Factoid Question Generation from Paragraph
(Group-11)

Akshat Anand(14055)	Akshay Paswan(14061)	Ankur Kumar(14109)
akanand@iitk.ac.in	akshaygr@iitk.ac.in	ankurk@iitk.ac.in
Meena Vikas(14384)	Shahzeb Haidar(14624)	Sudhir Kumar(14716)
vikasm@iitk.ac.in	shaidar@iitk.ac.in	sudhirk@iitk.ac.in

Introduction

Question generation in Natural Language Processing (NLP) is the task of designing models and algorithm to generate question(s) from given text. Further, the question generated should be close to those generated by humans in terms of syntax and semantics. This task is important because it has a lot of applications, both in academia and industry. For example, it can then be used to develop dataset for other important learning tasks like Question Answering. Or it can be used as important component of chatbots deployed for evaluating mental health. Despite its importance, there has not been much work in this field. Moreover, most of the works in past have tried to solve the problem using rule-based approach, which is not able to generate *natural* questions and capture semantics properly. Recent advancements in deep neural networks help us to tackle these problems efficiently.
In our work, we first study automatic question generation for sentences. We extend this idea of question generation to bigger texts like paragraph or whole document by introducing a novel approach. In the process we face challenges like question generation from long sentences which we solve using *attention mechanism* introduced by *Luong et al.* Finally, we discuss the shortcomings of our approach and future work to solve them as well.

Problem Statement

Our problem statement is to generate factoid (fact based) question(s) from given text where a text can be a sentence or a paragraph or a document. Most of the works have focused on generating question(s) from a sentence. Notably, the recent work by *Du et al.*[1] addresses the problem for a sentence quite well which they simply extend to paragraphs by considering a paragraph as long sentence. In their followup paper, *Du et al.*[2] introduce a novel of approach of focusing on parts of paragraph to generate questions from. Our approach is similar in spirit but differs slightly in how we choose to focus on parts of paragraph. We divide the entire problem of question generation into two sub-problems:

- **Senetence Selection**: Identifying candidate sentences from text
- **Question Generation**: Generate questions from candidate sentences

When we consider these sub-problems individually, we find that earlier one is a problem concerning semantics of sentences in a paragraph, while the later one is problem of machine translation. Our idea is based on finding sentences from which we can generate *meaningful* questions. We call such sentences as *candidate sentences*. Once we have candidate sentences, we try to generate questions for them. The problem can be formulated as given a paragraph P, containing n sentences, we have to find k most efficient sentences which can be converted to question. For this purpose, we will find maximum likelihood of sentences being candidate sentences given paragraph P.

$$y = \arg\max_{s} \ log\ p(s|P) = \arg\max_{s} \sum_{i=0}^{i=n} log\ p(s_i|P)$$

where $s_i = 1$ if that sentence is question worthy else 0.

For question generation we are trying to find a sequence of arbitrary length q from a sentence s such that it maximizes log likelihood of conditional probability of q given s. We will use attention mechanism for finding various words in q.

$$z = \arg\max_{q} \ log\ p(q|s) = \arg\max_{q} \sum_{i=0}^{i=|q|} log\ p(q_i|s, q_{<i})$$

Related Works

As discussed above, our approach consists of two part. The first one being a classification task which we have introduced and the second part is question generation task given an input sentence. Therefore, most of the existing works are related to question generation task.

We discuss some of the traditional approaches and recent approaches to question generation which use deep neural networks. So we also look at the recent works in Neural Machine Translation task which is also closely related to the problem of question generation.

Question Generation

We have an input sentence x (sequence of tokens) and we want to generate a question y (again a sequence) related to input. So our task is to optimize the following:

$$\overline{y} = \arg\max_y P(y|x) = \arg\max_y \log P(y|x)$$
$$= \arg\max_y \sum_{t=1}^{|y|} \log p(y_t|x, y_{<t})$$

These types of probability distribution is modeled by RNN based encoder-decoder model.

