11-747 Neural Networks for NLP
Survey of Machine Reading Comprehension & Baseline Implementation

Anirudha Rayasam
Carnegie Mellon University
arayasam@andrew.cmu.edu

Anusha Kamath
Carnegie Mellon University
akamath1@andrew.cmu.edu

Gabriel Bayomi Tinoco Kalejaiye
Carnegie Mellon University
gbayomi@cs.cmu.edu

1 INTRODUCTION

Teaching a computer to read and comprehend human languages is a challenging task for machines as this requires the understanding of natural language and the ability to reason over various clues. The task of Machine Reading Comprehension (MRC) is a useful benchmark to demonstrate true understanding of natural language. In recent years, several datasets have been created to focus on answering questions as a way to evaluate machine comprehension. The machine is first presented with a piece of text such as a news article or a story and is then expected to answer one or multiple questions related to the text. There are multiple forms of Machine Comprehension tasks: multiple choice questions, where the correct answer is to be chosen from a set of provided candidate answers; filling the blanks, where words are chosen from the passage to complete the statement; snippet extraction, which requires extraction of snippets from the given passage to answer the questions; another setup involves synthesizing sentences with words from within the passage or a combination of words from the passage and the question. The synthesized answers can also contain words that are not present in both. The goal of the Machine comprehension task is to build an agent that can answer questions posed by humans in natural language.

In this survey we briefly discuss different strategies and the various neural architectures that have been developed for the task of Machine reading comprehension. Numerous variants ranging from Matching Networks, Memory Networks to single and multi turn Reasoning Networks are covered in the following sections with a brief description of each. We also go over the reinforcement learning and transfer learning approaches that have been used in machine comprehension. Finally, the line of work in MRC called Commonsense Machine Comprehension which exploits common sense knowledge.

2 RELATED WORK

Traditionally, a pipeline of NLP models has been used for attempting question answering. These models make heavy use of linguistic annotations, structured world knowledge and semantic parsing and other similar NLP tasks. For a while, lexical matching with logistic regression was used for the task of machine comprehension [17]. Classical QA methods use a set of documents as an abstraction of memory, and information retrieval methods to find answers [6]. A technique known as frame semantic parsing attempts to identify predicates and their arguments, allowing models access to information about “who did what to whom" [5]. This is used to extract entity-predicate triples from both the query and context document and these outputs are used for the subsequent rule based query resolution. Despite being successful these models lack the capability to scale for large datasets and it has been proven difficult to build end-to-end expressive models using these techniques. These shortcomings have been countered through incorporation of neural architectures, which have become prevalent due to the recent advancement in generation of vast supervised data for these tasks and enhanced computational abilities.

Neural approaches

A lot of the work in Neural Machine comprehension focuses on how to extract the required information from the given passage. This has been achieved with boundary extraction models, span prediction models [11], Memory networks and reasoning models. The MRC task here involves a question $\mathbf{Q} = \{q_0, q_1, ..., q_{m-1}\}$ and a passage $\mathbf{P} = \{p_0, p_1, ..., p_{n-1}\}$ and aims to find an answer span $\mathbf{A} = \{a_{start}, a_{end}\}$ in P. The assumption is that the answer or at least the part of the answer exists in the passage P as a contiguous text string. A typical neural architecture consists of an embedding, encoding, interaction- and answer layer.

One of the network architectures used for this task is the R-net [23] proposed by Microsoft research, which reads the question and passage, aligns and matches the question and the passage to find supporting clues and then compares different answer candidates based on the aggregated evidence to determine the answer boundaries in the passage. They use a gated attention-based recurrent network, which adds an additional gate to the standard attention networks to account for the fact that words in the passage are of different importance to answer a particular question. In addition to this they use a self-matching mechanism, which can effectively aggregate evidence from the whole passage to infer the answer. This refines passage representation with information from the whole passage and overcomes the fact that LSTMs can loose out on long distance information despite their theoretical capabilities. However, the model is incapable of capturing complex inferences which span across multiple sentences and depend on structural dependencies.

