The Pennsylvania State University
The Graduate School
College of Information Sciences and Technology

CONVOLUTIONAL NEURAL NETWORK AND QUESTION GENERATION BASED APPROACHES TO SELECT BEST ANSWERS FOR NON-FACTOID QUESTIONS

A Thesis in
Information Sciences and Technology
by
Mukund Srinath

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2019

The thesis of Mukund Srinath was reviewed and approved* by the following:

Dongwon Lee
Associate Professor of Information Sciences and Technology
Thesis Advisor

C. Lee Giles
David Reese Professor of Information Sciences and Technology

Ting-Hao Huang
Assistant Professor of Information Sciences and Technology

Mary Beth Rosson
Professor of Information Sciences and Technology
Director, Graduate Programs, Information Sciences and Technology

*Signatures are on file in the Graduate School.

Abstract

The answer selection task involves selecting the most appropriate answer for a question given a list of answers for the question. The problem tackled in this thesis is a subset of the answer selection task and concentrates on answer selection for non-factoid questions. Non-factoid questions are ones which cannot be answered in a word or phrase. They usually have long answers which do not share a lot of common words with the question.

Two methods are discussed in this thesis. First is a Convolutional Neural Network method which creates distributed vector representations for the questions and answers, and learns to minimize the distance (in vector space) between questions and their most-appropriate answers. Second is a question generation approach in which a Seq2Seq model is trained to generate questions from the given answers and the previously discussed CNN is then used to create vector representations of the questions minimizing the distance between the true question and the question generated by the true answer.

Answer selection is treated as an information retrieval task and the precision@1 and mean reciprocal rank scores are reported. Evaluation is carried out on two datasets, the Yahoo Webscope L6 which is a standard dataset for the answer selection task and the Library corpus - a custom dataset created by collecting student responses to information literacy questions to earn online micro-credentials. The performance of the CNN model shows improvement in precision@1 scores over state of the art models on the library corpus and shows comparable performance on the Yahoo Answers corpus. The results obtained using the question generation approach are promising and suggest steps for future work.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Background

In the answer selection task, the objective is to find the best answer among a set of answers for a given question. The problem can be divided into answer selection for factoid questions and non-factoid questions. Factoid questions are ones which can be answered either in a word or a phrase. There is usually a single right answer, although this answer might change from time to time. For example, if we consider the question 'What is the capital of India?', it can only have one possible answer. This task involves finding the answer to the question in any given text. These factoid questions usually ask 'who', 'when', 'where' and 'what' questions and the answers tend to follow the question i.e. there is significant word overlap between the question and answer [1]. The answer to the question asked above would be 'The capital of India is New Delhi.' Here we can see that there is significant word overlap between the question and answer.

Non-factoid questions do not share these features of factoid questions. They ask 'how', 'comment', 'define' or 'explain' questions. The answers tend to be two or three sentences long. They have multiple ways in which the answer can be depicted. This is to say that the vocabulary used to answer these types of questions can be very varied [2]. Thus, there isn't usually a lot of word overlap between questions and answers, although they can be thought of as having a shared semantic space. The following example was taken from the Yahoo Webscope L6 collection. Question: 'How do you get rid of a beehive.' Answer: 'Call an area apiarist. They should be able to help you and would most likely remove them at no charge in exchange for the hive. The bees have value and they now belong to you.' This is a perfect example of a non-factoid question. The question and answer do not share a single word, but the answer correctly answers the question. Another possible answer for this question might be a way to get rid of the bees without calling an apiarist. But all possible answers and the question will be in the same semantic space.

1.2 Motivation

The three main applications of the answer selection task are in search engines, community question answering, and student evaluations. When we type a question in the Google search bar, sometimes we see that there is an answer box containing the answer to the question displayed before any of the other search results. This answer box is populated by selecting the most relevant answer from the list of answers available in the various links that Google provides. It is offered to help the user get a quick and immediate response to his or her question, and it usually contains a small extract of the text provided on the selected website. It is easy to see that many times, the answer provided falls short and also times when the answer box is left out altogether. Research in answer selection can significantly improve such results in the case of search engines. It should be noted here that the research explored in this thesis can not directly be applied to the selection of links in the case of a search engine. The answer to the user query would likely be buried in a paragraph of an article which deals with a lot of issues directly or indirectly related to the user query. Research to extract the correct sentences which relate to the query is separate from the approach followed here.

Community Question Answering sites are a rich source of questions and answers. High-quality comments on a question are provided by experts and the general public alike. They are a great source of information and are often frequented by people. These communities are usually self-maintaining with a few moderators who keep things in order [3]. A user posts a question, a few other users answer the question, and the user has the ability to select the best answer depending on the quality and usefulness. The answer selected by the asker, in turn, serves any other user who is searching for an answer to the same or a similar question. Sometimes other users can also vote on the usefulness of an answer so that the best answers get the most attention. But this happy flow works only in very active communities where users take an active part in maintaining the quality of answers. But in lesser known communities, questions are answered by many users, but the quality is not well maintained. Useful answers are sometimes buried under a heap of irrelevant and unhelpful answers. Automatically selecting appropriate answers to the question can thus be very helpful in these communities [4]. In other communities like Quora where answers can be both entertaining and helpful, answer selection can help drop answers which are not relevant to the question and therefore improve content quality.

I encountered one of the use cases of answer selection myself in student evaluations. The University Libraries at Penn State offers students the opportunity to earn online micro-credentials in information literacy skills. But as the number of students who signed up to take the badges increased, the load on the librarians who had to evaluate the non-objective type answers of the students also increased. A heavy load not only puts an extra burden on the evaluators, but students also do not receive their feedback on time. A classifier which is able to evaluate student answers will go a long way in easing the burden on evaluators. Thus, answer selection or classification, in this case, can play a major role in improving the efficiency of evaluators. A robust answer selection model will be able to improve the efficiency of evaluators in multiple arenas.