In the paper **Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus [3]**, authors plan to transduce facts into questions, thus producing 30M Factoid Question-Answer Corpus. For this purpose they use Freebase knowledge base which contains several facts. Triplet of (subject, relationship, object) is a fact. The authors chose to generate question which consists of subject and relationship from a fact. Answer to this question is simply corresponding object from the fact. They employ *SimpleQuestions* dataset for training purpose. The authors propose two models: *Single-Placeholder model* and *Multi-Placeholder model*. In *Single-Placeholder* model, the subject of the fact in the question is replaced by token $< placeholder >$ at training time. At test time, generated $< placeholder >$ tokens are replaced by the subject. In other model, there are different types of $< placeholder >$ token based on the *relationship* between *subject* and *object*. The models were evaluated on *BLEU* and *METEOR* metric against template-based baseline (proposed by the authors). The best performing model scored about two percentage points higher on both the metrics as compared to baseline.

In **Learning to Ask: Neural Question Generation for Reading Comprehension [1]**, the authors introduce *global attention-based mechanism* into the encoder-decoder model for the task. As there are two tasks, one related to sentence and the other related to paragraph, the input to decoder changes accordingly. For sentences, input to decoder is simply the encoding of input sentence whereas for paragraphs, input to decoder is concatenation of encoding of sentences and paragraph. For evaluation, the authors used the package by Chen et al.[4] which contains different metrics like BLEU n, METEOR etc. BLEU 3, BLEU 4 and METEOR scores for the models were two to four percentage points higher than H&S baseline which is a rule based overgenerate and rank system.

In **Automatic Chinese Factual Question Generation [5]**, the authors have come up with a novel approach to generate questions based on Chinese texts. The paper have a good architecture which can be applied to any question generation problem with different algorithms. The architecture consists for 4 stages: *Pre-processing, Sentence identification and simplification, Translation to questions and ranking of questions to generate top best questions.* Pre-processing deals with tokenizations, etc. For sentence identification and simplification, various algorithms can be applied. Translation to question requires identification of words or clauses that can be replaced by question words and then generation of sequence of words that make up the question. Finally, various question generated can be ranked using learning to rank.

In the paper, **Identifying Where to Focus in Reading Comprehension for Neural Question Generation [2]**, authors have explained a new novel idea of finding the question worthy parts or sentences in a paragraph. The authors used a bi-LSTM network to train the model to find question worthy portions of a paragraph. The model was trained using SQuAD dataset and bench marked against human and various text summarizers. The author has tried to maximize the log likelihood of conditional probability of any portion being a question worthy given the whole paragraph.

Neural Machine Translation

The basic form of Neural Machine Translation (NMT) method is encoder-decoder based model. Encoder part of the model encodes input sentence from source language. This encoding s is fed to the decoder which generates a sequence of tokens terminated by $< eos >$, one token at a time. Generally, we have to minimize the following objective

$$J = \sum_{(x,y)\in D} -\log p(y|x)$$
$$= \sum_{(x,y)\in D} \sum_j -\log p(y_j|y_{<j}, x)$$

where D is training corpus, x is input sequence and y is output sequence.

In **Effective Approaches to Attention-based Neural Machine Translation [6]**, the authors propose novel NMT method which is based on attention: global attention and local attention. Global attention based models focus on the entire source positions whereas local attention based models focus on a subset of source positions. The authors define context vector c_t which is used to find the context in the input sequence for current target token at decoder timestep t. They use c_t to update hidden state of decoder as $\tilde{h}_t = \tanh(W_c[c_t; h_t])$ where $[c_t; h_t]$ is the concatenation of context vector and original hidden state at timestep t. The context vector is the weighted average of set

of hidden states (which differ for global and local approach) on the source side. The authors use two different types of BLEU scores: tokenized BLEU, to compare their models with existing NMT work and NIST BLEU, to compare with WMT results. For tokenized BLEU metric, the best model from the authors outperformed the existing best performing model on newstest2014 by 1.4 percentage points. For NIST BLEU, the same was 1 percentage point on newstest2015.