An example for the boundary extraction model is the Match-LSTM network. The match-LSTM proposed by [22] goes through passage tokens sequentially and, at each position of the passage, produces an attention weighted representation of the question. This weighted representation of the question is then concatenated with a vector representation of the current passage token and fed into an

LSTM, which they refer to as the match-LSTM. The match-LSTM essentially sequentially aggregates the matching of the attention-weighted question to each token of the passage and uses the aggregated matching result to make a final prediction. This model may not be as effective when the answer is not a subset of the input text and also the generalizability of it to other datasets is in doubt.

The models presented above assume answers to be exact spans in the input passage. In most cases this might not be a good solution for MRC and there is a need of generating additional text not included in the passage or the question, augmenting the provided information when required. This idea of using synthesis in addition to selecting spans within provided passage has been explored in the S-net architecture by [19] which will be used as the state-of-the-art baseline for our task. In addition to finding spans within the passage, the S-net architecture also uses passage ranking to improve their evidence extraction. Post evidence extraction, the model uses the passage, question and aggregated evidence to synthesize answers that can contain words from the passage, question and words that are not included in either.

The work discussed above and most state-of-the-art machine reading systems are built on supervised training data, trained end-to-end on data examples, containing not only the articles but also manually labeled questions about articles and corresponding answers. However, for many domains, this supervised training data does not exist. To deal with this problem, [7] used a transfer learning approach for producing a scalable solution for the task. Their network, SynNet, produces both questions and answers from articles in new domains based on experience in previous domains, and uses these materials as training data to perform reading comprehension in the new domain. Though SynNet is not a direct contender for the state-of-the-art race, this work is a good move in the direction improving performance of MRC task in other lesser know datasets.

Another commonly used architecture for MRC are the memory networks [26]. Memory networks have four gating mechanisms (input, output, generalization and response) to convert inputs to internal feature representation, update memories given the new input, computes output features from the new input and memory and decodes output features to generate a response. In this MRC setup the Input Module takes an input text which is a statement or fact or question to be answered and saves it in the next available memory slot. The Generalization Module is only used to store this new memory, so old memories are not updated. The core of inference lies in the Output and Response Modules. The Output Module produces output features by finding K supporting memories for the given input. Finally, the Response module needs to produce a textual response which in its simplest form returns the memory text retrieved. The Output Module first scores all memories, i.e., all previously seen sentences, against the input to retrieve the most relevant fact and then uses the current memory to search again to find the next relevant fact. When the input is at the word level instead of the sentence level an additional segmentation function has to be learnt.

Different variants of the memory network architecture have been used in reading comprehension, most of which combine a memory component with an LSTM model for inference [21]. A variant of this model is the weakly supervised memory network. The memory network described above needs the supporting memories for a given question in the training data. This in itself is prohibitive to applying the MemNN on other general QA tasks because each dataset will need to be annotated with the supporting statements for each question. In the weakly supervised variant, for each input sentence, a corresponding memory is computed by summing up the embeddings of all the words in the sentence. The question is embedded using another matrix B and a match score is calculate between the question and each memory. Each memory vector has a corresponding output vector c_i, computed using another embedding matrix C. The output vector from the memory O is then a sum over the c_i, weighted by the probability vector from the input. Many layers can be stacked to produce a W matrix which is used to predict the answer from the output and question embedding.

Another variant of the Memory Network models are the Dynamic memory networks [10]. This architecture has an input module and the question module that encodes raw text representations. The output representations are fed into the episodic Memory Module, and forms the basis, or initial state, upon which the episodic Memory Module iterates. Episodic memory module chooses which parts of the inputs to focus on through the attention mechanism and produces a “memory" vector representation taking into account the question as well as the previous memory. At each iteration the episodic memory module has the ability to retrieve new information, in the form of input representations.

Attention-over-Attention neural networks proposed by [4] also focus on the MRC task. This model aims to place another attention mechanism over the existing document-level attention. Instead of using heuristic merging functions they use a mutual attention mechanism from query to the document and document to the query. They calculate pair-wise matching score between each document and query word, forming a matrix, where the matching score is the dot product of the word embeddings. After getting the pair-wise matching matrix M, they apply a column-wise softmax function to get probability distributions in each column, where each column is an individual document-level attention when considering a single query word. Now instead of using summing or averaging to combine these individual attentions into a final attention, they introduce another attention mechanism to automatically decide the importance of each individual attention. They apply a row-wise softmax function to the pair-wise matching matrix M to get query-level attentions. The query level attentions are averaged and thereby used to compute attended document level attention which is then used to synthesize answers.

