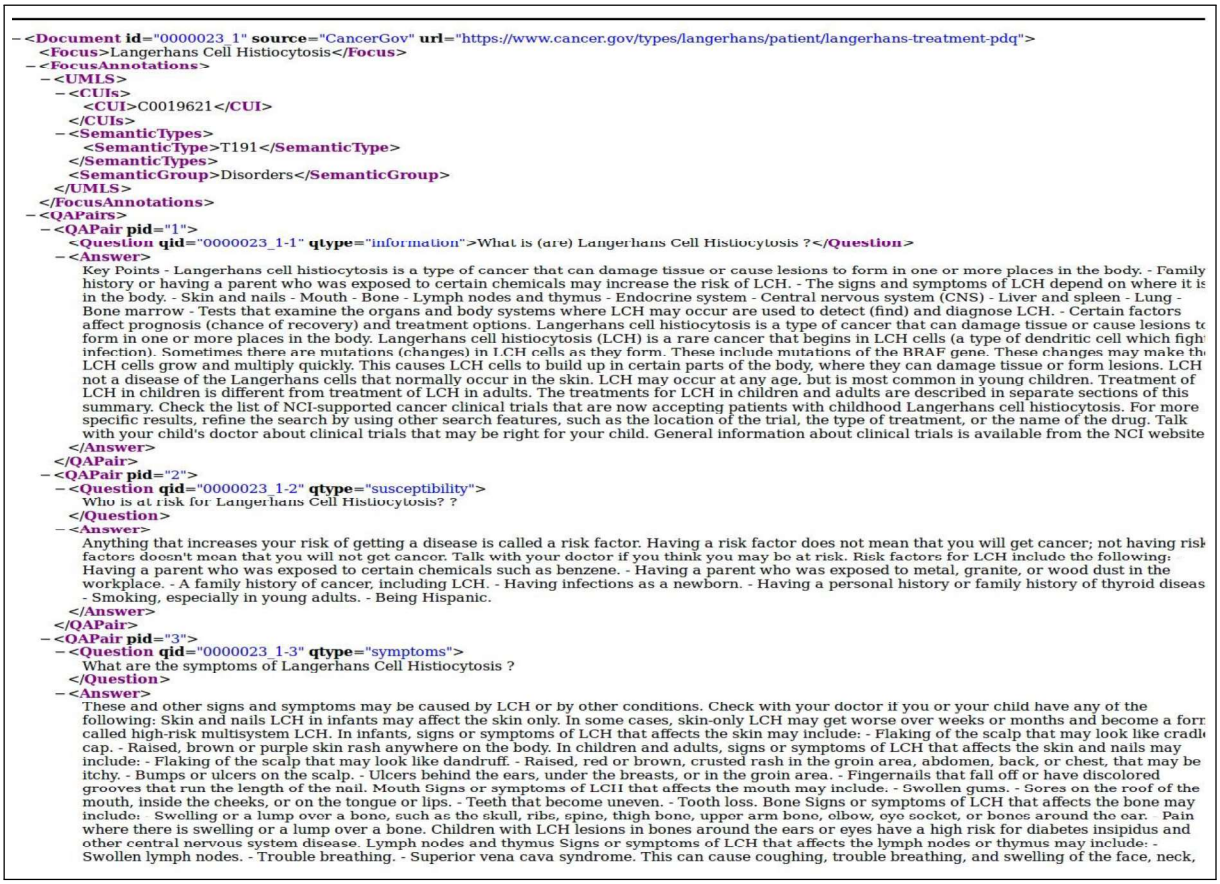
2. Genetic and Rare Diseases Information Center (GARD)[5]: This resource contains information about various aspects of genetic/rare diseases. We extracted all disease question/answer pairs from 4,278 topics (5,394 QA pairs).

[4] https://www.cancer.gov/types

[5] https://rarediseases.info.nih.gov/diseases

3. Genetics Home Reference (GHR)[6]: This NLM resource contains consumer-oriented information about the effects of genetic variation on human health. We extracted 1,099 articles about diseases from this resource (5,430 QA pairs).

4. MedlinePlus Health Topics[7]: This portion of MedlinePlus contains information on symptoms, causes, treatment and prevention for diseases, health conditions and wellness issues. We extracted the free texts in summary sections of 981 articles (981 QA pairs).