Our Method

The problem at hand can be summarized as : *Given a paragraph, automatically generate questions based on the facts presented in the paragraph.*
We take inspiration from the mechanism adopted by humans for simple question generation. When we see a text, we focus on the facts given in the text. Then we proceed to extract important factual statements from the text. Next, we convert the sentence into one or more interrogative sentences preceded by 'What', 'Who', 'When', 'Where', 'How' and so on.

We use the same approach to generate questions from paragraph/document. We first extract factual sentence from text and then generate questions from them. This approach has several advantages. First, directly generating questions from paragraph is a much harder problem as compared to generating questions from sentences. For the later a number of ways are available. Secondly, it makes our approach modular since our question generation part is completely unaffected by the way we extract factual sentences. Hence, any method can be used for the question generation part.

- Factual sentence extraction : We plan to use bidirectional LSTM based approach.[2] First step is to create encoded vectors (s_i) from sentences in paragraph. For every encoded sentence vector (s_i), we obtain a hidden state $h_{i,t}$.

$$\overrightarrow{h_{i,t}} = \overrightarrow{LSTM}(s_i, \overrightarrow{h_{i,t-1}})$$
$$\overleftarrow{h_{i,t}} = \overleftarrow{LSTM}(s_i, \overleftarrow{h_{i,t+1}})$$
$$h_{i,t} = [\overrightarrow{h_{i,t}}, \overleftarrow{h_{i,t}}]$$

  Now this state is used to obtain the likelihood of the sentence s_i being a suitable sentence for question generation.

- Sentence to Question Generation: There are several techniques available for this task. We will be using RNN Encoder–Decoder framework as described in [1]. In this framework, an encoder encodes the input sentence s into a vector c. This vector c is created using a bidirectional LSTM network. The decoder is then trained to generate the required question for sentence s using the vector c as the initial hidden state. The decoder predicts the next word s'_t, given previous state and already predicted words $\{s'_1, s'_2, ..., s'_{t-1}\}$.

The overall pipeline is shown in Figure 1 where the sentence classifier model outputs candidate sentences from text which is then used to generate questions.
We will now describe our model for each subproblems.

Sentence Selection

In this subproblem, we are splitting any textual file into separate sentences and then classifying those sentences as being suitable candidate for question generation or not. For the model, we have first tokenize all the words of sentence and then apply punctuation and stop word removal. The tokens achieved after this pre-processing step is Embedded with word2vec vectors, each of 100 dimensionn. We now pad number of tokens/ embeddings in each sentence to 20 which we got from hyperparameter tuning. We prune from last if tokens exceed 20 counts or append zeros embedding to pad to 20 vector size. Now, we reshape the data to make it in shape of input to LSTM network. We take 20 timestamps and each timestamp have a word each of which is embedded with word vectors of 100 dimensions.
Coming to our architecture, the model consist of 2 LSTM networks stacked on top of each other followed by a dropout layer. The number of nodes in first LSTM network is 100 while in second it is 25. We have applied sigmoid based attention mechanism and then again a dropout layer. The final layer is Dense layer which uses sigmoid activation and converts into output vector.
The optimizer used is Adams and loss is binary cross entropy. The complete model can visualized in Figure 2.

Question generation

For question generation we have used *Encoder-Decoder* architecture [7]. In this architecture, an input sentence is first passed to an *Encoder* which outputs a vector which is encoding of the sentence. This encoding is used to initialize hidden state of decoder. Then decoder uses this hidden state with along with a <*SOS*> tag to predict next word and generates new hidden state as well. Using this predicted word as input word and new hidden state we iterate the whole process untill we get a *EOS* tag. This model works fine for shorter sequences but not for longer sentences. To solve this problem, we use attention mechanism specified by *Luong et al.*[6]. In each decoder step a context vector is created using all the output states of encoder. This context vector is the weighted average of output states of encoder. The weights used here are attention weights which are calculated by comparing hidden state of decoder with output state of encoder. This helps the decoder in focusing on selected part of input sentence in order to predict the next word.
We use 300 dimensional pretrained GLoVe vectors for word embeddings. The encoder consists of 2 LSTM