An interesting line of work in this task involves using single or multiple turns of reasoning to effectively exploit the relation among queries, documents, and answers. Single turn reasoning models utilize an attention mechanism to emphasize some sections of a document which are relevant to a query. This can be thought of as treating some parts as unimportant while focusing

on other important ones to find the most probable answer. Single turn reasoning models like the one proposed by [8] uses the attentive reader and the impatient reader models where the Attentive Reader is able to focus on the passages of a context document that are most likely to inform the answer to the query. The impatient reader model further has the ability to reread from the document as each query token is read. [20] propose the EpiReader model which uses two neural network structures: one extracts candidates using the attention-sum reader; the other re-ranks candidates based on a bilinear term similarity score calculated from query and passage representations.

The multi turn reasoning models [16] are based on the idea that human readers often revisit the given document in order to perform deeper inference after reading a document. This mechanism produces a new query glimpse and document glimpse in each iteration and utilizes them alternatively in the next iteration. By reading documents and enriching the query in an iterative fashion, multi-turn reasoning has demonstrated their superior performance consistently. Taking this further, the Reasonet[16] includes a reinforcement learning based stopping criterion for the reasoning depth (number of turns in multi-turn reasoning) instead of using a fixed number of iterations. Analogous to the work by [16] the MUlti-Strategy Inference for Comprehension (MUSIC) model [27] incorporates multi-step inference. Additionally, it is capable of key comprehension skills like handling rich variations in question types, understanding potential answer choices and also able to draw inference through multiple sentences by applying different attention strategies to different types of questions on the fly.

Several other MRC models have embraced this kind of multistep strategy, where predictions are generated after making multiple passes through the same text and integrating intermediate information in the process. The Stochastic Answer Networks proposed by [13] use a multi-step answer network, that searches over all the outputs of the multi-step reasoning. Unlike other multi-step reasoning models, which only uses a single output at the last step or some dynamically determined final step, this module employs all the outputs of multiple step reasoning. They apply dropout [18], to avoid bias problems where models places too much emphasis on one particular step's predictions and forces the model to produce good predictions at every individual step.

A Fast QA approach is discussed in [24]. They have a Type Matching mechanism following the embedding layer which extracts the span in the question that refers to the expected, lexical answer type (LAT) by extracting either the question word(s) like who, when, why, how, how many, etc. or the first noun phrase of the question after the question words "what" or "which" (e.g., "what year did..."). They encode the LAT by concatenating the embedding of the first and last word together with the average embedding of all words within the LAT. Following this, they have a context matching mechanism between the question and the answer span and finally they score these individual answer spans. RNN-based Fast-QA system turns out to be an efficient neural baseline architecture for extractive question answering. They essentially focus on the awareness of question words while processing the context and a composition function that goes beyond simple bag-of-words modeling.

The last and very intuitive line of work uses common sense knowledge along with the comprehension text to generate answers. Common-sense knowledge increases the accuracy of machine comprehension systems. The challenge is to find a way to include this additional data and improve the system's performance. There are many possible common-sense knowledge sources. Generally, script knowledge which is sequences of events that describe typical human actions in an everyday situations is used. One of such script knowledge sources is DeScript. Texts from the same topic clusters are used as a source of common-sense knowledge, additionally DeScript paraphrases can also be used. The work by [12] focuses on reasoning with heterogeneous commonsense knowledge. They use three kinds of common sense knowledge: causal relations, semantic relations (like co-reference and association) and, finally, sentiment knowledge - (sentiment coherence, positivity and negativity) between two elements. In human reasoning process, not all inference rules have the same possibility to be applied, because the scale of reasonability of the inference is will impact the likelihood of usage. They use attention to weigh the inferences based on the nature of the rule and the given context. Their attention mechanism models the possibility that an inference rule is applied during the inference from a premise document to a hypothesis by considering the relatedness between elements and knowledge category, as well as the relatedness between two elements. They answer the comprehension task by summarizing over all valid inference rules.