Chapter 2
Literature Review

2.1 Overview

The answer selection task can be divided based on the type of answers, i.e., factoid vs. non-factoid. As explained in the introduction, there is a significant difference between the two kinds of answers and the methods that work well in their retrieval. For factoid questions, it is observed that both linguistic and machine learning approaches work reasonably well. But for non-factoid questions, linguistic approaches are not very successful. This observation can be explained by the fact that there is low word overlap between the question and the answer for non-factoid question-answers. It can also be seen that retrieving non-factoid answers is the harder problem. Although the comparison of the results between the two methods is only somewhat appropriate, it can be seen that the overall performance of models in retrieving factoid answers is superior to the performance on non-factoid questions. The problem is compounded by the fact that there are very few non-factoid question-answer corpora to help train models better.

2.2 Datasets

One of the first domain independent datasets released for question answering was the TREC-QA dataset. It has 200 questions which are mostly factoid questions collected from about five hundred thousand news articles [5]. The TREC dataset was subsequently improved further to include non-factoid answers. The Yahoo Webscope L6 is another example of a comprehensive open domain question-answering corpus for non-factoid questions. It has more than four million question-answers collected from Yahoo Answers - a community question answering website [2]. The Insurance QA corpus contains questions and answers collected from the website 'Insurance Library.' It is the first non-factoid QA dataset in the insurance domain. The authors note that the questions are from real-world users and the answers are from experts with deep domain knowledge thus ensuring the quality of the corpus [6]. Although this is not a comprehensive list, these are most of the major datasets used for non-factoid question answering. There are a variety of machine comprehension and factoid question answering datasets. Machine comprehension is a sub-task of general question answering and involves the question answering task given a short paragraph from which the answer is to be extracted. WikiQA [7], CNN-DailyMail [8], SQuAD [9], NewsQA [10] and Google Natural Questions [11] are just a few datasets used for the machine comprehension task.

2.3 Factoid Question Answering

Chronologically, factoid question answering was first tackled using linguistic approaches followed by machine learning and deep learning approaches. We will first look at the methods used for factoid question answering because the improvement in the methods to solve factoid question answering also lead to improvements in non-factoid question answering. Punyakanok et al. used approximate tree matching and tree-edit-distance to compute a similarity score between the question and answer parse trees. This method works well for factoid question answers as they have significant word overlap. The edit distance in a tree is the minimum number of changes that need to be made to a tree to reach another structure [12].

Similarly, Shen et al. experimented with dependency tree kernels to compute similarity between parse trees. In this paper, the authors propose a pattern extraction method to extract the various relationships between proper answers and questions and also propose a tree kernel to match syntactic relation patterns. Finally, they use a maximum entropy model to rank the answers [13]. Bian et al. use textual, statistical, and user interaction features to rank answers for a given question. The authors concentrate on factoid questions in community question answering websites like Yahoo Answers and achieve state of the art results [1].

Iyyer et al. use recursive neural networks to reason over inputs by modeling textual compositionality. The authors introduce a model QANTA which outperforms multiple QA baselines [14]. Although the techniques used for factoid question answering are different from those used for non-factoid question answering, they follow a similar pattern as seen from the above literature. Next, we will see literature on non-factoid questions answering.

2.4 Non-Factoid Question Answering

2.4.1 Answer Ranking

One of the first works specifically relating to non-factoid answers was by Girju et al. who extract answers by searching for certain semantic structures like causation related to answers. The authors design an inductive learning approach to the automatic discovery of lexical and semantic constraints necessary in the disambiguation of causal relations [15].

Surdeanu et al. greatly improve the accuracy using a plethora of features. They consider four different kinds of features. Similarity features which relate to the similarity between questions and answers using tf-idf, translation features which treat answers as a translation of the question using a translation model, density features which compute the density of the same word sequences, answer spans, number of informative words etc., and collection features which convert the text into syntactic dependency chains or a bag of predicate-argument relations extracted using a semantic parser [16]. This work also uses the Yahoo QA dataset which is now one of the benchmark datasets used for non-factoid question answering tasks. They report a precision@1 score of 51.16. The use of linguistic features and extensive feature extraction helps improve the results but comes at the price of time. Each new dataset will have to be separately extensively analyzed before applying the feature extraction methods mentioned in the above paper.

Yu et al. introduce the novel idea of solving the task by using distributed vector representations and matching questions and answers based on semantic encoding [17]. This approach suggested by the authors starkly contrasts previous work which worked on handcrafted syntactic features. Much of the future work is built on this approach introduced by the authors. The authors explore their approach on TREC QA dataset and demonstrate state of the art performance in the answer selection task.

A lot of improvement is performance is seen with the use of deep learning models. In their paper, Feng et al. use Convolutional Neural Networks to solve the non-factoid question answering problem. They try various CNN architectures to learn distributed vector representations of the questions and answers and try to use the model to minimize the distance between questions and answers. The CNN model discussed in this thesis is similar in many ways to model 2 discussed in this paper, the main difference being that the same model is used in this thesis to learn embeddings for questions and for the answers whereas there are some parameters that are unshared in the method discussed by the authors. The authors report a precision@1 score of 65.3 for the InsuranceQA dataset [6].

Cohen et al. use a bidirectional LSTM with a rank sensitive loss function rather than treating the learning stage as a classification task. While previous work either used pretrained word embeddings or trained word embeddings independent from the network, Cohen's method treats them as a layer below the BLSTM network such that the performance during training would backpropagate to the word embedding layer. The authors report a precision@1 score of 0.2375. This score appears lower than many other reported scores in spite of being the superior model since the test set was built by choosing the top 10 answers retrieved using the BM25 [18] method. Thus, due to the effect of high-quality distractors, the raw numbers are lower [19].