Figure 1: Overall model pipeline

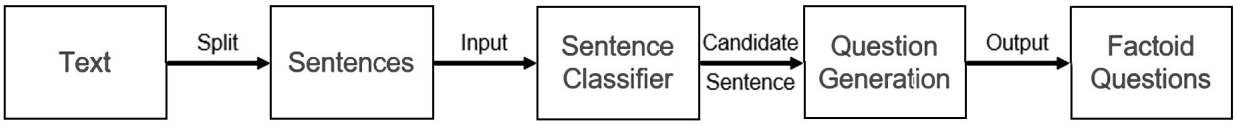


Figure 2: Sentence selection pipeline

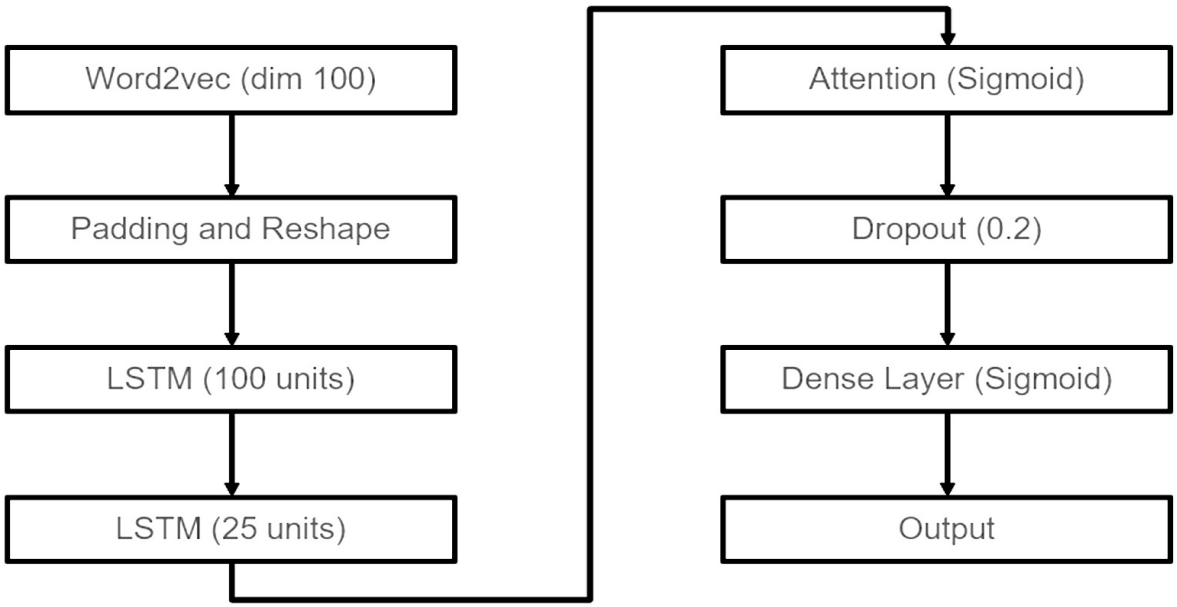
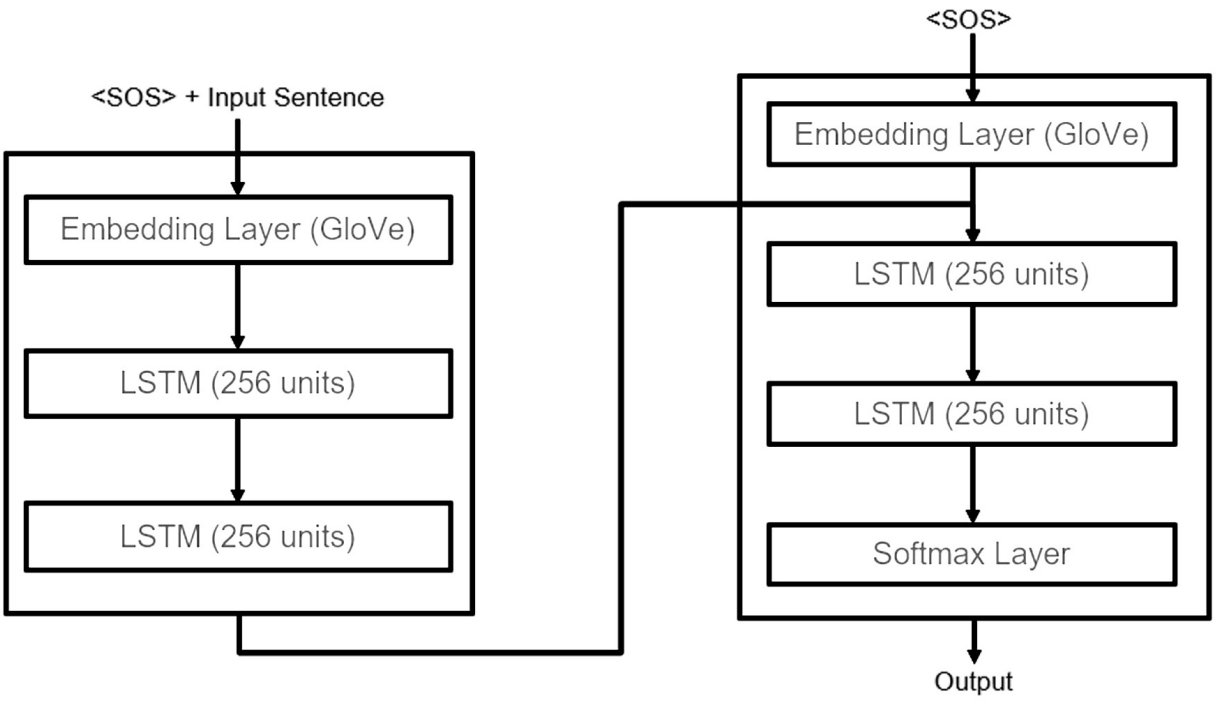