<Document id="0000023_1" source="CancerGov" url="https://www.cancer.gov/types/langerhans/patient/langerhans-treatment-pdq">
<Focus>Langerhans Cell Histiocytosis</Focus>
<FocusAnnotations>
<UMLS>
<CUIs>
<CUI>C0019621</CUI>
</CUIs>
<SemanticTypes>
<SemanticType>T191</SemanticType>
</SemanticTypes>
<SemanticGroup>Disorders</SemanticGroup>
</UMLS>
</FocusAnnotations>
<QAPairs>
<QAPair pid="1">
<Question qid="0000023_1-1" qtype="information">What is (are) Langerhans Cell Histiocytosis ?</Question>
<Answer>
Key Points - Langerhans cell histiocytosis is a type of cancer that can damage tissue or cause lesions to form in one or more places in the body. - Family history or having a parent who was exposed to certain chemicals may increase the risk of LCH. - The signs and symptoms of LCH depend on where it is in the body. - Skin and nails - Mouth - Bone - Lymph nodes and thymus - Endocrine system - Central nervous system (CNS) - Liver and spleen - Lung - Bone marrow - Tests that examine the organs and body systems where LCH may occur are used to detect (find) and diagnose LCH. - Certain factors affect prognosis (chance of recovery) and treatment options. Langerhans cell histiocytosis is a type of cancer that can damage tissue or cause lesions to form in one or more places in the body. Langerhans cell histiocytosis (LCH) is a rare cancer that begins in LCH cells (a type of dendritic cell which fights infection). Sometimes there are mutations (changes) in LCH cells as they form. These include mutations of the BRAF gene. These changes may make the LCH cells grow and multiply quickly. This causes LCH cells to build up in certain parts of the body, where they can damage tissue or form lesions. LCH is not a disease of the Langerhans cells that normally occur in the skin. LCH may occur at any age, but is most common in young children. Treatment of LCH in children is different from treatment of LCH in adults. The treatments for LCH in children and adults are described in separate sections of this summary. Check the list of NCI-supported cancer clinical trials that are now accepting patients with childhood Langerhans cell histiocytosis. For more specific results, refine the search by using other search features, such as the location of the trial, the type of treatment, or the name of the drug. Talk with your child's doctor about clinical trials that may be right for your child. General information about clinical trials is available from the NCI website
</Answer>
</QAPair>
<QAPair pid="2">
<Question qid="0000023_1-2" qtype="susceptibility">
Who is at risk for Langerhans Cell Histiocytosis? ?
</Question>
<Answer>
Anything that increases your risk of getting a disease is called a risk factor. Having a risk factor does not mean that you will get cancer; not having risk factors doesn't mean that you will not get cancer. Talk with your doctor if you think you may be at risk. Risk factors for LCH include the following: Having a parent who was exposed to certain chemicals such as benzene. - Having a parent who was exposed to metal, granite, or wood dust in the workplace. - A family history of cancer, including LCH. - Having infections as a newborn. - Having a personal history or family history of thyroid disease - Smoking, especially in young adults. - Being Hispanic.
</Answer>
</QAPair>
<QAPair pid="3">
<Question qid="0000023_1-3" qtype="symptoms">
What are the symptoms of Langerhans Cell Histiocytosis ?
</Question>
<Answer>
These and other signs and symptoms may be caused by LCH or by other conditions. Check with your doctor if you or your child have any of the following: Skin and nails LCH in infants may affect the skin only. In some cases, skin-only LCH may get worse over weeks or months and become a form called high-risk multisystem LCH. In infants, signs or symptoms of LCH that affects the skin may include: - Flaking of the scalp that may look like cradle cap. - Raised, brown or purple skin rash anywhere on the body. In children and adults, signs or symptoms of LCH that affects the skin and nails may include: - Flaking of the scalp that may look like dandruff. - Raised, red or brown, crusted rash in the groin area, abdomen, back, or chest, that may be itchy. - Bumps or ulcers on the scalp. - Ulcers behind the ears, under the breasts, or in the groin area. - Fingernails that fall off or have discolored grooves that run the length of the nail. Mouth Signs or symptoms of LCH that affects the mouth may include: - Swollen gums. - Sores on the roof of the mouth, inside the cheeks, or on the tongue or lips. - Teeth that become uneven. - Tooth loss. Bone Signs or symptoms of LCH that affects the bone may include: Swelling or a lump over a bone, such as the skull, ribs, spine, thigh bone, upper arm bone, elbow, eye socket, or bones around the ear. Pain where there is swelling or a lump over a bone. Children with LCH lesions in bones around the ears or eyes have a high risk for diabetes insipidus and other central nervous system disease. Lymph nodes and thymus Signs or symptoms of LCH that affects the lymph nodes or thymus may include: - Swollen lymph nodes. - Trouble breathing. - Superior vena cava syndrome. This can cause coughing, trouble breathing, and swelling of the face, neck,

Figure 4.3: Examples of QA pairs generated from an article about Langerhans Cell Histiocytosis (NCI)

[6] https://ghr.nlm.nih.gov

[7] https://medlineplus.gov/healthtopics.html

In complex type question answering the dataset contains total 4120 number of disease related questions which has 2 columns and we can get a brief answer by asking the medical related question. Our source for this dataset is biomedical question answering (QA) dataset collected from MedQuAD_GHR_QA. We update it manually in notepad and import in code as a text file.

In a recent research paper Recognizing question entailment between a given user question and all questions in a large collection is not practical for real-time QA systems. Therefore, they first filter the questions of the MedQuAD dataset with an IR method to retrieve candidate questions, then classify them as entailed (or not) by the user/test question. Based on the positive results of the combination method tested on SemEval-cQA data ("Results of RQE Approaches" section), they adopted a combination method to merge the results obtained by the search engine and the RQE scores.