3 DATASETS

There are many interesting datasets for the task of machine reading comprehension. For instance, the SQuAD (Stanford Question Answering Dataset) dataset [15] is one of the most popular options. It consists of a set of question and answers, where each question has varied responses by different crowdworkers. It means that it's possible to assign rankings for answers instead of only using boolean variables, making it similar to real world environments where not necessarily there is a single unique solution for a question. On the other hand, the training passages are short, with approximately five sentences each, and it restricts the answer to be a specific span of the passage, which is not realistic on real-life settings.

Another important dataset is MS-Marco (A Human Generated MAchine Reading COmprehension Dataset) [14]: a large scale, real world and human sourced QA dataset. Although it holds a lot of similarity with the SQuAD dataset, there are two main differences between the two structures: MS-Marco does not constrain the answer to be in a specific span of the passage, and MS-Marco provides multiple passages for an answer and not a specific one. These are important differences, it means that MS-Marco is closer to a real-life setting in terms of a reading comprehension task. However, it also means that it involves a more complicated task and, likely, a more complex model to understand these nuances.

MCTest (Open-Domain Machine Comprehension of Text) [1] is another dataset from Microsoft. It is constituted of more than 600 short stories. For each of the stories, there are 4 human-annotated

Table 1: Examples of some common error patterns observed in the output of our model and their corresponding targets

	Model generated output and corresponding target from test samples	Error Class
Target	['one', 'to', 'three', 'years']	***REP***
Output	['years', 'years', 'years', 'years']	
Target	['yes']	***Other***
Output	['no']	
Target	['in', 'map', 'it', 'shows', 'that', 'nicaragua', 'is', 'located', 'in', 'the', 'central', 'america']	***INF, REP***
Output	['in', 'the', 'is', 'along', 'is', 'and', 'located', 'in', 'the', 'south', 'and']	
Target	['regulation', 'of', 'healthy', 'hormones', 'in', 'the', 'body']	***INF, REP***
Output	['regulation', 'of', 'thyroid', 'thyroid', 'triiodothyronine', 'the', 'body']	
Target	['bob', 'dylan']	***UNK***
Output	['UNK', 'dylan']	
Target	['law', 'enforcement']	***INF***
Output	['the', 'forensic']	
Target	['to', 'control', 'the', 'saliva', 'released', 'into', 'the', 'area', 'of', 'the', 'mouth', 'that', 'is', 'just', 'under', 'the', 'tongue']	***INF, REP***
Output	['this', 'stimulates', 'the', 'body', 'externally', 'the', 'body', 'the', 'body', 'is', 'a', 'in', 'like', ' 'body']	
Target	['sugar', 'phosphate']	***INF***
Output	['ribose']	

Table 2: Some answers generated by our model and their corresponding targets from the test-set :)

	Cherry picked examples from our model
Target	['median', 'paralegal', 'salary', 'is']
Output	['average', 'paralegal', 'salary', 'of']
Target	['mesothelioma', 'is', 'a', 'rare', 'form', 'of', 'cancer']
Output	['It', 'is', 'a', 'rare', 'form', 'of', 'cancer']
Target	['audio', 'speakers']
Output	['speakers']
Target	['in', 'UNK', 'wisconsin']
Output	['in', 'wisconsin']
Target	['represented', 'the', 'engine', 'code']
Output	['represented', 'the', 'locomotive', 'code']
Target	['per', 'year']
Output	['every', 'year']
Target	['yes']
Output	['yes']

questions. For each one of the questions, there are 4 candidate answers. It's specially challenging due to the fact that the dataset and the vocabulary is relatively small. It's specially interesting for settings where not necessarily a lot of data is available and it's hard to rely on more traditional Deep Learning models.

CNN/Daily Mail QA dataset [3] was released by Google DeepMind and it is another popular choice for comprehension tasks. It is slightly different than the previous datasets: all the queries are *Cloze* type questions, meaning goal is to fill the blanks of the entities in a passage. It contains over 90K of of CNN news and 200K Daily Mail news, all with approximately 4 queries per story. Different from MCTest, the vocabulary and dataset is large. Although the information is guaranteed to be in the query (they are formulated based on highlights of the news), there might be more than 500 entities on a story, making it extremely challenging.