In their paper, Tan et al. use a bidirectional Long Short-Term Memory network to learn distributed vector representations of the questions and answers and measure their similarity by using a cosine similarity metric. They further extend the model by using convolutional neural networks to learn a more composite representation of the questions and answers. They also use an attention mechanism to learn answer representations based on the question context. They also combine these two extensions and perform experiments on the TREC-QA and Insurance-QA datasets. The authors find that the bidirectional LSTM combined with attention and average pooling is the best model for the Insurance-QA dataset and report a precision@1 score of 68 [20]. As with Feng et al. the authors construct the test set with 500 random distractors. Table 2.1 provides a snapshot of the non-factoid literature discussed in this thesis.

Implementation	Corpus	Comments
Surdeanu et. al. 2011	Yahoo Webscope L6	Explore coarse word sense disambiguation, named-entity identification, syntactic parsing and semantic role labeling
Yu et. al. 2014	TREC QA	Introduce the novel idea of solving the task by using distributed vector representations and matching questions and answers based on semantic encoding.
Feng et. al. 2015	Insurance QA	Explore multiple Convolutional Neural Network Architectures
Tan et. al. 2015	Insurance QA	Explore bidirectional LSTM with attention mechanism and average pooling.
Cohen et. al. 2016	Yahoo Webscope L6	Explore Long Short Term Memory networks. Use a tougher set of distractors than related literature during testing.
Zhou et. al. 2017	SQuAD QA	Explore a question generation approach to question answering in a machine comprehension task.
Yang et. al. 2018	SemEval 2016, 2017	Explore a Generative Adversarial Network based approach to answer selection.

Table 2.1. Snapshot of Literature Survey

2.4.2 Question Generation

Question generation has been very effectively used in helping improve question answering systems, but almost all of the improvement comes from the machine comprehension task of question answering. Zhou et al. attempt to generate questions from a passage where the generated questions can be answered by sections in the text passage. The authors use a Seq2Seq model to generate natural language questions by reading the input text and answer position. The encoder thus uses this information to produce a representation of the answer which is then used by the decoder to produce a question with respect to the answer [21]. This technique is similar to the one used in this thesis, where a Seq2Seq network is used to generate the answer from the question. The main difference being that instead of using the entire paragraph, the question is generated strictly from the perspective of the answer. The authors report a BLEU score of 13.2 and evaluate on SQuAD QA dataset.

In their paper yang et al. use a Generative Adversarial Network based approach to answer selection. The generator aims to generate answers based on the question by capturing the distribution of the data. The discriminator, on the other hand, aims to differentiate between relevant and irrelevant answers based on the question. The authors propose a multi-scale matching model to generate challenging negative samples and use the samples to help train another model to help differentiate relevant answers from irrelevant ones. The model is tested on a SemEval 2016 and 2017 Task 3 and achieved state of the art results [22]. One of the main drawbacks with this method is that the stability of the GAN can often be an issue. Thus the datasets on which the method can be applied is limited to high-quality ones.

Chapter 3
Methodology

This chapter describes the methods used to solve the answer selection problem. First, a baseline approach is discussed followed by a Convolutional Neural Network approach and finally followed by a question generation based approach. The results of the baseline approach influence the methodology in the convolutional neural network approach and the CNN approach is also used in the second half of the question generation model. All the approaches are applied on both the Yahoo Answers corpus and the Library corpus and, the results are discussed in the next chapter.

3.1 Baseline Approach

In the baseline model, the best answer was selected by finding the similarity between a distributed representation of the question and the answers. The answer that was most similar to the question in terms of a cosine similarity measure was selected as the best answer. To create the distributed vector representation of the question and the answers, tf-idf (term frequency inverse document frequency) of the words in the corpus, as well as Word2Vec based word embeddings trained on the given corpus, were used [23]. To create the distributed representation of either a question or an answer, the Word2Vec representation of the word was weighed by its tf-idf score and summed up. The final vector for any question or answer was thus of the same size, i.e., the size of a single word embedding as obtained from the Word2Vec word embeddings.

Different methods were considered to construct the sentence level embeddings. Experiments were conducted where the average of the word embeddings of the question or answer were taken; the word vectors were multiplied together such that each dimension of each word vector was separately multiplied together to get the final sentence embeddings. Pretrained GloVe vector embeddings [24] were tried instead of Word2Vec embeddings and embeddings of different sizes were separately created and experiment with for each dataset. The results of many of these experiments are reported in the following chapter.

The cosine similarity between the question and each of the candidate answer was then taken and the answer with the highest cosine similarity with the question was regarded as the most appropriate answer. Although this method has its obvious flaws and fails to capture the real essence of the semantics of the represented sentence, it provides a good baseline against which results can be compared.

Cosine similarity was chosen as the metric with which to measure the similarity between the two sentence representations, as it measures the angle between the two vectors without paying attention to the magnitude. In this problem, the vector space has as many dimensions as the size of the vector. The magnitude of the vector in any particular dimension does not matter as it would simply measure the extent to which a word or concept was reinforced in the sentence. But as in this problem, the comparison is between the vector representation of a question and an answer it is very likely that some concepts that are reinforced in the answer might not be effectively reinforced in a short question. This is usually the case with uneven documents and cosine similarity works best in these cases. Below is the formula to calculate the cosine similarity between to vectors u and v.

$$cosine(\theta) = \frac{u \cdot v}{\mid u \mid\mid v \mid} \tag{3.1}$$

Thus, smaller the angle between two vectors i.e. larger the similarity between them, greater is the value of cosine similarity between them.

3.2 Convolutional Neural Network Approach

The deep learning based approach to solving this problem has a similar intuition compared to the baseline model. As in the baseline model, in the Convolutional Neural Network (CNN) based approach, we try to learn distributed vectors representations of the question and answers separately.

