Generative Multi-Hop Question Answering with Compositional Attention Networks

Ammar Alqatari
Computer Science
ammarq@stanford.edu

Abstract

Current state-of-the-art models in question answering tasks fail at achieving comparable accuracies in so-called multi-hop settings, which require composing multiple pieces of evidence from a context document to arrive at the correct answer. To overcome the limitations in standard attention-based approaches, I adapt the recently developed compositional attention network (MAC) architecture, originally implemented for visual question answering, and test its performance on the HotpotQA dataset for interpretable multi-hop question answering. The trained model exceeds the performance of provided baseline models and approaches state-of-the-art performance.

A custom project under the mentorship of Drew Hudson (dorarad@cs.stanford.edu).

1 Introduction

Attention-based recurrent neural network architectures have shown great success in extractive question answering tasks, reaching better-than-human accuracies on datasets such as the Stanford Question Answering Dataset (SQuAD) [8]. However, the models' success in SQuAD is largely due to their capability of finding patterns in the question and locating a match at a local context in the supporting document, rather than a deeper comprehension or reasoning ability on the text [14]. Predictably, the standard approaches which have succeeded in extractive fact-finding QA dataests fail to achieve comparable accuracies in multi-hop QA, which involves generating an answer to a given question by combining several pieces of evidence from a given context. Succeeding at multi-hop QA requires that the model be able to direct attention to multiple parts of both the question and the context, then compose the attended components to arrive at the answer.

Multi-hop QA is becoming an increasingly relevant benchmark task for evaluating machine reading comprehension (MRC) at a higher-level than offered by extractive fact-finding QA datasets. Several multi-hop QA datasets have been developed in recent years to test MRC capability. The bAbI dataset features several multi-hop QA tasks on synthetically generated examples which include questions that can be answered with one, two, or three and more supporting facts; yes/no questions; and argument relation questions [1] . QAngaroo provides multi-hop QA examples in a more natural language setting with longer context, though it is extractive rather than generative since the answer to every question can always be found in the context [2].

In this paper, I focus on the recently developed HotpotQA dataset [17], which consists of a larger number of wikipedia context passages, along with crowd-sourced questions and answers design to test specific reasoning and comprehension patterns. HotpotQA can be more challenging than bAbI and QAngaroo because its answers are generative rather than being multiple-choice, and the examples come from a curated, human-generated source. An additional feature in HotpotQA is the annotation of each answer with the sentences in the context document relevant to inferring the answer, a feature designed to further test the comprehension and explainability of the model.

CS224N: Natural Language Processing with Deep Learning, Winter 2019, Stanford University, CA.

A recently developed model, the compositional attention network (MAC) [15], has an architecture particularly well-suited to the task. The MAC network uses a memory-augmented architecture which models its computations as a series of reasoning steps. It is designed, in general, to learn to perform given queries based on composing information from a provided knowledge base. The network's demonstrated success in a visual question answering task, which requires compositional reasoning on a question and an image, gives promise that the network will perform similarly well in the multi-hop question answering task, which requires composition over a question and a corresponding context document. Additionally, its design facilitates interpretability and enables explanation of the reasoning process behind a given answer.

I adapt the MAC network architecture and train a model which I apply to the HotpotQA dataset. The model achieves an exact match accuracy of 51.5% and F1 score of 61.26, improving on the provided baseline by 3% and 2%, respectively.

2 Related Work

2.1 Multi-hop question answering

It has been shown that architectures which perform well on previous QA tasks, such as the bi-directional attention flow (BIDAF) network, do not perform as well in multi-hop question answering [3]. Datasets such as QAngaroo and HotpotQA which have very long context documents and disjointed evidence sentences have called for a new kind of architecture which can collect such disjointed evidence and synthesize it into an answer. A variety of new approaches have been proposed which vary from end-to-end neural network approaches to ones which use information retrieval based methods.