Total column in complex type dataset are given below:

What is (are) abdominal wall defect?	An opening in the abdomen through which various abdominal organs can protrude.
What are the treatments for abdominal wall defect?	Diagnostic Tests-Drug Therapy-Surgery and Rehabilitation-Genetic Counseling-Palliative Care.
What is (are) Osteoporosis ?	A Bone Disease that thins and weakens the bones become fragile and break easily.
Who is at risk for Osteoporosis?	Women are at higher risk for osteoporosis than men.
What are the symptoms of Osteoporosis ?	Fractures-A Possible Warning Sign Osteoporosis does not have any symptoms until a fracture occurs
How to diagnose Osteoporosis ?	Taking glucocorticoid medications such as prednisone, cortisone, or dexamethasone for 2 months o
What are the treatments for Osteoporosis ?	Although there is no cure for osteoporosis, it can be treated.
what research is being done for Osteoporosis ?	Scientists are pursuing a wide range of basic and clinical studies on osteoporosis.
How to prevent Osteoporosis ?	Preventing falls is a special concern for men and women with osteoporosis.
What is (are) Paget's Disease of Bone ?	Enlarged and Misshapen Bones disease of bone causes affected bones to become enlarged and mis
What are the symptoms of Paget's Disease of Bone ?	Bone pain,misshapen bones,fractures,osteoarthritis of the joints adjacent to bone affected by the
How to diagnose Paget's Disease of Bone ?	Diagnostic Tests X-rays are almost always used,but the disease may be discovered using one of thre
What are the treatments for Paget's Disease of Bone ?	Although there is no cure for disease of bone, it is treatable.
What are the complications of Paget's Disease of Bone ?	Arthritis,headaches,hearing loss,and nervous system problems,depending on which bones are affe
What is (are) Parkinson's Disease ?	A Brain Disorder is a brain disorder that leads to shaking, stiffness, and difficulty with walking

Figure 4.4: Total column in complex type dataset

CHAPTER 5

IMPLEMENTATION

Actually, two types of approaches have been focused and implemented in this work. . Here we prepare two types of question answering system. One is factoid type approach and the other one is complex type approach.

In case of factoid approach, we have mostly concentrated on our proposed approach which is Convolutional Neural Network Gated recurrent unit (GRUs), bidirectional Long Short Term Memory cells (LSTMs), Long Short Term Memory cells (LSTMs) ,Recurrent Neural Network (RNN). Meanwhile, for complex type purpose we have conducted Encoder-Decoder Sequence to Sequence Model without attention, & Encoder-Decoder Sequence to Sequence Model with attention.

5.1 Factoid type question answering

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 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 120-word limit. Finally, to generate the summary, we concatenate the list of selected sentences to create the final answer.

In the TREC-10 IBM system (Ittycheriah, Franz, and Roukos 2001) a maximum-entropy model was used for automatically learning features weights and combining multiple features. In the maximum entropy model, the probability of the event that an answer candidate A is correct given question Q is defined as:

$$P(x = C|A, Q) = \frac{\exp \quad f(x = C,A,Q) \cdot \sim \theta}{\exp f(x=C,A,Q)\theta + \exp f(x=W,A,Q)\theta}$$

Here x is the judgment of A. It is a binary random variable that can take on values of either C (correct) or W (wrong). ~ f is a feature vector and ~ θ are the feature weights. Inputs to their model were the question and the answer string. A total of 31 features were used in the system. They were mainly identity features over named-entity types and certain pre-defined syntactic patterns.

Factoid Multiclass classification (Pseudocode):

START

Input: input sentence {Given a sequence of inputs (x1, . . . , xT)}
Output: target class level {level of outputs (y1,.., yT)}

1. Preprocessing:

- convert xml file into data frame
- encode categorical to numerical
- clean the data
- tokenize the text
- apply stop word removal

2. Train test split:

3. define Custom word2vec model

Model = gensim.models.Word2Vec (
Pass the question,
Set the dimensionality of word vector,
Set the window size,
set number of cpu core as workers size,
define the minimum count words for model
Set sg <- 0 for cbow)

4. Save word2vec model

5. Embedding layer

- read the save model text file
- Initialize embedding matrix
- Define the embedding_layer
  emb_layer = Embedding (
  define the vocabulary size,
  Set Embedding dimensions,
  Weights <- embedding matrix,
  Input documents length,
  trainable parameters as false)

6. define LSTM model

- first create the sequential model
- add the fist embedding layer
- Add the LSTM layer and set return_sequences as false
- Add Fully connected dense layer with 'relu' activation function
- Add dropout layer for regularization
- Add dense output layer with softmax activation
- Compile the model

7. LSTM mode training

History = model1.fit(
pass X_train data,
pass Y_train data,
Set the epochs as 10,
Set the batch_size as 32,
Define the validation_data from test dataset,
Set verbose as 1)

8. Make predictions from test dataset

END

5.2 Complex type question answering

A simple strategy for general sequence learning is to map the input sequence to a fixed-sized vector using one RNN, and then to map the vector to the target sequence with another RNN (this approach has also been taken by Cho et al. .However, the Long Short-Term Memory (LSTM) is known to learn problems with long range temporal dependencies, so an LSTM may succeed in this setting. The goal of the LSTM is to estimate the conditional probability p(y1,...,yT ′|x1,...,xT) where (x1,...,xT) is an input sequence and y1,...,yT ′ is its corresponding output sequence whose length T′ may differ from T. The LSTM computes this conditional probability by first obtaining the fixed dimensional representation v of the input sequence (x1,...,xT) given by the last hidden state of the LSTM, and then computing the probability of y1,...,yT ′ with a standard LSTM-LM formulation whose initial hidden state is set to the representation v of x1,...,xT