In the CNN approach, true representations of the questions and answers are learned by the CNN and compared [25]. While training the model, if we assume that each question Q has a correct answer $A+$, then another answer from the whole dataset is taken at random ($A-$) which does not appropriately answer the considered question. The model creates vectors for each of the items involved - the question vector, Q, the positive answer vector $A+$ and the negative answer vector $A-$, and the cosine similarity between the question and each pair of answers is measured, i.e. $cosine(Q, A+)$ and $cosine(Q, A-)$. The loss function is written such that when cosine$(Q, A+)$ is greater than $cosine(Q, A-)$ by a tuned margin m, then the model learns nothing new as it already has correctly learned representations where it appropriately predicts that the ground-truth answer as more similar to the question than an answer which has been selected from the corpus at random. But when the $cosine(Q, A+)$ is greater than $cosine(Q, A-)$ by a margin less than 'm' or it is smaller, then the weights of the model are updated in a direction so as to rectify this issue. The loss function, in this case, is based on hinge loss and represented below [6].

$$L = max\{0, m - cos(Q, A+) + cos(Q, A-)\} \tag{3.2}$$

Dropout was used to prevent the model from overfitting. Dropout is a regularization method which refers to randomly ignoring some units or neurons while training. Dropout has the effect of making a layer seem like it has a different number of nodes than it actually does. All incoming and outgoing connections to the unit which has been dropped out are temporarily removed. This has the effect of making the training process noisy forcing nodes to take on more responsibility for the outputs, meaning neurons in a layer learn to co-adapt to correct mistakes from a previous layer. This has the effect of making the model more robust and insensitive to overfitting [26].

3.2.1 Architecture

CNNs have historically been used for image classification. But in recent times, they have shown application in Natural Language Processing and text classification. The 2D CNN which was used here is made up of a convolutional layer, followed by a max pooling layer and a fully connected layer. The CNN makes use of a number of filters of various sizes in order to learn the sentence level embeddings of the word vector representations of the questions and the answers fed to it. The size of the breadth dimension of the filters is equal to the size of the word vector representations. Figure 3.1 shows the structure of the CNN model which has been used.

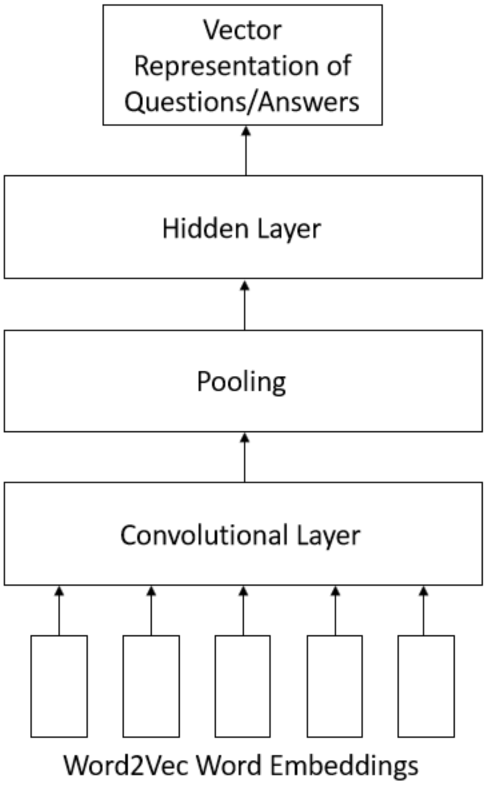

Figure 3.1. CNN Architecture to Create Question/Answer Embeddings

The Convolutional Neural Networks leverage two important concepts to learn the distributed vector representations of the sentences - sparse interactions and parameter sharing. Traditional Neural Networks have fully connected layers where each output is a result of each input. But in a CNN, due to the convolutional layer, the output only interacts with a narrow window which is equal to the filter size. This sparse interaction is a very important part of a CNN and greatly improves the performance of the CNN. Parameter sharing also helps reduce the number of parameters that the CNN has to learn. As the same filter is used as a sliding window across the entire output, it learns the various features of the input using the same parameters. This not only helps reduce the number of parameters but also helps learn the same kind of features uniformly across the input. After the filter has extracted the features from the input, the pooling layer is used to reduce the dimensionality of the input [27]. Here 1-max pooling is used where the maximum value of the vector from each feature vector is extracted and concatenated to form the final vector.

3.2.2 Setup

Questions or answers are fed in the form of their word embeddings to the CNN. The length of the questions and answers are fixed, and a <PAD> token is used to offset any discrepancies in length. With a dimension of 'd' for word vectors and a length, 'l' for each sentence, the sentence matrix has the input shape of d * l.

As discussed previously, CNNs preserve 2D spatial orientation in images. Text, like images, have a sequence and a structure which is 1 dimensional. But as the input is fed in terms of a 2D structure due to the dimension of the word vectors, the size of the filters can be fixed in one dimension and varied in the other. This means that one of the dimensions of the filter is fixed to the size of the word vector and the size is varied in the other dimension. Four different sizes of filters were used in this thesis.

In order to create the feature map, each filter performs an element-wise product across the window that it covers and then sums them up to obtain one number. The filter then moves down one word and overlays across the next set of words or word vectors. This is continued across the entire vector to obtain a vector. A non-linear activation function ReLU is applied on the constructed vector to obtain the feature map. 1-max pooling is applied across each of the feature maps and concatenated. Thus, after max pooling, a vector of size equal to the number of filters is created. This is passed through a fully connected layer to finally obtain the question/answer vector.