Figure 3: Question generation model pipeline



layers, both having 256 units. The decoder also consists of 2 layers of LSTM having 256 units. After these layers, there is a softmax layer to output probability distribution over whole vocabulary. The architecture for question generation is shown in Figure 3.

Experiment

Dataset

We used *Stanford Question Answering Dataset (SQuAD)* dataset [8] for our purpose. SQuAD dataset consists of 442 topics taken from Wikipedia. For each of the topics, there is a list of paragraphs (paragraphs in the Wikipedia article of that topic). Number of paragraphs for each of the topics may vary. Each paragraph is a dictionary consisting of keys: *context*, i.e. actual paragraph content, and *qas*, i.e. a list of dictionary. Each dictionary in the list *qas* contains a question id, (actual) question, answer to the question and start index of the answer in the context paragraph. The dataset contains over 100,000 pairs of question-answer in total. Amazon Mechanical Turk workers have generated questions in their own words about the Wikipedia articles. Answers to these questions are exact phrases from the paragraph.

Sentence Classifier

We tweaked the architecture of sentence classifier. For instance, we experimented with 1-LSTM layer and 2-LSTM layers. Dropout was used in all the variants. Additionally, to capture semantics in a better way we also tried Bi-directional LSTM and Attention mechanism. Both of them achieve different purposes. Bi-directional LSTM helps to take better decisions by looking at the information present at both ends simultaneously whereas attention mechanism helps to take better decision by selecting relevant information only. We tried two different attention mechanisms: *Luong attention* and *Sigmoid* attention. We note that the best result for classification task was achieved when we used 2-LSTM layers with Sigmoid attention.

- Word Embedding : word2vec trained on sentence-question pairs
- Input Feature (x) : Sequence of word2vec embedding, Output Label (y) : 0/1
- Hyperparameters: w2v dim: 100, sentence sequence padding: 20

Accuracies for all the architectures can be seen below.