P (y1,...,yT ′|x1,...,xT) = $\prod_{t=1}^{T'} p(y1|v, y1 \ldots\ldots, yt-1)$

In this equation, each p (yt|v,y1,...,yt−1) distribution is represented with a softmax over all the words in the vocabulary. We use the LSTM formulation from Graves (R. D. Burke, K. J. Hammond, V. Kulyukin , S. L. Lytinen , N. Tomuro, and S. Schoenberg, 1997). Note that we require that each sentence ends with a special end-of-sentence symbol “<EOS>”, which enables the model to define a distribution over sequences of all possible lengths. The overall scheme is outlined in figure 1, where the shown LSTM computes the representation of “A”, “B”, “C”, “<EOS>” and then uses this representation to compute the probability of “W”, “X”, “Y”, “Z”, “<EOS>”.

For sequence to sequence encoder decoder with attention a new model architecture, we define each conditional probability in Eq. (2) as: p(yi|y1,...,yi−1,x) = g(yi−1,si,ci), (4) where si is an RNN hidden state for time i, computed by si = f(si−1,yi−1,ci). It should be noted that unlike the existing encoder–decoder approach (seeEq .(2)),here the probability is conditioned on a distinct context vector ci for each target word yi.

The context vector ci depends on a sequence of annotations (h1,⋯ ,hTx) to which an encoder maps the input sentence.

Each annotation hi contains information about the whole input sequence with a strong focus on the parts surrounding the i-th word of the input sequence. We explain in detail how the annotations are computed in the next section. The context vector ci is, then, computed as a weighted sum of these annotations hi:

$ci = \sum_{j=1}^{Tx} \alpha ijhj$

The weight αij of each annotation hj is computed by

$\alpha ij = \frac{\exp(eij)}{\sum_{k=1}^{Tx} \exp(eik)}$

Where eij = a (si−1,hj) is an alignment model which scores how well the inputs around position j and the output at position i match. The score is based on the RNN hidden state si−1 (just before emitting yi, Eq. (4)) and the j-th annotation hj of the input sentence. We parametrize the alignment model as a feed forward neural network which is jointly trained with all the other components of the proposed system.

Encoder decoder without attention (Pseudocode):

START

Input: input sentence {Given a sequence of inputs(x1, . , xT)}

Output: target sentence {sequence of outputs (y1, . . , yT)}

1. Preprocessing

2. Create the dictionary of unique source and target words to vector and vice-versa

- find all the source and target words and sort them
- Find maximum sentence length in the source and target data
- create a word to index(word2idx) for source and target
- create dictionary for index to word for source and target vocabulary

3. Split the dataset into train and test data

4. Create the data for training the encoder-decoder model

```
Function generate_batch(X, y, batch_size)
while true
        for each j in range(0,,len(x,batch_size)
                Initialize encoder input,decoder input,decoder output
   For i = 0,(input_text, target_text) in enumerate
        For t = 0, word in input_text
                encoder_input_data[i, t] <- source_word2idx[word]

  if t less than len(target_text.split())-1
        decoder_input_data[i, t] <- target_word2idx[word]
```

If t greater than zero decoder target, data will be ahead by one timestep
decoder_target_data[i, t - 1, target_word2idx[word]] <- 1
yield encoder_input_data, decoder_input_data and
decoder_target_data

5. Build the encoder using Embedding and LSTM layers

- pass the input through the input layer.
- first hidden layer will be the embedding layer.
- Create the LSTM layer and only set return_state to True
- discard the encoder_output and preserve the hidden state and cell state only

6. Build the decoder using Embedding and LSTM layers

- embedding is again the first hidden layer in the decoder
- LSTM layer will return output sequences as well as the internal states.
- apply a softmax activation to the Dense layer
- generate the decoder outputs

7. Compile and train the model

- Define the model that takes encoder and decoder input to output decoder_outputs
- compile the model using "rmsprop" optimizer and categorical_crossentrop

8. Define the inference model

9. Make predictions from on the Test dataset

END

Encoder decoder with attention (Pseudocode):

The steps of attention mechanism is similar with encoder decoder model except the Encoder, Attention layer and Decoder. The attention mechanism is an implementation of 'Bahdanau Attention'.