3.3 Question Generation Approach

In the CNN based approach, we created distributed vector representations of the questions and answers from the Word2Vec embeddings of the words in the questions and answers. The similarity of the embeddings thus created were compared by a cosine similarity metric to find the ones that are closest to each other. The disadvantage of the method was that we were comparing questions with answers. This is not ideal because, in non-factoid question answering, there is little overlap between the questions and the answers. The answers are also a lot longer than the questions, and there might even be significant variation in the concepts discussed in the question and the answer. In the question generation based approach, an encoder-decoder model is used to learn the questions from the answers as usually seen in the case of machine translation. The primary intuition behind this approach is that only a correct answer can be used to generate a question which is similar or close to the real question.

3.3.1 Encoder Decoder Model

The encoder-decoder model used in this thesis is based on the machine translation technique used by Cho et al. [28]. The difference being that instead of translation, both the encoder and the decoder have access to the same vocabulary. The encoder has access to the answer and the decoder to the question. Figure 3.2 shows the structure of the model used in the question generation approach.

3.3.2 Encoder

The encoder is a neural model acted upon by the answer sequence. The distributed vector embeddings of the answer are fed to the encoder. The encoder is bidirectional LSTM [29] with 2 hidden layers. The BLSTM maintains two hidden states, one for the forward pass of the input and the other for the backward pass of the input in order to incorporate information from both the past and the future of the input sentence.

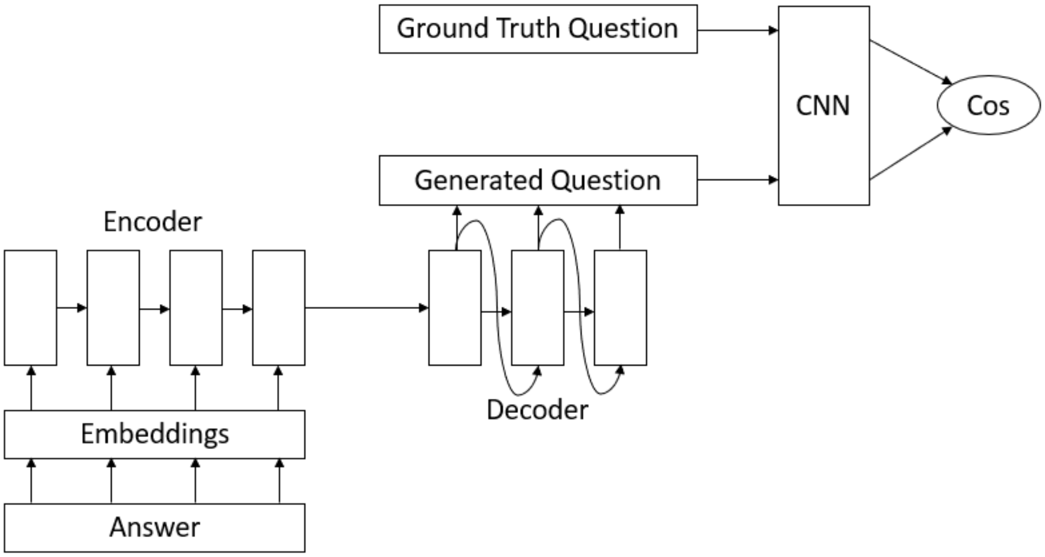

Figure 3.2. Architecture of Question Generation Approach

3.3.3 Decoder

The decoder is a neural model that generates outputs sequentially thereby generating the question sequence. The decoder models a conditional distribution at each time step which is parameterized by θ,

$$p_\theta(y_t \mid y_{<t}, A) \tag{3.3}$$

where y_t represents the output at the current timestep and $y_{<t}$ represents the output at previous timesteps and A represents the distribution of the answers. Thus at each timestep, a word is sampled according to the above equation [30]. A softmax function is used which outputs the probability of each word in the vocabulary and the one with the highest probability is chosen to feed into and fed as input for the next timestep.

3.3.4 Training

For training this model, two different approaches were used. First, the Seq2Seq model was trained to generate questions using maximum likelihood estimation. Dropout was used to make the model more insensitive to overfitting and gradient clipping was used to mitigate the exploding gradient problem as the length of the input sequences in the encoder are long due to long answers [31]. Thus whenever the norm exceeded the set threshold, the gradients were clipped to the threshold value.

In the case of natural language generation, BLEU score is usually reported. BLEU or the bilingual evaluation understudy is a metric for comparing a translated text to the original translation reference in the case of machine translation. It can also be used in the case of any text generation task and can be compared with the original. It usually measures the overlap of the n-grams between the generated text and the original text. In the case of translation and even factoid question generation, we can see that the questions are fairly straightforward and need to have a strict sequence of words for them to make any real sense. But here we do not care about the sequence of words. We only care if the concepts from the answer were translated to the question. Thus we simply measure the average number of words which are shared between the generated question and the reference question. The metric chosen is loose due to the fact that the weights are to be further tuned as a part of the full model. As the questions generated as a part of the process do not matter and are not going to used separate from the full model, a casual approach is taken to tune the parameters of the encoder-decoder model.

The weights of the trained model are used to initialize the weights of the Seq2Seq section of the full model as seen in figure 3.2. The weights of the Seq2Seq model are further tuned along with the weights of the Convolutional Neural Network. The loss function used to train the CNN and further tune the Seq2Seq model has already been discussed and can be seen in equation 3.2. The loss function used to separately train the Seq2Seq section of the model can be defined as the weighted softmax cross entropy distribution between the predicted distribution p_t and the ground truth distribution of the questions in the training set o_t.

$$L = -\sum_{t=1}^{Y} o_t log(p_t) \tag{3.4}$$

Thus when training the Seq2Seq network separately, the encoder was fed with the answers in the dataset and the decoder was fed with the corresponding questions. Teacher forcing was used during training of the encoder-decoder model separately to avoid accumulation of error along the series. While testing, the output of the previous timestep was fed as the input to the next time step. To start decoding, a start-of-sentence token was used. While training the full model, the armed Seq2Seq model was used to predict the question of various answers (both positive and negative) and the Convolutional Neural Network was used as seen in section 3.2 to reduce the distance between the true question of the true answer and the question generated from the true answer, while maximizing the distance between the true question of the true answer and the questions generated from all of the negative answers.

