Project Report
Machine Comprehension using Commonsense Knowledge

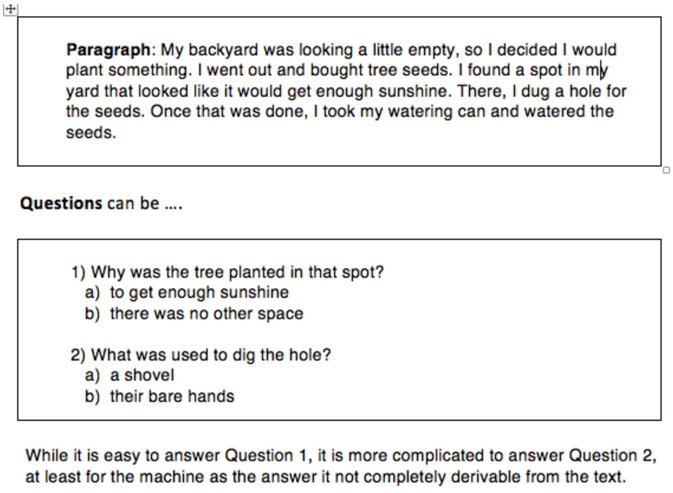
Lakshmimanaswitha Chimakurthi , Apoorva Kasoju, Romil Rathi

I. Introduction

Machine Reading Comprehension (MRC) has become a spotlight topic with advances in natural language processing field. Machine comprehension evaluates a machines understanding by posing a series of reading comprehension questions from associated texts. One important problem is how these systems employ commonsense in real-life reading comprehension tasks to build adaptable Natural Language Processing(NLP) systems.

Description: Given a short paragraph about the narrative texts about everyday activities and several following questions about the paragraph, we are required to build a system to solve the question by choosing the correct answer from two candidate choices. To answer questions that require commonsense reasoning and whose answers might not be readily available even after the syntactic and semantic understanding of the posed question, we incorporate common-sense knowledge bases.

An example can be



Paragraph: My backyard was looking a little empty, so I decided I would plant something. I went out and bought tree seeds. I found a spot in my yard that looked like it would get enough sunshine. There, I dug a hole for the seeds. Once that was done, I took my watering can and watered the seeds.

Questions can be ….

1) Why was the tree planted in that spot?
   a) to get enough sunshine
   b) there was no other space

2) What was used to dig the hole?
   a) a shovel
   b) their bare hands

While it is easy to answer Question 1, it is more complicated to answer Question 2, at least for the machine as the answer it not completely derivable from the text.

Fig. 1. Example from DataSet

II. Related Work:

Traditional work in machine comprehension has been in building models to predict answer given context paragraph and question.

In a paper, Wang and Jiang[1] used an end-to-end neural network method that uses a Match-LSTM to model the question and the passage In tasks related to assessing machine comprehension , a classifier based machine comprehension system with baseline and syntactic features is used to compute score for each candidate answer

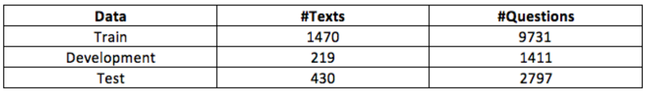
In the paper introducing the MCTest Dataset[2], a sliding window method that matches the words in the window with the bag of words constructed from the question and a candidate answer.

In one of the work, assumption that there exists hidden structure between the question, correct answer, and text, a latent structural SVM (LS SVM)[3] is trained to learn the latent answer- entailing structures that helps answer questions about a text.

However, they did not focus on answering the question by incorporating some type of commonsense knowledge such as Script knowledge, they tried to match the answer from the text. In one of the recent works Hongyu Lin1, Le Sun and Xianpei Han tried reasoning with Heterogeneous Commonsense knowledge for Machine Comprehension[4] i.e. they used 1)Event narrative knowledge, which captures temporal and causal relations between events 2)Entity semantic knowledge, which captures semantic relations between entities 3)Sentiment coherent knowledge, which captures sentimental coherence between elements to solve this task by Deep Learning techniques .