START

1. Preprocessing

2. Create the dictionary of unique source and target words to vector and vice-versa

- find all the source and target words and sort them
- Find maximum sentence length in the source and target data
- create a word to index(word2idx) for source and target
- create dictionary for index to word for source and target vocabulary

3. Split the dataset into train and test data

4. Build the encoder

- pass the input through the input layer.
- the first hidden layer will be the embedding layer.
- Create the first LSTM layer

- Create the second LSTM layer
- Create the third LSTM layer
- set return_sequences=True and return_state to True for all LSTM layer

2. Build the decoder

- create an input layer for the decoder_input
- embedding is again the first hidden layer in the decoder
- Create LSTM layer
- apply a softmax activation to the Dense layer
- generate the decoder outputs

3. Build the Attention Layer

END

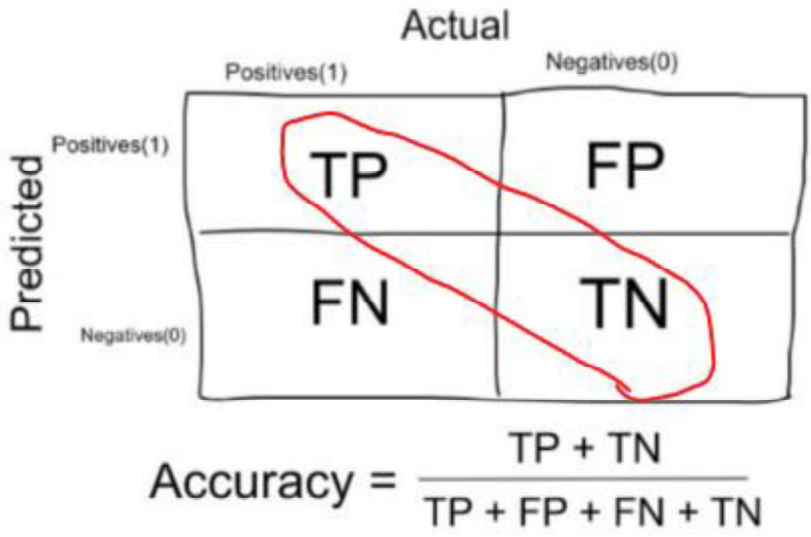
CHAPTER 6

RESULT ANALYSIS

In this section performance of all the considered approaches have been evaluated and analyzed using various performance measurements including accuracy, precision , recall etc. moreover, accuracy have also been illustrated theoretically and graphically for comparative study. Additionally precision as well as recall by the proposed approach along with all the considered approaches have been conducted. All these performance evaluation metrics have been concentrated in order to avoid overfitting problem and finding superior approach among all the approaches applied here in this work. Comparative studies of all the approaches are as follows:

6.1 Performance Analysis of factoid question answering

The most common performance measurement metric for machine learning approaches is accuracy comparison. Usually, accuracy is computed by summing the values of true positive and true negative divided by total instances in the test set which can be formulated as follows:



$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Here, TP, TN, FP, FN are respectively True Positive, True Negative, False Positive and False Negative.

Also we calculate precision, recall which pattern recognition, information retrieval,

Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances, while recall (also known as sensitivity) is the fraction of the total amount of relevant instances that were actually retrieved. Both precision and recall are therefore based on an understanding and measure of relevance.

Recall in this context is also referred to as the true positive rate or sensitivity, and precision is also referred to as positive predictive value (PPV); other related measures used in classification include true negative rate and accuracy. True negative rate is also called specificity.

The formula for precision, recall given below-

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Comparison of the accuracies, precisions, recalls of all the approaches are mentioned in table 6.1.

Table 6.1: Accuracies, precisions, recalls of the algorithms

Name of the Algorithm	Depicted Accuracy (%)	Depicted Precision (%)	Depicted Recall (%)
LSTM	95.19	95.50	94.84
Bidirectional LSTM	97.31	97.44	97.15
GRU	83.63	86.18	81.24
Simple RNN	83.70	85.50	82.00
CNN	94.98	95.38	94.52

Our proposed bidirectional LSTM reveals better accuracy, precision, recall compared to the other alternatives. Accuracy, precision, recall comparison has been graphically represented in Figure 6.1.

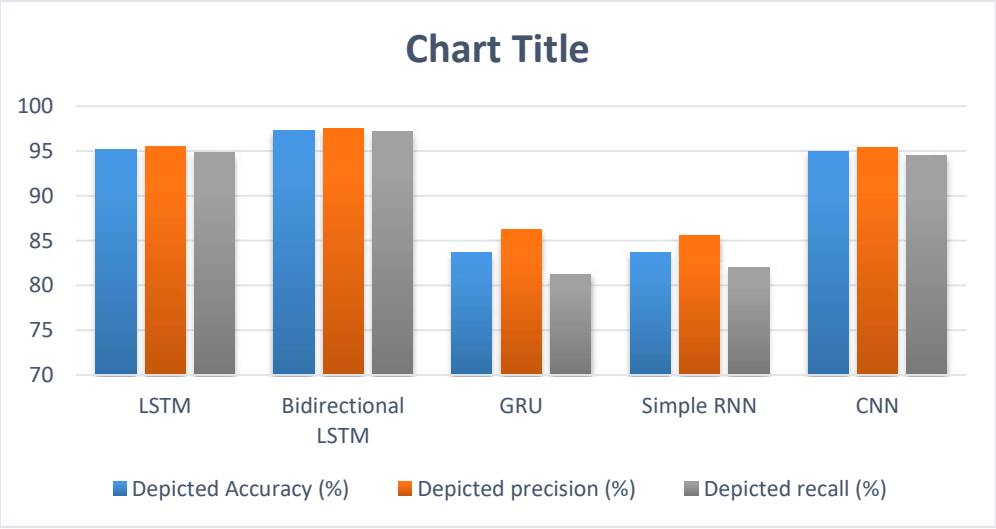


Figure 6.1: Accuracy, precision, recall comparison of the algorithms

Performance comparison of the proposed approach with recent paper

We compare our work with most recent biomedical question answering research paper and in case of accuracy, precision, recall our proposed approach seems to be more promising as well as effective. However, memory requirement of our proposed approach is quite higher compared to recent work but overall demonstration states that our proposed approach reveals the best performance while detecting biomedical question answering. Graphical representation of Table 6.2 also depicts the same in Figure 6.3.