Chapter 4
Evaluation and Results

4.1 Datasets

Two datasets were used to evaluate the models discussed. The Yahoo L6 webscope and the Library dataset.

4.1.1 Yahoo Webscope L6

The Yahoo Webscope L6 dataset has been collected from the Yahoo Answers website. Yahoo Answers is a community question answering website where users can ask questions and other users can answer them. The asker also gets to select the best answer out of all the possible answers given. A voting mechanism also exists where users can vote on a particular answer. If the asker does not select the best answer, then the one with the maximum votes gets selected as the best answer.

The Webscope L6 follows a similar trend. It contains a total of 4,483,032 question answers. Out of these, there are 87362 unique questions. Each question has 'n best answers' and a top answer selected based on the voting mechanism described above. The average number of words in an answer rounded to a whole number is 46, while that in a question is 10. Table 4.1 offers a few example question and answers.

4.1.2 Library Dataset

A custom dataset was created in order to evaluate the answer selection models. This dataset was created from answers of students to information literacy questions on an online portal. Penn State libraries has an information literacy branch which offers

Q	How to get rid of a beehive?
A	Call an area apiarist. They should be able to help you and would most likely remove them at no charge in exchange for the hive. The bees have value and they now belong to you.
Q	how do Elephants communicate?
A	There make very low frequency sounds that we cannot hear. They also make sounds that we do hear and it is because they what other animals to hear it.
Q	Why is it considered unlucky to open an umbrella indoors?
A	Possibly because African royalty used umbrellas as protection from the sun, and opening one in the shade was considered an insult to the sun god. Opening it indoors was also taboo.

Table 4.1. Sample Questions and Answers from Yahoo Webscope L6

online micro-credentials on information literacy [32]. Students are introduced to the badges and recommended to complete them as a part of the course curriculum. The students can access the portal, read relevant material and can finally answer questions. The librarians then read the answers that students provided and annotate them 1 for accepted and 0 for rejected. These annotated answers were used to create the dataset.

This dataset contains 3300 questions and answers with 15 unique questions. The average length of an answer is 88 while that of a question is 19. Table 4.2 offers a few example questions and answers from the dataset.

4.2 Evaluation

The evaluation metrics used are Precision@1 and Mean Reciprocal Rank. Both these metrics are commonly used in information retrieval and question answering frameworks. For the precision@1 metric, the rank threshold is set at first place, meaning, if the correct answer is ranked highest by the model, then a point 1 is awarded, 0 otherwise. Precision@1 is a very aggressive evaluation metric and the mean is taken to evaluate performance over a collection of questions. This metric is offset by the mean reciprocal rank metric. Mean reciprocal rank is the inverse

Q	Create a summary on the the main features of both popularly and scholarly journal articles.
A	The popular article is directed at a broad, general audience. There are ads, pictures, usually magazines, newspapers, found at bookstore, book stands, public places. The scholarly article is directed other scholars. Written by experts with advanced degrees. Graphs, data, statistics, references, peer reviewed with few images.
Q	Provide your citation generator and your citation manager and the reasons for picking each of them.
A	The citation manager that I have chosen is the Zotero. The citation builder that I will be using will be the NCSU Library citation builder. The reason I will be using the NCSU citation generator is because it is simple tool that is easy to use and makes citing a lot easier for me. I have started using it and it has made finding information on the author a lot easier.

Table 4.2. Sample Questions and Answers from Library Dataset

of the highest ranked answer which is correct for a given question. The formula to calculate the mean reciprocal rank can be seen in equation 4.1. Here Q depicts the set of questions and *rank* depicts the position at which the model ranks the correct answer for the question given multiple distractors [19].

$$MRR = \frac{1}{\mid Q \mid} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{4.1}$$

The datasets were first randomly divided into train and test sets in the ratio 90:10. The test set was further processed where for each correct question answer pair, 499 other random incorrect answers were sampled from the test set. Thus each question has 499 random distractors collected from the test set [20] [6].

4.3 Results

4.3.1 Yahoo Answers

The results of the models on the Yahoo dataset is shown in Table 4.3. The baseline of the model was created by taking tf-idf weighted sum of the Word2Vec word embeddings of the words in the question and the answer and taking the cosine similarity between them. The results of all the proposed methods have a lot of improvement over the baseline which is a simple naive implementation. The performance of the model developed by Feng et al. [6] is very similar to the CNN approach discussed in this thesis as the architectures are similar. Both of them follow the same structure, but the CNN approach discussed in this thesis uses the same parameters throughout the model to learn the vector distributions of both the questions and the answers.

Implementation	**P@1**	**MRR**
Baseline Approach	0.323	0.477
Feng et. al.	0.582	0.723
Cohen et. al.	**0.604**	**0.749**
CNN Approach	0.585	0.730
Question Generation Approach	0.459	0.602

Table 4.3. Results for Yahoo Answers

As is evident from the table, the LSTM approach followed by Cohen et al. [19] gives the best results. They use the attention mechanism on a bidirectional LSTM model to learn long term dependencies. As seen from Severyn et al. on the TREC-QA task, CNNs perform very well on the factoid question-answering task [33] and thus the improved performance of the LSTM approach could be explained by the fact that it is able to capture more of the non-factoid nature of the overlap between the questions and the answers while the CNN captures more of the factoid nature. But the CNN approach has the upper hand due to the fact that it takes a lot less time to train.