In [5], the authors make use of multi-attention mechanisms to perform the reasoning steps along with a pointer-generator decoder to synthesize the answer. The pointer generator decoder, built in the original paper for a summarization task, is suited for answer generation due to its ability of both producing tokens from the provided context as well as generating relevant new tokens [6]. The authors apply their model to NarrativeQA [7], a QA dataset based on understanding stories or complex narratives, achieving state-of-the-art results on the dataset. Query reduction networks [4] also take an end-to-end approach to multi-hop QA, performing multi-step reasoning by reducing the original query to a presumably more direct query as the network comes across new context information. QRNs are similar to the MAC network in the way they encode local information from the context document in a string of cells. The model is currently the best-performing on the bAbI conversational QA task.

In [9], the authors opt for a hierarchical approach which performs coarse-grained reasoning on documents then a fine-grained module which scores answer candidates, making use of self-attention and co-attention for each module. The model achieves state-of-the-art results on QAngaroo.

2.2 Memory-augmented networks

Compositional attention networks belong to a class of neural networks which make use of an external memory and interface with it through a control. Memory networks have been applied successfully to many kinds of NLP and question answering tasks [10] [11] [12] [13]. The external memory aids the neural network in keeping track of the long-term dependencies which exist across long documents. The MAC network makes novel use of memory augmentation by stringing an arbitrary number of cells each of which has its own memory and control states, rather than having a global external memory which can be obfuscated or overwritten by different parts of the input [15]. Memory-augmented networks, such as the dynamic neural turing machine, have shown success in QA and MRC tasks, and achieved near state-of-the-art results in datasets including bAbI.

3 Approach

3.1 Model Overview

The compositional attention network is a memory-augmented neural network architecture which introduces a novel "memory, attention, and composition" (MAC) cell which decomposes a recurrent computation into a "a series of attention-based reasoning steps". The model performs a computation

over a knowledge base K (an image in the original paper, and a document in this task), and a task description q (a question in this task) to produce an output.

In general, the model is composed of an input unit, a MAC recurrent network, and an output unit. The input unit transforms the raw question string and knowledge base into vector representations which can be fed to the recurrent component. The recurrent component consists of a string of p MAC cells, each consisting of hidden control and memory states. Finally, the output unit transforms the distributed vector representation outputted by the recurrent component into the desired output form.

The original model is made available by the authors on github [1]. Since the model is built for a VQA task, which takes an image as the knowledge base input and produces a a one-word answer as output, I had to make modifications to the model to train on this task. For the purposes of this project, I had to replace the original input and output units in the model, and make some small modifications to the MAC component.

3.2 Input Unit

Each HotpotQA example consists of a 10-paragraph document (a string), a question string, answer string, and a list of supporting facts (sentence ids). I modify the preprocessor from the original model to process each example and build a vocabulary of all tokens present in either a document, question, or answer (without punctuation). I use shared embeddings for all vocabulary tokens across documents, questions, and answers, which are pre-trained through GLoVe [16].

I replace the CNN-based network for processing the knowledge base with a 3-layer bi-directional RNN-LSTM encoder followed by a linear projection of the final hidden state of the RNN into the dimension of the MAC control state. The projection is used as the input to the MAC component.

3.3 Output Unit

Because answers in the original dataset consist of a single-word, the output unit in the original model is a fully connected classification network. Answers in HotpotQA are strings of arbitrary length, with up to 15 words per answer. I replace the original output unit with a one-layer, uni-directional RNN-LSTM variable-length decoder. The decoder's initial state is the final memory state from the last mac cell, and takes as input the sentence embedding of the question. The loss is then computed through softmax cross-entropy between the resulting sequence and ground truth labels.