III. Methodology:

Data: MCScript[2] is a collection of narrative texts with questions of various types referring to these texts, and pairs of answer candidates for each question. It comprises of approx. 2,100 texts and over 14,000 questions. A substantial subset of questions require knowledge beyond the facts mentioned in the text, i.e. they require inference using commonsense knowledge about everyday activities



Data	#Texts	#Questions
Train	1470	9731
Development	219	1411
Test	430	2797

Fig. 2. Description of MCScript Dataset

The question type composition of the dataset is as follows:

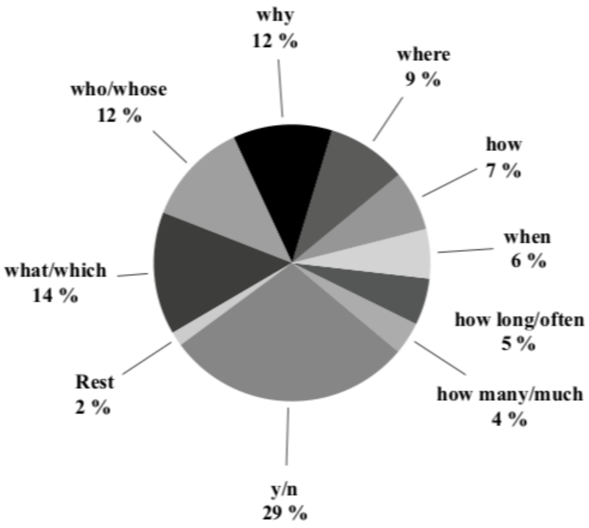


Fig. 3. Composition of Question Types in MCScript

In order to incorporate commonsense reasoning , we have used DeScript[7] ,a large-scale crowdsourced collection of explicit linguistic descriptions of script-specific event sequences (40 scenarios with 100 sequences each) that includes scenarios requiring simple general knowledge (e.g. BORROWING A BOOK FROM THE LIBRARY) to more complex ones that require some amount of expert knowledge (e.g. RENOVATE HOUSE)

Data Pre-processing: Pre-processing procedure can be listed as follows :
(i) Replacement of abbreviated characters
(ii) Removal of meaningless characters and stop words
(iii) Word tokenization using NLTK package
From Dataset, a list of text_id, question_id and answer_id was extracted.

The data is in the form of triplets: context, question and answer-index span (Answer along with initial and final indices of the answer within the context passage). XML files from official dataset and MCScript were parsed into six text files one with instance_id, instances and one with instance_id, question_id, question, answerspan on each line. To get accurate vector representation of words,industry-standard pre-trained word embeddings from Glove are used with embedding size as 300.

Methods:

[a] Baseline: Word Match We first use a simple word matching baseline, by selecting the answer that has the highest literal overlap with the original paragraph/text and score accordingly. If the answer matches exactly to the text, then the score assigned was the length of the longest answer which was 28 for our dataset, else for each word match, count was increased by 1. The intuition was simple human instinct i.e. we tried to find word to word similarity between answer and text

[b] Using Using topics as knowledge: From the baseline , we understood that a mere word match does not provide great performance when either of the candidate answers does not have their text matching that of original paragraph, leading to a score of zero for each candidate answer. So to overcome such problems, approach used in this model is not to use any external knowledge like commonsense knowledge bases but to use latent topics inferred from paragraphs/texts as background knowledge. We used BigARTM for topic modeling as it provides additive regularization and it is known to combine well very different objectives, including sparsing, smoothing, topics decorrelation and many others. BatchVectorizer from BigARTM[6] is applied to segregate paragraphs into topics and answer is chosen from original paragraph using word match and if not found in the original text, similar paragraphs under same topic are considered. The model was then trained with 15 passes through the collection of texts.

[c] Using DeScript as Commonsense Knowledge: As the given paragraphs refer to everyday scenarios,it makes sense to use script knowledge that contains event sequences describing typical human actions in an everyday situation.One of such script knowledge sources is DeScript , which covers most basic scenarios. After preprocessing and combining events that belong to same topic from the DeScript[7] files , most frequent words both in original paragraph(based on overall word frequency for text and TFIDF for topics) and each DeScript topic are found .The method for choosing the answer was the same as in the previous part except that instead of using similar paragraphs/texts from the same topic as knowledge source , DeScript paraphrases were used. Cosine similarity between the given text and each DeScript topic was calculated to find most suitable events.