Table 6.2: Performance comparison of the proposed approach

Name of the Algorithm	Depicted Accuracy (%)	Depicted Precision (%)	Depicted Recall (%)
Proposed Approach	97.31	97.44	97.15
Recent paper(Mohasseb∗, et al. 2018)	95.5	86.1	90.80

We can clearly see that our proposed work has higher rate of accuracy, precision, recall. So Graphical representation of Table 6.2 also depicts the same in Figure 6.2.

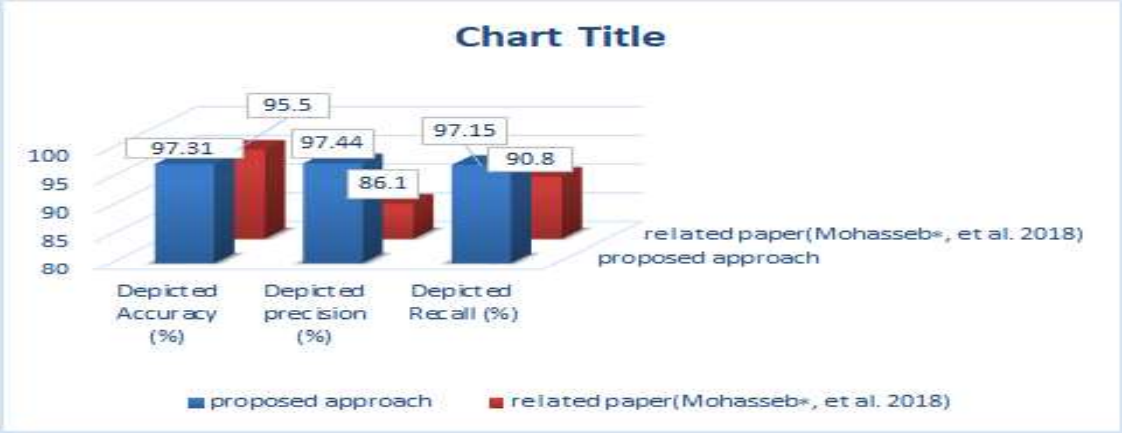


Figure 6.2: Comparison of proposed work and related paper work

6.2 Performance Analysis of complex question answering

We have implemented encoder decoder as well as attention model for generation of complex type question answering. Moreover, for being confident on the outcome of our implemented approach we have also conducted manual generation of rules from the dataset. Table 6.3 represents the encoder decoder and attention model on our dataset. We have extracted similar rules from previously constructed paper applying encoder decoder except from few exceptions. Our proposed work for two types of approach are given below as comparison.

Table 6.3: Accuracies, precisions, recalls of the algorithms

Name of the Algorithm	Depicted Accuracy (%)	Depicted Precision (%)	Depicted Recall (%)
Encoder decoder (1st approach)	99.67	99.68	99.67
Attention (2nd approach)	37.25	47.57	99.49

Our proposed encoder decoder reveals better accuracy, precision, recall compared to its alternative. Accuracy, precision, recall comparison has been graphically represented in Figure 6.3.

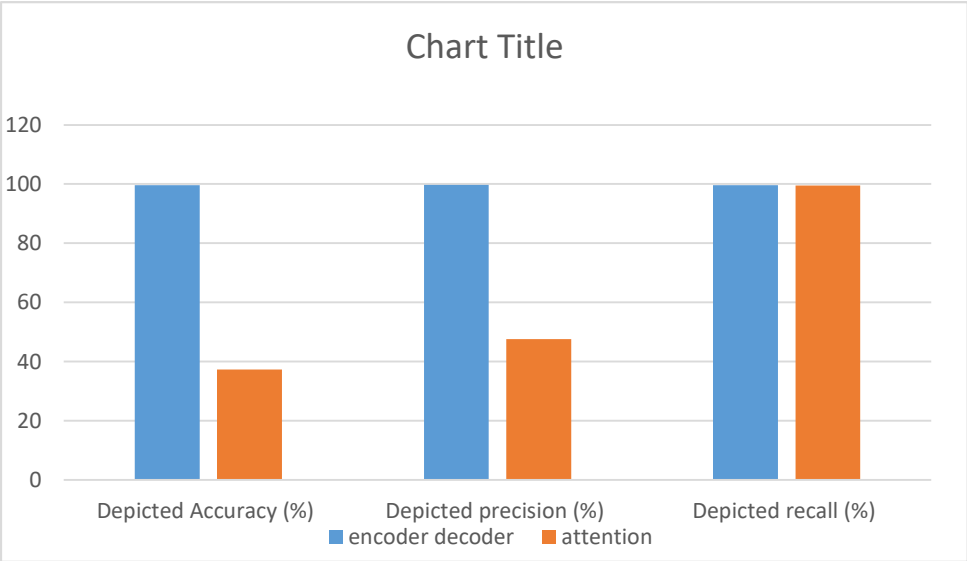


Figure 6.3: Accuracy, precision, recall comparison of all algorithm

Performance comparison of the proposed approach with recent paper

We compare our complex type work with most recent biomedical question answering research paper and in case of accuracy, precision, recall our proposed approach seems to be more promising as well as effective. The comparison table given below in table 6.4