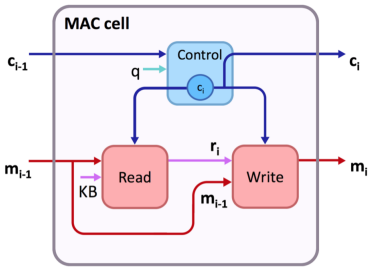


Figure 1: MAC cell architecure [15]

3.3.1 MAC Component

The MAC component consists of a string of p MAC cells. A larger number of cells generally leads to better but less interpretable results, and a smaller number of cells leads to the converse outcome. Each cell consists of a hidden control state c_i and a memory state m_i, which interact through 3 operational units: a control unit, a read unit, and a write unit.

The control is a soft-attention based weighted average of question words $cw_s; s = 1, ..., S$, and corresponds to a reasoning step to be performed by the cell based on a part of the question.

The memory is similarly a weighted average over the regions of the knowledge base (context document) $k_i; i = 1, ..., K$.

[1] https://github.com/stanfordnlp/mac-network

The control unit performs a reasoning step i by attending to a certain part of the question and updating the control state c_i. It computes a linear combination of the question representation q_i and current control state c_{i-1} to produce $cq_i = W^{d\times 2d}[c_{i-1}, q_i] + b^d$. This combination is then projected into the space of question words cw_i as $ca_{i,s} = W^{1,d}(cq_i \odot cw_s) + b^1$. Finally, the new control state is computed through a softmax operation on $ca_i : c_i = \sum_{s=1}^{S} \text{softmax}(ca_{i,s}) \cdot cw_s$.

The read unit performs the current reasoning step through a two-step attention process to extract the necessary information from the knowledge base. The model first finds relevant regions j of the knowledge base through a direct computation based on the current memory state m_{i-1} and a knowledge-base element k_j: $I_{i,j} = [W_m^{d\times d} m_{i-1} + b_m^d] \odot [W_k^{d\times d} k_j + b_k^d]$. k_j is then concatenated and linearly combined with $I_{i,j}$ to produce $I'_{i,j}$. We then measure the similarity of the control state c_i to the interactions $I'_{i,j}$, producing $ra_{i,j} = W^{d\times d}(c_i \odot I'_{i,j}) + b^d$. Finally, the a weighted distribution over knowledge-base regions is obtained through a softmax operation on $ra : r_i = \sum_{j=1}^{K} \text{softmax}(ra_{i,j})) \cdot k_j$.

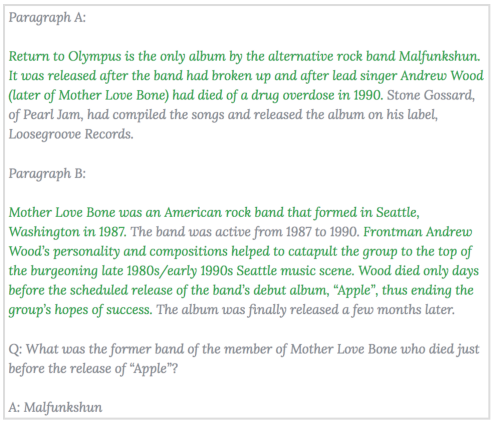
The write unit computes the result of the ith reasoning step and correspondingly updates the memory state. The newly extracted information r_i is combined with the prior memory state m_i through a linear transformation, resulting in $m_i = W^{d\times 2d}[r_i, m_{i-1}] + b^d$.

4 Experiments

4.1 Dataset and Evaluation

I evaluate the model on the recently introduced HotpotQA dataset, designed for "diverse, explainable, multi-hop question answering" [17]. The dataset consists of a quality-controlled collection of crowd-sourced questions and answers based on passages from related Wikipedia articles. Each example is also associated with a set of sentences from the passage which constitute evidence for the answer. The main task is to predict the answer given a question and a context passage. An additional task measures the justification ability of the model by asking for it to provide supporting sentences as part of the output. There are two settings to the dataset: a `distractor` setting, in which the context passage consists of 10-paragraphs, and a `fullwiki` where the context is the entire Wikipedia corpus. In this project, I focus solely on the `distractor` setting of the dataset, and do not yet tackle the justification task.