[d] Using Bi-LSTM with Attention : Since Reading Comprehension can be viewed as mapping sequence of words representing question to that of answer, a technique like LSTM that learns to map sequences and find representations that captures meaning is useful.

Although LSTM does great job in capturing long dependencies,Bidirectional - LSTM[5] takes this capability further by considering both the previous and future context for processing. Usually machine comprehension use attention to focus important parts in story when predicting answer for a question.Since our task is to assess candidate answers ,using attention for question and answer is informative in choosing right answer among candidate answers.A quick glance of the neural architecture for this model is below

Pre-trained embeddings for each of paragraph, question and answer are fed into three separate Bi-LSTM encoders. Attention is applied to Bi-LSTM encoded paragraph - question and question - answer to obtain paragraph-aware question representation and question-aware answer representation respectively. The attention is calculated by taking the outer-

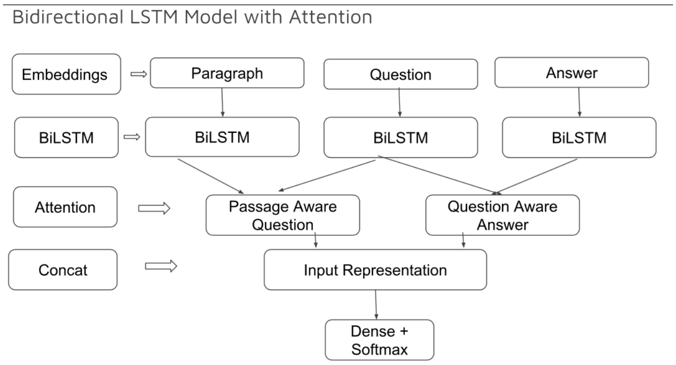
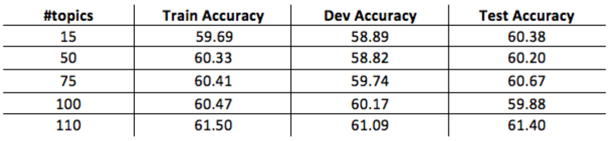


Fig. 4. Architecture for Bi-LSTM model for Machine Comprehension Task

product of (context question) and (question answer) i.e we are taking all the combinations of words in the context with all words These representations are concatenated and fed into a soft max layer which outputs the predicted probabilities for each candidate answer.

IV. Experiments

Setup and Parameter Tuning: For the method based on topic clustering using BigARTM ,we tried with different number of topics that paragraphs/texts must be divided into , ranging from 15 to 110 and measured performance by metric accuracy.



#topics	Train Accuracy	Dev Accuracy	Test Accuracy
15	59.69	58.89	60.38
50	60.33	58.82	60.20
75	60.41	59.74	60.67
100	60.47	60.17	59.88
110	61.50	61.09	61.40

Fig. 5. Results for different number of topics

We can see that as the number of topics increase , so are topic clusters becoming more specific as noise is being reduced and further train and dev accuracies improving.

The reason behind the dip in test accuracy for 100 topics is while some answers get an advantage with background knowledge from topics , for others in case answer is not derivable from original paragraph/text , because the model finds similar paragraphs with same topic , which might contain general-purpose language leading to chosen answer being incorrect.At the end we have chosen the number of topics to be 15, because this parameter value resulted in the least number of topics that could identified correctly given a word from everyday scenario.

For Bi-LSTM based model we used deep learning library Keras.

For training, as loss function we use categorical cross entropy and we output predicted probabilities for each candidate answer. The optimizer we use is Adam optimizer and trained for 6 epochs each . The paragraph ,question, answer sizes are all set to 60. In order to improve performance of the model,we tried with different dropout rates ranging from 0.3 to 0.5 and best test accuracy is achieved at 0.4.