It's also important to mention a few datasets with slightly different conceptualizations, but that have been also tackled by the Machine Comprehension community. The bAbI QA tasks [25] use a synthesized dataset where the text is generated from a simulation consisting of a few actors, objects and places. The questions in this task are of basic factoid type with a single supporting fact, since the answer depends on information from a single sentence of text. This is the standard benchmark for memory based QA models. The CBT (Children's Book Test) [9] comes from the same overall bAbi Project from Facebook Research. The goal of the CBT task is to fill the removed information: each story in this dataset is formed of 21 consecutive sentences. The first 20 sentences form the context, and a word is removed from the 21st sentence, which becomes the query. The tasks from the bAbi Project are definitely popular, but aim to solve different research questions than the previous ones (MS-Marco, SQuAD, etc).

This group of datasets represent the majority of the relevant recent work of Machine Comprehension [28]. It's important to notice that although each one of them contains a group of advantages and disadvantages compared to each other or to a real-life setting, they are all extremely useful in terms of answering important research questions about modeling natural language and understanding nuance for textual information.

4 BASELINE MODEL

The model incorporated for this checkpoint is a standard attention driven encoder-decoder architecture similar to the one shown in Figure 1, but we do not incorporate the highway networks as the current model is not very deep. The detailed architecture we used for the task is shown in Figure 2. The model has a question encoder and a passage encoder which passes the inputs through an embedding layer and a subsequent encoding Bidirectional GRU.

The last hidden state of the encoded question along with the previous decoder hidden state is used to get an attention weighted representation of the passage embedding. The resulting context vector is then passed through the decoder GRU. In order to obtain the output vocabulary distribution for each decoder time step, we concatenate the word embedding from the previous time step with the context vector and pass it through a softmax activated linear layer. We are using a masked cross entropy loss function to train the network. Masking of the passage is crucial here because the passage lengths vary a lot. The masked cross entropy loss will take care of the length distributions when evaluating the loss.
The baseline model architecture is presented in Figure 2 for reference.

4.1 Experiments and Result analysis

Our baseline paper (S-net) uses Bleu-1 and Rogue-L metrics for evaluating the model. We obtained an average Bleu score of 31.9 and a Rogue score of 31 on the MS-Marco train-set. On the test-set we obtain an average Bleu score of 20.8 and Rouge score of 20.4. The tabulated results can be found in Table 3. Some of the results for other reigning models on the MS-MARCO test dataset are also summarized in Table 4. The checkpoint model performance is far from ideal as compared to the leading models. We do a simple study on the errors encountered in the model output and observe that there are some common recurring patterns which are prevalent as seen in Table 1. We also present some obligatory cherry-picked samples from our model in Table 2.

Our basic model fails in the following scenarios:

- [***UNK***] Like in most limited vocabulary systems, rare key words in the sentence are replaced by UNK tokens.
- [***REP***] The network tends to produce repeated words in the synthesized sentence.
- [***INF***] Many outputs face the more pressing problem in comprehension, of not being able to extract the right information from the passage.

Modifications that might help remedying the above problems:

- We incorporate attention in this model. We can use the attention scores to replace the UNK produced at the output with the source word that has the highest attention at the time step.
- Repeated words are generally resolved by employing a coverage mechanism. We can keep track of visited words by adding the attention scores at all time steps and penalizing revisiting the words multiple times by means of a coverage loss. Instead, we can also use the simple and effective solution of dropping words that are sequentially repeated more than twice at the output. However this might create problems in some situations.
- The passages in MS-Marco dataset are quite long and unlike general comprehension tasks, these passages extend over multiple paragraphs. Even after employing an attention mechanism it is hard for the neural network to focus on the right sentences. Hence our baseline paper employs an extraction followed by synthesis setup where the sentences relevant to the question at hand are extracted and used in subsequent synthesis. Using self-attention and a gating mechanism when encoding the passage might also help in fixing this problem.