The data consists of around 90k training examples, which the authors classify into easy single-hop questions (18k), medium multi-hop questions (56k), and hard (15k) multi-hop questions. The dev and test sets consist of 7k hard multi-hop questions each.


Paragraph A:

Return to Olympus is the only album by the alternative rock band Malfunkshun. It was released after the band had broken up and after lead singer Andrew Wood (later of Mother Love Bone) had died of a drug overdose in 1990. Stone Gossard, of Pearl Jam, had compiled the songs and released the album on his label, Loosegroove Records.

Paragraph B:

Mother Love Bone was an American rock band that formed in Seattle, Washington in 1987. The band was active from 1987 to 1990. Frontman Andrew Wood's personality and compositions helped to catapult the group to the top of the burgeoning late 1980s/early 1990s Seattle music scene. Wood died only days before the scheduled release of the band's debut album, "Apple", thus ending the group's hopes of success. The album was finally released a few months later.

Q: What was the former band of the member of Mother Love Bone who died just before the release of "Apple"?

A: Malfunkshun

Figure 2: Example question from HotpotQA [17]

The authors also provide a baseline model, evaluation metrics, and a leaderboard for the dataset. The baseline model is an attention-based model which uses character-level models, self-attention, and bi-attention and achieves nearly state-of-the-art accuracy on SQuAD. They evaluate the model based

on exact match (EM) accuracy and the F1 score of the model's predicted answers. EM is defined as the number of predicted answers which exactly match the label, and F1 is the average over all examples of the harmonic mean of precision and accuracy of each predicted answer.

4.2 Hyperparameters and Training

I train the model using the Adam optimizer to minimize cross-entropy loss. I trained several models with a small grid search over the learning rate, number of MAC cells, hidden encoding dimension, and number of RNN layers. I use early stopping to stop the models if they show only small improvement after an entire epoch; since the number of examples is very large, most models stop training after 4 to 5 epochs, which takes around 15 hours of training. In the following section, I report the result of the best performing model, which made use of 16 MAC cells and a hidden dimension of 512.

4.3 Results

The best-performing model achieved an EM score of 51.5% and F1 score of 61.26 on the dev set. The model already exceeds the performance of the baseline model developed by the authors of HotpotQA, which achieves an EM score of 45.60% and an F1 score of 59.02 (albeit on the test set).

The current state-of-the art result on the leaderboard achieves an EM score of 55.84% and F1 score of 69.69.

	Model	Code	Ans		Sup		Joint	
			EM	F_1	EM	F_1	EM	F_1
1 Nov 21, 2018	QFE (single model) *NTT Media Intelligence Laboratories*		53.86	68.06	**57.75**	**84.49**	**34.63**	**59.61**
2 Mar 4, 2019	GRN (single model) *Anonymous*		52.92	66.71	52.37	84.11	31.77	58.47
3 Mar 1, 2019	DFGN + BERT (single model) *Anonymous*		55.17	68.49	49.85	81.06	31.87	58.23
4 Mar 4, 2019	BERT Plus (single model) *CIS Lab*		**55.84**	**69.76**	42.88	80.74	27.13	58.23
5 Oct 10, 2018	Baseline Model (single model) *Carnegie Mellon University, Stanford University, & Universite de Montreal* (Yang, Qi, Zhang, et al. 2018)		45.60	59.02	20.32	64.49	10.83	40.16

Figure 3: Current HotpotQA leaderboard

5 Analysis

5.1 Different versions of the model

Among all trained models, the best performing model had the largest number of MAC cells, the largest encoding dimension, and largest number of RNN layers in the encoder. Models with larger encoding dimensions lead to out-of-memory errors, while models with a larger number of RNN layers were much slower to train. This result suggests that with more computational power, a model with larger parameters could achieve an even better performance.