Table 6.4: Performance comparison of the proposed approach

Name of the Algorithm	Depicted Accuracy (%)	Depicted Precision (%)	Depicted Recall (%)
Encoder decoder 1st Approach	99.67	99.68	99.67
Attention 2nd approach	37.25	47.57	99.49
Recent paper (Asma Ben Abacha*,et al. 2019)	80.57	70.29	72.10

We can see here our proposed work encoder decoder which is without attention has higher rate of accuracy, precision, recall than related paper work whereas encoder decoder with attention has less accuracy, precision, recall percentage than related paper work . So Graphical representation of Table 6.4 also depicts the same in Figure 6.4

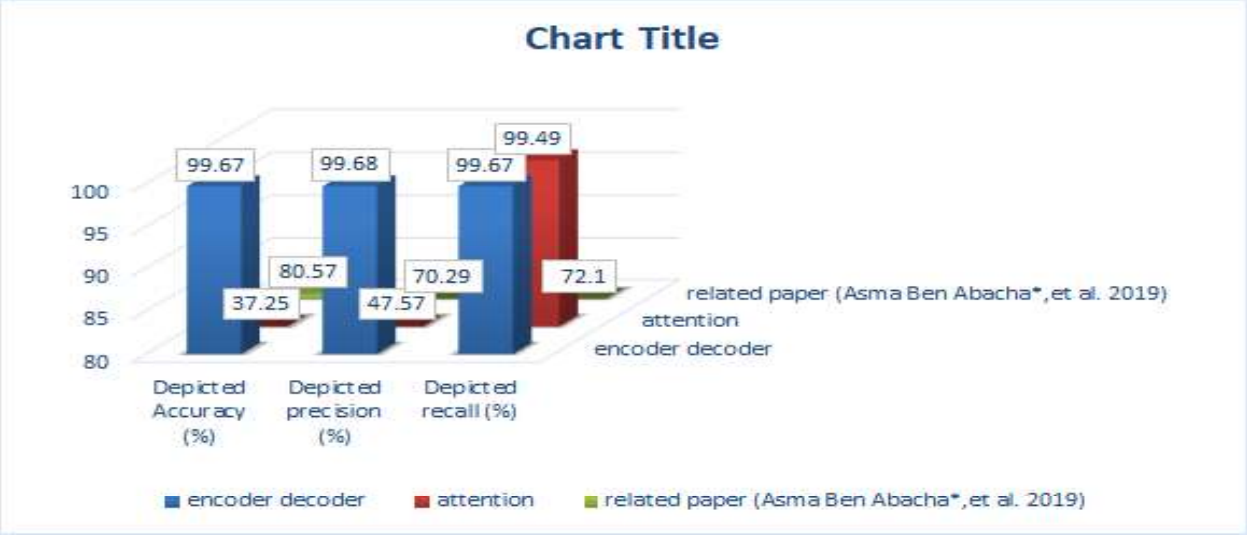


Figure 6.4: Comparison of proposed work and related paper work

CHAPTER 7

CONCLUSION & FUTURE WORK

This paper presents a new and intuitive method for answering complex temporal questions using an embedded current factoid-based Q.A. system. Lots of research have been already conducted to find significant information to make a system about biomedical. In spite of all the invention no automated mechanism has been developed or proposed yet to detect or find any pattern for finding the exact answer in this era of big data. Manual procedure is not feasible any more. Therefore, motive perspective of this research is to conduct an experimental analysis to evaluate the performance of our proposed approach.

In this paper, we carried out an empirical study of machine learning and deep learning methods for medical question answering using several datasets. We developed a QA system to answer medical questions using existing question-answer pairs. We built and shared a collection of 5K medical question answer pairs. The proposed approach can be applied and adapted to open-domain as well as specific-domain QA. Deep learning models achieved interesting results on open-domain and clinical datasets, but obtained a lower performance on consumer health questions. We will continue investigating other network architectures including transfer learning, as well as creation of a large collection of consumer health questions for training to improve the performance of models.

We present MedQuAD, a novel dataset aimed at biomedical research question answering using yes/no/maybe, where complex quantitative reasoning is required to solve the task. MedQuAD has substantial automatically collected instances as well as the largest size of expert annotated yes/no/maybe questions in biomedical domain. We provide a strong baseline using multi-phase fine-tuning of BioBERT with long answer as additional supervision, but it's still much worse than just single human performance. Generally, MedQuAD can serve as a benchmark for testing scientific reasoning abilities of machine reading comprehension models.

We analyzed that better generation of multi-label question classification corpus not only improved the performance of question classification but also has a positive impact on the overall QA system. This paper has specifically focused on a process of factoid type answer and decomposition of complex temporal questions and on its evaluation on a temporal question corpus. In the future, our work is directed to fine tune this system and increase its capabilities towards processing questions of higher complexity.