Finally, the question generation approach shows promise, and provides significant improvement over the baseline results, but falls short of the state of the art results for this dataset. This can be explained by the fact that the dataset is rather small to train the Seq2Seq network. The questions generated by the model on many occasions seem to fall short from ideal responses. Table 4.4 shows a few sample questions generated by the sequence to sequence model compared to the original questions. The first three rows of the table list some of the best responses learned by the seq2seq section. Higher quality questions are generated when the answers are short. For longer answers, the model does not output good quality questions. It was also noticed that shorter questions were favoured by the model during generation. This can be explained by the fact that the dataset contained more instances of short questions than long ones.

True	how do i remove yahoo messenger ?
Generated	how to delete yahoo messenger ?
True	what is that distinctive smell just before it rains ?
Generated	what causes it to smell like rain water ?
True	what are some ways to get rid of the hiccups ?
Generated	how do i stop hiccups ?
True	how would you describe the differences between organizational behavior and organizational theory ?
Generated	what are the difference between ... ... ?
True	how does the food processing factory remove corn from maize ?
Generated	how do you remove anything ?

Table 4.4. Sample Questions Generated by the Seq2Seq Model

4.3.2 Library Corpus

The results of the models on the Library corpus are shown in Table 4.5. As can be seen, the results on this dataset are not as impressive as that on the Yahoo Answers dataset. Even though the CNN approach gives the best precision@1 scores, the MRR of the Feng et al. model is higher. It can also be seen that the results are very closely matched with the difference only being in the third decimal place. This poor performance can be explained by the fact that the dataset is very small. The word embeddings used were those created from the Yahoo Answers dataset and the Library corpus put together as the performance dipped while using pretrained Google Word2Vec embeddings. To generate the word embeddings for this dataset, gensim framework was used with a skip-gram technique [34], with a window size of 5 while skipping words that did not occur at least twice in the corpus.

The performance of the convolutional neural network based architectures was better than that of the LSTM networks because there is usually significant word overlap between the questions and the answers. We saw previously that CNN architectures were good at capturing factoid information and as factoid information usually have high word overlap, it is not a giant leap to suggest that CNNs work well when there is significant word overlap. As the library corpus dataset was built of students' answers to questions, it is reasonable that word overlap between the questions and the answers in the library corpus is much more significant than in the Yahoo Answers corpus. This is due to the fact that students tend to repeat the question in the form of an answer in their answers. For example, when asked the question, 'Are you surprised by the level of organization of cheating and plagiarism? Explain', students tend to answer with 'I am surprised by the level of organization of cheating and plagiarism.'

Implementation	**P@1**	**MRR**
Baseline Approach	0.111	0.249
Feng et. al.	0.309	**0.441**
Cohen et. al.	0.305	0.430
CNN Approach	**0.311**	0.438
Question Generation Approach	0.158	0.281

Table 4.5. Results for Library Corpus

It can also be seen that the performance on the question generation approach is only a small improvement on the baseline results. This can also be explained by the fact that the Seq2Seq model needs a lot more data to train. When it was simply trained on the Library corpus dataset, the performance was much worse than when trained on a combination of the Yahoo answers and the library corpus.

Chapter 5
Discussion and Future Work

5.1 Conclusion

In this thesis, the answer selection task was examined for non-factoid question answers. A convolution neural network based model and a question generation based model built on an encoder-decoder framework were examined. From the results, it's evident that both LSTM models and CNN models work well in the case of non-factoid answer selection. Although the convolutional neural network based methods seem to work better when there is an overlap between the words in the question and answer, it can be readily used for non-factoid question answering as the results are comparable to that of state of the art. With no significant difference between the results from the different models, the convolutional neural network based approaches offer a better alternative as they can be trained in a much shorter time.

The question generation based approach was built using a Seq2Seq model to generate the questions and the CNN model was used to compare the generated questions with the real questions. This method is a novel one as it has not been undertaken before for the answer selection task. From the results it is evident that it does not perform as well as the other approach taken. The main pitfall of this approach is that the encoder-decoder model does not always generate high-quality questions which are comparable to the real one. This generates unnecessary noise in the dataset and thus the performance suffers. The reason the encoder-decoder model does not generate relevant questions consistently is because in long answers it is a difficult task to pick concepts or words the model has to pay attention to. Question generation approaches which are undertaken in machine comprehension tasks usually have a paragraph or more of text and a separate answer. The confluence of the two gives the model a concrete concept to pay attention to. But in cases where there is only an answer, the task of generating relevant questions is a harder one.

5.2 Future Work

For future work, it would be interesting to further develop and apply the models put forward in this thesis to varied datasets to investigate the hypothesis that convolutional neural network based approaches work better for questions answers that have a higher word overlap, while LSTM based approaches work better where there is only a conceptual overlap between questions and answers. It would also be interesting to investigate the absolute answer quality of given answers instead of evaluating them in terms of an information retrieval problem.

Various improvements can be made to the question generation approach. Beam search applied to the encoder-decoder model might improve the quality of questions generated. The loss function used was a naive one and can be improved to improve the quality of the questions generated. A strategy to improve the attention mechanism could be developed whereby the model can learn what concepts in the answers are most important. Although the question generation based approach did not perform as well as the other approach, it has a much larger room for improvement and further development.

References

[1] Bian, J., Y. Liu, E. Agichtein, and H. Zha (2008) ‘Finding the right facts in the crowd: factoid question answering over social media,’ in *Proceedings of the 17th international conference on World Wide Web*, ACM, pp. 467–476.

[2] Surdeanu, M., M. Ciaramita, and H. Zaragoza (2008) ‘Learning to rank answers on large online QA collections,’ *Proceedings of ACL-08: HLT*, pp. 719–727.

[3] Wang, B., B. Liu, C. Sun, X. Wang, and L. Sun (2009) ‘Extracting Chinese question-answer pairs from online forums,’ in *2009 IEEE International Conference on Systems, Man and Cybernetics*, IEEE, pp. 1159–1164.