Table 1: Sentence Classifier

Architecture	Accuracy
1-LSTM layer with dropout followed by Dense layer	65.86 %
2-LSTM layer with dropout followed by Dense layer	67.25 %
Bidirectional LSTM layer with dropout followed by Dense layer	66.45 %
2-LSTM layer with Luong Attention followed by Dropout and dense layer	68.73 %
2-LSTM layer with sigmoid Attention followed by Dropout and dense layer	71.30 %

Question Generation

For Question Generation model also, we experimented with various hyperparameters keeping some of them (word embedding dimension, batch size, beam search width) fixed. Two metrics to measure performance were used: *Perplexity* and *BLEU*. Dropout was always used during training. Initially the number of hidden units in LSTM were 128 and all 2-layers were used, all of them unidirectonal. But as we increased the number of hidden units to 256, performance of model increased. However, further increasing it worsened the performance. Therefore, 256 hidden units were used in subsequent architectures. Introducing dropout during test time also resulted in better performance. Further, it also allows variation in the output generated. Finally, using Bi-directional LSTM gave the best performance (we note that it was trained for only 10 epochs whereas all the other settings were trained for 15 epochs).

- Word Embedding : Pre-trained GloVe vectors (40B, 300 dim) for sentence and questions.
- Hyperparameters: Beam Search Width - 3, Batch Size - 64, 10-15 Epochs, Optimizer - SGD

Table 2: Question Generation

LSTM Units	Dropout		Bidirectional LSTM	Dev PPl	Dev Bleu	Test PPl	Test Bleu
	Dev	Test					
128	0.25	X	No	110.89	3.8	117.41	3.7
256	0.3	X	No	32.74	6.0	14.62	9.2
300	0.3	X	No	39.37	5.3	14.52	8.9
256	0.3	0.3	No	41.69	5.1	41.17	5.1
256	0.3	0.1	No	43.90	4.9	14.63	X
256	0.3	0.1	Yes	36.60	5.8	14.06	9.5

Results

We show the output of both the parts here. In the output of sentence selection, if a line is underlined with green colour it means that line is a probable candidate sentence and our classifier has correctly classified it as candidate sentence. If it is coloured in red, it means that it is not a candidate sentence but our classifier has wrongly classified it as candidate sentence. If a line is not underlined then it is probably not a candidate sentence and our classifier has correctly not classified it as candidate sentence.
For the question generation part, we show the input sentence labeled as *Sentence* and the output of our question generation model, labeled with *Question*.

Sentence Selection

- A pub /pŏ28cb/, or public house is, despite its name, a private house, but is called a public house because it is licensed to sell alcohol to the general public. It is a drinking establishment in Britain, Ireland, New Zealand, Australia, Canada, Denmark and New England. In many places, especially in villages, a pub can be the focal point of the community. The writings of Samuel Pepys describe the pub as the heart of England.

- Historically, pubs have been socially and culturally distinct from caf00e9s, bars and German beer halls. Most pubs offer a range of beers, wines, spirits, and soft drinks and snacks. Traditionally the windows of town pubs were of smoked or frosted glass to obscure the clientele from the street but from the 1990s onwards, there has been a move towards clear glass, in keeping with brighter interiors.

- The inhabitants of the British Isles have been drinking ale since the Bronze Age, but it was with the arrival of the Roman Empire in its shores in the 1st Century, and the construction of the Roman road networks that the first inns, called tabernae, in which travellers could obtain refreshment began to appear. After the departure of Roman authority in the 5th Century and the fall of the Romano-British kingdoms, the Anglo-Saxons established alehouses that grew out of domestic dwellings, the Anglo-Saxon alewife would put a green bush up on a pole to let people know her brew was ready. These alehouses quickly evolved into meeting houses for the folk to socially congregate, gossip and arrange mutual help within their communities. Herein lies the origin of the modern public house, or P̈ubäs it is colloquially called in England. They rapidly spread across the Kingdom, becoming so commonplace that in 965 King Edgar decreed that there should be no more than one alehouse per village.

Question Generation

Model 4 (LSTM Units : 256, Dev Dropout : 0.3, Test Dropout : 0.3, Bidirectional : No)

- **Sentence :** the vedas , the earliest texts on indian philosophy and hindu philosophy dating back to the late 2nd millennium bc , describe ancient hindu cosmology , in which the universe goes through repeated cycles of creation , destruction and rebirth , with each cycle lasting 4,320 million years .
  Question : how long ago did the vedas first appear

- **Sentence :** a new project for la defense , called hermitage plaza , launched in 2009 , proposes to build two towers , 85 and 86 stories or 320 metres high , which would be the tallest buildings in the european union , just slightly shorter than the eiffel tower .
  Question : what is the tallest building in the european union ?