Results and Discussions:

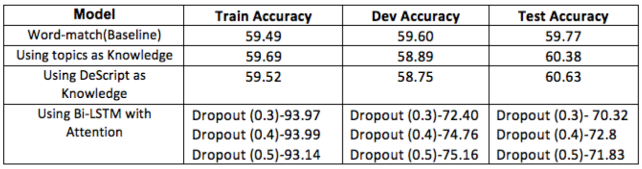
Word Match (Baseline): The features used were simply the words in context and answer. The low accuracy is mostly due to the nature of correct answers in our data: Each correct answer has a low overlap with the text by design. Since the overlap model selects the answer with a high overlap to the text, it does not perform well. Additionally, it explains the very bad result on text-based questions

Using topics as Knowledge: As this model still employs the word match logic from baseline method ,just extending the match process to similar paragraphs/texts as of original paragraph/text. For questions with 'Yes' or 'No' as answers, a simple not found in original/similar texts might be interpreted as no even when not in texts might completely be unrelated, leading to incorrect answer. It seems that finding answers by exact word matching might work well for lengthy answers.

Using DeScript as Commonsense Knowledge: In this method, the features considered were TFIDF and word frequency for text. This model resulted in slight increase in test accuracy from previous model based on topic clustering using BIGARTM . This could be due to the fact that DeScript events are short and clearly expressed, possibly eliminating noise present in paragraphs/texts clustered with BigARTM. Further , using cosine similarity between most frequent words in original paragraph as well as DeScript topic is also better measure when comparing texts.

Bi-LSTM Model: The Bi-LSTM could give substantial improvements over word match baseline model by more than 12 percent on test set and even more on train. Each epoch took approx 10 minutes to run. The features we considered for this model are Glove embeddings for words in text, we didnt embed features like POS tags, NER tags etc in our model which could capture better semantic and syntactic relations in the text, thus we would try to incorporate them in our model and improve our current model.

Here are the final results :



Model	Train Accuracy	Dev Accuracy	Test Accuracy
Word-match(Baseline)	59.49	59.60	59.77
Using topics as Knowledge	59.69	58.89	60.38
Using DeScript as Knowledge	59.52	58.75	60.63
Using Bi-LSTM with Attention	Dropout (0.3)-93.97 Dropout (0.4)-93.99 Dropout (0.5)-93.14	Dropout (0.3)-72.40 Dropout (0.4)-74.76 Dropout (0.5)-75.16	Dropout (0.3)- 70.32 Dropout (0.4)-72.8 Dropout (0.5)-71.83

Fig. 6. Results obtained from all methods

References

[1] Qiaojing Yan, Yixin Wang. Classifier Based Machine Comprehension https://nlp.stanford.edu/courses/cs224n/2015/reports/22.pdf

[2] Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater and Manfred Pinkal. MCScript: A Novel Dataset for Assessing Machine Comprehension Using Script Knowledge https://arxiv.org/pdf/1803.05223.pdf

[3] Mrinmaya Sachan, Avinava Dubey, Eric P. Xing and Matthew Richardson. Learning Answer-Entailing Structures for Machine Comprehension https://pdfs.semanticscholar.org/f26e/088bc4659a9b7fce28b6604d26de779bcf93.pdf

[4] Hongyu Lin, Le Sun and Xianpei Han. Reasoning with Heterogeneous Knowledge for Commonsense Machine Comprehension. http://aclweb.org/anthology/D17-1216

[5] Reza Ghaeini, Sadid A. Hasan, Vivek Datla, Joey Liu, Kathy Lee, Ashequl Qadir, Yuan Ling, Aaditya Prakash, Xiaoli Z. Fern1 and Oladimeji Farri. DR-BiLSTM: Dependent Reading Bidirectional LSTM for Natural Language Inference https://arxiv.org/pdf/1802.05577.pdf

[6] Konstantin Vorontsov, Oleksandr Frei, Murat Apishev, Peter Romov, and Marina Dudarenko. BigARTM: Open Source Library for Regularized Multimodal Topic Modeling of Large Collections http://www.machinelearning.ru/wiki/images/e/ea/Voron15aist.pdf

[7] Lilian D. A. Wanzare, Alessandra Zarcone, Stefan Thater, Manfred Pinkal. 2016. DeScript A Crowdsourced Corpus for the Acquisition of High Quality Script Knowledge. http://www.lrecconf.org/proceedings/lrec2016/pdf/913Paper.pdf