[4] Fichman, P. (2011) ‘A comparative assessment of answer quality on four question answering sites,’ *Journal of Information Science*, **37**(5), pp. 476–486.

[5] Voorhees, E. M. and D. M. Tice (2000) ‘Building a question answering test collection,’ in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, pp. 200–207.

[6] Feng, M., B. Xiang, M. R. Glass, L. Wang, and B. Zhou (2015) ‘Applying deep learning to answer selection: A study and an open task,’ in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, IEEE, pp. 813–820.

[7] Yang, Y., W.-t. Yih, and C. Meek (2015) ‘Wikiqa: A challenge dataset for open-domain question answering,’ in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2013–2018.

[8] Hermann, K. M., T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom (2015) ‘Teaching machines to read and comprehend,’ in *Advances in Neural Information Processing Systems*, pp. 1693–1701.

[9] Rajpurkar, P., J. Zhang, K. Lopyrev, and P. Liang (2016) ‘Squad: 100,000+ questions for machine comprehension of text,’ *arXiv preprint arXiv:1606.05250.*

[10] Trischler, A., T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, and K. Suleman (2016) 'Newsqa: A machine comprehension dataset,' *arXiv preprint arXiv:1611.09830*.

[11] Kwiatkowski, T., J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, M. Kelcey, J. Devlin, K. Lee, K. N. Toutanova, L. Jones, M.-W. Chang, A. Dai, J. Uszkoreit, Q. Le, and S. Petrov (2019) 'Natural Questions: a Benchmark for Question Answering Research,' *Transactions of the Association of Computational Linguistics*.

[12] Punyakanok, V., D. Roth, and W.-t. Yih (2004) *Natural language inference via dependency tree mapping: An application to question answering*, Tech. rep.

[13] Shen, D., G.-J. M. Kruijff, and D. Klakow (2005) 'Exploring syntactic relation patterns for question answering,' in *International Conference on Natural Language Processing*, Springer, pp. 507–518.

[14] Iyyer, M., J. Boyd-Graber, L. Claudino, R. Socher, and H. Daumé III (2014) 'A neural network for factoid question answering over paragraphs,' in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 633–644.

[15] Girju, R. (2003) 'Automatic detection of causal relations for question answering,' in *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering-Volume 12*, Association for Computational Linguistics, pp. 76–83.

[16] Surdeanu, M., M. Ciaramita, and H. Zaragoza (2011) 'Learning to rank answers to non-factoid questions from web collections,' *Computational linguistics*, **37**(2), pp. 351–383.

[17] Yu, L., K. M. Hermann, P. Blunsom, and S. Pulman (2014) 'Deep learning for answer sentence selection,' *arXiv preprint arXiv:1412.1632*.

[18] Robertson, S. E. and S. Walker (1994) 'Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval,' in *SIGIR'94*, Springer, pp. 232–241.

[19] Cohen, D. and W. B. Croft (2016) 'End to end long short term memory networks for non-factoid question answering,' in *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, ACM, pp. 143–146.

[20] Tan, M., C. d. Santos, B. Xiang, and B. Zhou (2015) 'LSTM-based deep learning models for non-factoid answer selection,' *arXiv preprint arXiv:1511.04108.*

[21] Zhou, Q., N. Yang, F. Wei, C. Tan, H. Bao, and M. Zhou (2017) 'Neural question generation from text: A preliminary study,' in *National CCF Conference on Natural Language Processing and Chinese Computing*, Springer, pp. 662–671.

[22] Yang, X., M. Wang, W. Wang, M. Khabsa, and A. Awadallah (2018) 'Adversarial Training for Community Question Answer Selection Based on Multi-scale Matching,' *arXiv preprint arXiv:1804.08058.*

[23] Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013) 'Distributed representations of words and phrases and their compositionality,' in *Advances in neural information processing systems*, pp. 3111–3119.

[24] Pennington, J., R. Socher, and C. Manning (2014) 'Glove: Global vectors for word representation,' in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

[25] Hu, B., Z. Lu, H. Li, and Q. Chen (2014) 'Convolutional neural network architectures for matching natural language sentences,' in *Advances in neural information processing systems*, pp. 2042–2050.

[26] Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014) 'Dropout: a simple way to prevent neural networks from overfitting,' *The Journal of Machine Learning Research*, **15**(1), pp. 1929–1958.

[27] Kim, Y. (2014) 'Convolutional neural networks for sentence classification,' *arXiv preprint arXiv:1408.5882.*

[28] Cho, K., B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio (2014) 'Learning phrase representations using RNN encoder-decoder for statistical machine translation,' *arXiv preprint arXiv:1406.1078.*

[29] Schuster, M. and K. K. Paliwal (1997) 'Bidirectional recurrent neural networks,' *IEEE Transactions on Signal Processing*, **45**(11), pp. 2673–2681.

[30] Yuan, X., T. Wang, C. Gulcehre, A. Sordoni, P. Bachman, S. Subramanian, S. Zhang, and A. Trischler (2017) 'Machine comprehension by text-to-text neural question generation,' *arXiv preprint arXiv:1705.02012.*

[31] Pascanu, R., T. Mikolov, and Y. Bengio (2013) 'On the difficulty of training recurrent neural networks,' in *International conference on machine learning*, pp. 1310–1318.

[32] Rimland, E. and V. Raish (2017) 'Design principles for digital badges used in libraries,' *Journal of Electronic Resources Librarianship*, **29**(4), pp. 211–220.

[33] Severyn, A. and A. Moschitti (2015) 'Learning to rank short text pairs with convolutional deep neural networks,' in *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, ACM, pp. 373–382.

[34] Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013) 'Efficient estimation of word representations in vector space,' *arXiv preprint arXiv:1301.3781*.