Table 3: Bleu-1 and Rogue-L scores of our checkpoint model on the MS-Marco train and test sets

Scores/Dataset	Train-set	Test-set
Checkpoint1 Bleu-1	31.9	20.8
Checkpoint1 Rogue-L	31.05	20.4

Table 4: Bleu-1 and Rogue-L scores of the leading models on the MS-Marco test set

Models/Scores	Bleu-1	Rouge-L
V-Net	46.15	44.46
S-Net	45.23	43.78
R-Net	42.89	42.22
ReasonNet	38.81	39.86
Prediction	37.33	40.72
FastQA_Ext	33.67	33.93
FastQA	32.09	33.99

5 FUTURE WORK

We plan to implement the S-net architecture for checkpoint 2. The architecture will have two key components, the first for evidence extraction followed by another for answer synthesis. The evidence extraction will be implemented as a pointer network and the extracted evidence sentences will be used alongside the question for synthesis. The synthesis network has a similar architecture to the one we have used for this checkpoint. The networks are trained separately. The targets for the first network are extracted from within the passage by finding sentences that have the best Rogue scores with the provided answers. The target for the synthesis network are the answers themselves.

Post this checkpoint we would like to extend our model by incorporating the promising components of Baidu's V-Net which recently outperformed S-Net on the MS-MARCO dataset. However, their work has not been made public yet and we are currently unaware of the techniques they employed. Other improvements to the architecture we would like to explore includes gating mechanism to selectively weigh each passage word, highway connections if we resort to using a deeper architecture, self attention over the passage to be able to better represent all information in the passage especially for long passages.

6 APPENDIX

In this section we have included the general architecture for the attention driven encoder-decoder neural architecture for reference, along with a detailed diagram of our checkpoint model implementation. Additionally, we have also presented some attention activation regions in the passage to serve as reference and validation for the attention module. The attention regions have been cropped to highlight the activated regions in the large passage.