Moreover there are 5k dataset in total which doesn't cover the whole disease related question all over the world. A limitation of current flow is that the system requires explicit factoid and list type identification. That is the limitation of our work, and in future we will increase the number of question-answer so that it can cover most of the disease occurs in the world. In real-world applications, such information might not be available. To overcome this issue, we can use a classifier to classify questions in factoid and list type. Also in future recent sequence-to-sequence model transformer, bart can be used to implement for complex question answering system
Finally, MedQuAD collection requires further refinements, and human evaluation of the dataset can further improve the performance of biomedical question answering.

References

N. G. VasuSharma*, "BioAMA:TowardsanEndtoEndBioMedicalQuestionAnsweringSystem," *Proceedings of the BioNLP 2018 workshop,* p. 109–117 , 2018.

A. DinaDemner-Fushman, "A QUESTION-ENTAILMENT APPROACH TO QUESTION ANSWERING," *A PREPRINT,* 2019.

QiaoJin, "PubMedQA:ADatasetforBiomedicalResearchQuestionAnswering," 2019.

VasuSharma*,NitishKulkarni*,SrividyaPranaviPotharaju*, GabrielBayomi*,EricNyberg,TerukoMitamura , "BioAMA:TowardsanEndtoEndBioMedicalQuestionAnsweringSystem," *Proceedings of the BioNLP 2018 workshop,* p. 109–117 , 2018.

“What is convolutional neural network? - Definition from WhatIs.com.”

“Understanding of Convolutional Neural Network (CNN) — Deep Learning.” [Online]. Available: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neuralnetwork-cnn-deep-learning-99760835f148. [Accessed: 30-Mar-2019].

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 EmpiricalMethodsinNaturalLanguageProcessing (EMNLP). Association for Computational Linguistics.

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

W. An, Q. Chen, W. Tao, J. Zhang, J. Yu, Y. Yang, Q. Hu, L. He, and B. Li. ECNU at 2017 LiveQA track: Learning question similarity with adapted long short-term memory networks. In Proceedings of The Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017, 2017.

R. D. Burke, K. J. Hammond, V. Kulyukin, S. L. Lytinen, N. Tomuro, and S. Schoenberg. Question answering from frequently asked question files: Experiences with the faq finder system. AI magazine, 18(2):57, 1997.

V. V. Datla, T. R. Arora, J. Liu, V. Adduru, S. A. Hasan, K. Lee, A. Qadir, Y. Ling, A. Prakash, and O. Farri. Open domain real-time question answering based on asynchronous multiperspective context-driven retrieval and neural paraphrasing. In Proceedings of The Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017, 2017. URL https://trec.nist.gov/pubs/trec26/papers/prna-QA.pdf.

V. Jijkounand, M. de Rijke. Retrieving answers from frequently asked questions pages on the web. In Proceedings ofthe2005ACMCIKMInternationalConferenceonInformationandKnowledgeManagement, Bremen,Germany, October 31 - November 5, 2005, pages 76–83, 2005. URL http://doi.acm.org/10.1145/1099554. 1099571.

J. Luo, G.-Q. Zhang, S. Wentz, L. Cui, and R. Xu. Simq: Real-time retrieval of similar consumer health questions. J Med Internet Res, 17(2):e43, Feb 2015. URL https://doi.org/10.2196/jmir.3388.

D. A. Lindberg, B. L. Humphreys, and A. T. McCray. The unified medical language system. Methods of Information in Medicine, 32:281–291, 1993.

D. Wang and E. Nyberg. CMU OAQA at TREC 2017 LiveQA: A neural dual entailment approach for question paraphrase identification. In Proceedings of The Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017, 2017. URL https://trec.nist.gov/pubs/trec26/ papers/CMU-OAQA-QA.pdf.

D.Demner-Fushman, W.J.Rogers,andA.R.Aronson. Metamaplite: an evaluation of a new java implementation of meta map. JAMIA, 24(4):841–844, 2017. URL https://doi.org/10.1093/jamia/ocw177.

D.Demner-Fushman, W.J.Rogers, and A.R. Aronson. Metamaplite: an evaluation of a new java implementation of meta map. JAMIA, 24(4):841–844, 2017. URL https://doi.org/10.1093/jamia/ocw177.

D. Wang and X. Tan, “Label-Denoising Auto-encoder for Classification with Inaccurate Supervision Information,” in 2014 22nd International Conference on Pattern Recognition, 2014, pp. 3648–3653.

M. Cassel and F. L. Kastensmidt, “Evaluating One-Hot Encoding Finite State Machines for SEU Reliability in SRAM-based FPGAs,” in 12th IEEE International On-Line Testing Symposium (IOLTS’06), 2006, pp. 139–144.

(Dzmitry Bahdanau, NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE, 2016)

Asma Ben Abacha* & Dina Demner-Fushman, 2019. A question-entailment approach to question answering. *Ben Abacha and Demner-Fushman BMC Bioinformatics,* p. 20:511.

Mohasseb∗, A., Bader-El-Den, M. & Cocea, M., 2018. Classification of factoid questions intent using grammatical features. *The Korean institute of communication and information sciences,* pp. 239-242.