5.2 Leaderboard Models

The top-performing leaderboard models make use of BERT. Since my developed model makes use of pre-trained word embeddings but not contextual embeddings, I expect that incorporating contextual embeddings will improve the model.

6 Conclusion

Multi-hop question answering presents a new frontier in machine reading comprehension tasks which requires new approaches which are simple and explainable. I show that the MAC network, which is simple, efficient, and easily interpretable, achieves good performance on the multi-hop question answering task. Its success on the HotpotQA dataset calls for further testing on other machine comprehension tasks which require compositional reasoning, such as conversational question answering, to further show the architecture's robustness and versetality. It also suggests promise to exploring variants of memory-augmented networks and their effectiveness in various MRC tasks.

A key continuation point of this project is evaluating the network's selection of supplementary facts, which is not included in this project due to time limitations. An addition to the network could be the incorporation of the supplementary fact data into the model to directly learn the attention mappings through strong supervision during training. Other possible additions to the model include integrating contextual embeddings to the input unit, using a pointer-generator decoder model in the output unit, and adding modules which can extend the network to perform on the `fullwiki` setting of HotpotQA.

References

[1] Weston, J., Bordes, A., Chopra, S., Rush, A. M., van Merriënboer, B., Joulin, A., Mikolov, T. (2015). Towards ai-complete question answering: A set of prerequisite toy tasks. arXiv preprint arXiv:1502.05698.

[2] Welbl, J., Stenetorp, P., Riedel, S. (2018). Constructing datasets for multi-hop reading comprehension across documents. Transactions of the Association of Computational Linguistics, 6, 287-302.

[3] Seo, M., Kembhavi, A., Farhadi, A., Hajishirzi, H. (2016). Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:1611.01603.

[4] Seo, M., Min, S., Farhadi, A., Hajishirzi, H. (2016). Query-reduction networks for question answering. arXiv preprint arXiv:1606.04582.

[5] Bauer, L., Wang, Y., Bansal, M. (2018). Commonsense for Generative Multi-Hop Question Answering Tasks. arXiv preprint arXiv:1809.06309.

[6] See, A., Liu, P. J., Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368.

[7] Kočiský, T., Schwarz, J., Blunsom, P., Dyer, C., Hermann, K. M., Melis, G., Grefenstette, E. (2018). The narrativeqa reading comprehension challenge. Transactions of the Association of Computational Linguistics, 6, 317-328.

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

[9] Zhong, V., Xiong, C., Keskar, N. S., Socher, R. (2019). Coarse-grain Fine-grain Coattention Network for Multi-evidence Question Answering. arXiv preprint arXiv:1901.00603.

[10] Weston, J., Chopra, S., Bordes, A. (2014). Memory networks. arXiv preprint arXiv:1410.3916.

[11] Gulcehre, C., Chandar, S., Cho, K., Bengio, Y. (2016). Dynamic neural turing machine with soft and hard addressing schemes. arXiv preprint arXiv:1607.00036.

[12] Anything, A. M. (2015). Dynamic memory networks for natural language processing. Kumar et al. arXiv Pre-Print.

[13] Na, S., Lee, S., Kim, J., Kim, G. (2017). A read-write memory network for movie story understanding. In Proceedings of the IEEE International Conference on Computer Vision (pp. 677-685).

[14] Qi, P., & Chen, D. (2019, February 26). Beyond Local Pattern Matching: Recent Advances in Machine Reading. Retrieved from http://ai.stanford.edu/blog/beyond_local_pattern_matching/

[15] Hudson, D. A., & Manning, C. D. (2018). Compositional attention networks for machine reasoning. arXiv preprint arXiv:1803.03067.

[16] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation

[17] Yang, Zhilin, et al. "Hotpotqa: A dataset for diverse, explainable multi-hop question answering." arXiv preprint arXiv:1809.09600 (2018).