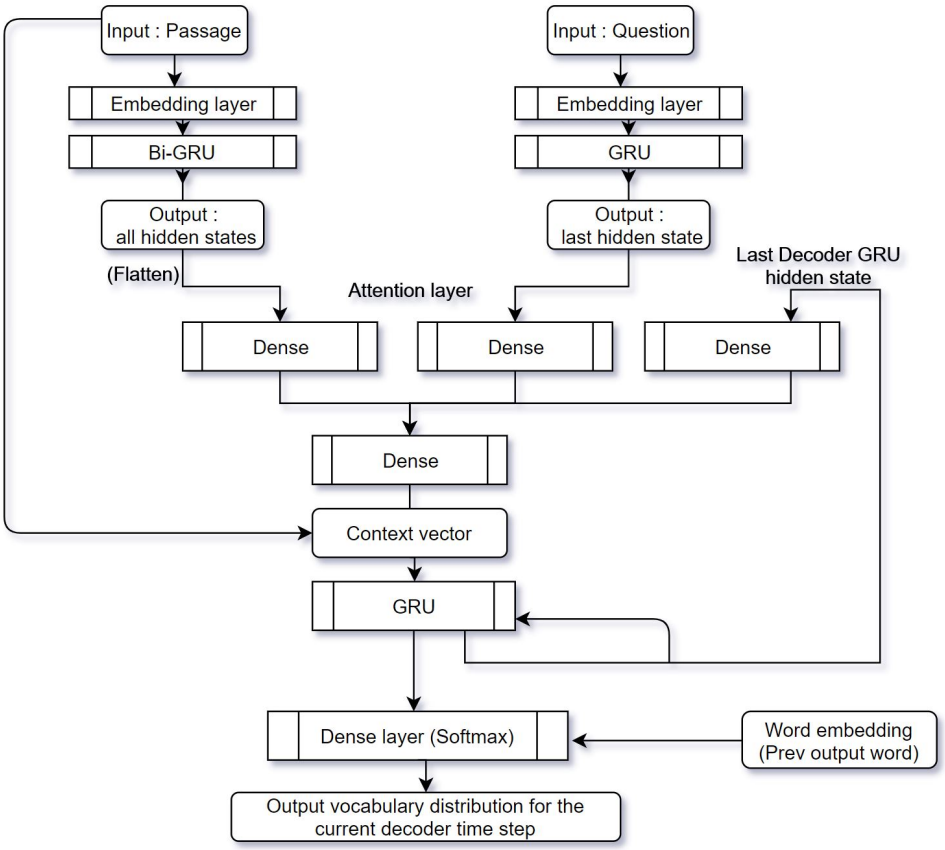

Figure 1: Model architecture used in Checkpoint 1

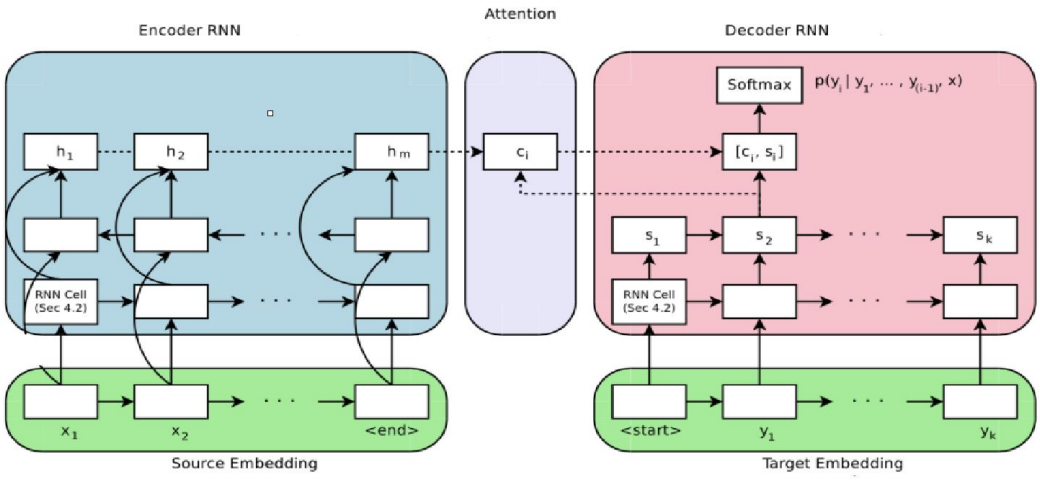

Figure 2: A standard encoder decoder architecture with attention [2]. This model also includes highway connections in the encoder, while we do not incorporate this in our checkpoint model.

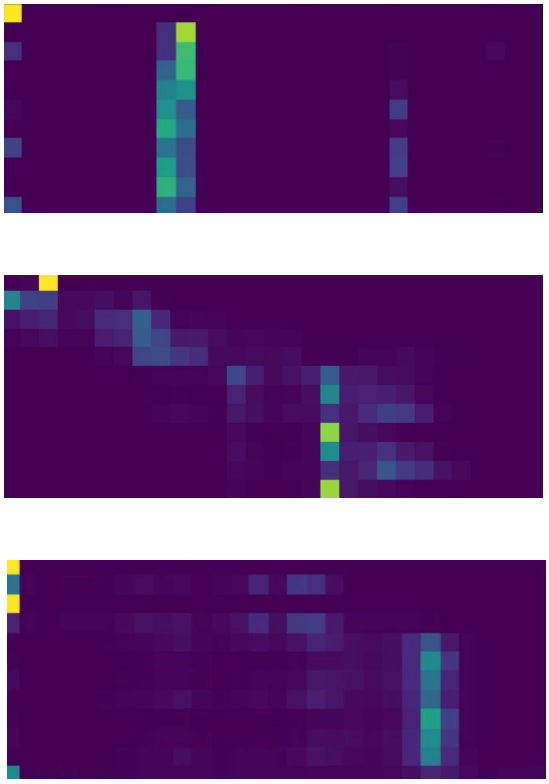

Figure 3: Cropped attention activated region (Cropped because of the very long passage size)

REFERENCES

[1] 2013. *MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text.* https://www.microsoft.com/en-us/research/publication/mctest-challenge-dataset-open-domain-machine-comprehension-text/

[2] Jason BrownleeC. 2018. How to Configure an Encoder-Decoder Model for Neural Machine Translation. (2018). https://machinelearningmastery.com/configure-encoder-decoder-model-neural-machine-translation

[3] Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task. *CoRR* abs/1606.02858 (2016). arXiv:1606.02858 http://arxiv.org/abs/1606.02858

[4] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2016. Attention-over-Attention Neural Networks for Reading Comprehension. *CoRR* abs/1607.04423 (2016). arXiv:1607.04423 http://arxiv.org/abs/1607.04423

[5] Dipanjan Das, Desai Chen, André FT Martins, Nathan Schneider, and Noah A Smith. 2014. Frame-semantic parsing. *Computational linguistics* 40, 1 (2014), 9–56.