- **Sentence :** malaria kills more residents ; 9 % of the population have reported infection , it causes three times as many deaths as aids .
  Question : what percentage of the population has malaria malaria ?

Model 5 (LSTM Units : 256, Dev Dropout : 0.3, Test Dropout : 0.1, Bidirectional : No)

- **Sentence :** the city struck fortune in the late 18th century with the inventions and industrial activity of eli whitney , a yale graduate who remained in new haven to develop the cotton gin and establish a gun-manufacturing factory in the northern part of the city near the hamden town line.
  Question 1 : what was the name of the business that was founded in new haven ?
  Question 2 : what was the name of the inventor who founded new haven ?

- **Sentence :** there were 13 finalists this season , but two were eliminated in the first result show of the finals.
  Question : how many contestants were eliminated in the first season ?

- **Sentence :** he was told wheat had to be dehusked through a laborious process and at the age of 12 , bell built a homemade device that combined rotating paddles with sets of nail brushes , creating a simple dehusking machine that was put into operation and used steadily for a number of years .
  Question : what was bell ' s profession ?

Model 6 (LSTM Units : 256, Dev Dropout : 0.3, Test Dropout : 0.1, Bidirectional : Yes)

- **Sentence :** this tibetan regime or government is also referred to as the ganden phodrang .
  Question : what is the name of the <unk> <unk> ?

- **Sentence :** the most significant event between the 7th and 11th century was the tripartite struggle centered on kannauj that lasted for more than two centuries between the pala empire , rashtrakuta empire , and gurjara pratihara empire .
  Question : when was the tripartite war ?
- **Sentence :** the western coast of australia had been discovered for Europeans by the dutch explorer willem jansz in 1606 and was later named new Holland by the dutch east India company , but there was no attempt to colonise it .
  Question : what was the name of the dutch dutch explorer ?

Observations

We have presented a fully data-driven neural networks approach to automatic question generation for text. In sentence selection, the encoder for tokens of sentence worked better for stacked LSTM rather than bi-directional LSTM, which is consider as state of the art. The "sigmoid" attention gave better classification accuracy than "loung" attention. The model is able to generate consistent and quality question from text. The model sometimes also generates questions fluent in syntax and semantics but has no relation with the context.Coming to the quality of questions, it is observed that question generation is better for short input sentences but deteriorates for long sentences.We used attention mechanism which helped to generate better output as compared to approaches without attention. In architecture, we observed that bi-directional Encoder produces better results as compared to Unidirectional Encoder.

Future Work

Here we point out several interesting future research directions. Future work involves training model in end-to-end fashion in order to minimize the losses. The process of sentence selection can be made abstractive. To yield better results we can encode whole paragraph to get paragraph level information for question generation model. We can extend our idea of question generation and build a single model for the complete problem. we can encode sentences into context vector and use those context vector as features to train deep RNN to get auto encoder model which generates questions directly from paragraphs.

Contribution

Name	Contribution
Akshar Anand	17
Akshay Paswan	17
Ankur Kumar	17
Meena Vikas	16
Shahzeb Haider	17
Sudhir Kumar	16

References

[1] X. Du, J. Shao, and C. Cardie, "Learning to ask: Neural question generation for reading comprehension," *arXiv preprint arXiv:1705.00106*, 2017.

[2] X. Du and C. Cardie, "Identifying where to focus in reading comprehension for neural question generation," *Cornell University*, 2017.

[3] I. V. Serban, A. García-Durán, C. Gulcehre, S. Ahn, S. Chandar, A. Courville, and Y. Bengio, "Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus," *arXiv preprint arXiv:1603.06807*, 2016.

[4] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, "Microsoft coco captions: Data collection and evaluation server," *arXiv preprint arXiv:1504.00325*, 2015.

[5] V. R. Ming Liu and L. Liu, "Automatic chinese factual question generation," *IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES, VOL. 10, NO. 2*, 2017.

[6] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[7] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, pp. 3104–3112, 2014.

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