[6] David Ferrucci. 2010. Build Watson: an overview of DeepQA for the Jeopardy! challenge. In *Parallel Architectures and Compilation Techniques (PACT), 2010 19th International Conference on*. IEEE, 1–1.

[7] David Golub, Po-Sen Huang, Xiaodong He, and Li Deng. 2017. Two-Stage Synthesis Networks for Transfer Learning in Machine Comprehension. *CoRR* abs/1706.09789 (2017). arXiv:1706.09789 http://arxiv.org/abs/1706.09789

[8] Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. *CoRR* abs/1506.03340 (2015). arXiv:1506.03340 http://arxiv.org/abs/1506.03340

[9] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. *CoRR* abs/1511.02301 (2015). arXiv:1511.02301 http://arxiv.org/abs/1511.02301

[10] Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2015. Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. *CoRR* abs/1506.07285 (2015). arXiv:1506.07285 http://arxiv.org/abs/1506.07285

[11] Kenton Lee, Tom Kwiatkowski, Ankur P. Parikh, and Dipanjan Das. 2016. Learning Recurrent Span Representations for Extractive Question Answering. *CoRR* abs/1611.01436 (2016). arXiv:1611.01436 http://arxiv.org/abs/1611.01436

[12] Hongyu Lin, Le Sun, and Xianpei Han. 2017. Reasoning with Heterogeneous Knowledge for Commonsense Machine Comprehension. (01 2017), 2032-2043 pages.

[13] Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2017. Stochastic Answer Networks for Machine Reading Comprehension. *arXiv preprint arXiv:1712.03556* (2017).

[14] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *CoRR* abs/1611.09268 (2016). arXiv:1611.09268 http://arxiv.org/abs/1611.09268

[15] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. *CoRR* abs/1606.05250 (2016). arXiv:1606.05250 http://arxiv.org/abs/1606.05250

[16] Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. 2016. ReasoNet: Learning to Stop Reading in Machine Comprehension. *CoRR* abs/1609.05284 (2016). arXiv:1609.05284 http://arxiv.org/abs/1609.05284

[17] Ellery Smith, Nicola Greco, Matko Bosnjak, and Andreas Vlachos. 2015. A strong lexical matching method for the machine comprehension test. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1693–1698.

[18] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[19] Chuanqi Tan, Furu Wei, Nan Yang, Weifeng Lv, and Ming Zhou. 2017. S-Net: From Answer Extraction to Answer Generation for Machine Reading Comprehension. *CoRR* abs/1706.04815 (2017). arXiv:1706.04815 http://arxiv.org/abs/1706.04815

[20] Adam Trischler, Zheng Ye, Xingdi Yuan, and Kaheer Suleman. 2016. Natural Language Comprehension with the EpiReader. *CoRR* abs/1606.02270 (2016). arXiv:1606.02270 http://arxiv.org/abs/1606.02270

[21] Shuohang Wang and Jing Jiang. 2015. Learning natural language inference with LSTM. *arXiv preprint arXiv:1512.08849* (2015).

[22] Shuohang Wang and Jing Jiang. 2016. Machine Comprehension Using Match-LSTM and Answer Pointer. *CoRR* abs/1608.07905 (2016). arXiv:1608.07905 http://arxiv.org/abs/1608.07905

[23] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 189–198.

[24] Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Making Neural QA as Simple as Possible but not Simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Association for Computational Linguistics, 271–280. https://doi.org/10.18653/v1/K17-1028

[25] Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. *CoRR* abs/1502.05698 (2015). arXiv:1502.05698 http://arxiv.org/abs/1502.05698

[26] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory Networks. *CoRR* abs/1410.3916 (2014). arXiv:1410.3916 http://arxiv.org/abs/1410.3916

[27] Yichong Xu, Jingjing Liu, Jianfeng Gao, Yelong Shen, and Xiaodong Liu. 2017. Towards Human-level Machine Reading Comprehension: Reasoning and Inference with Multiple Strategies. (11 2017).

[28] Eric Yuan. 2015. Compare Among Popular Machine Reading Comprehension Datasets. *Blog* (2015). http://eric-yuan.me/compare-popular-mrc-datasets/