# An Effective Method to Answer Multi-hop Questions by Single-hop QA System

Kong Yuntao, Nguyen Minh Phuong, Teeradaj Racharak, Tung Le and Nguyen Le Minh
*School of Information Science, Japan Advanced Institute of Science and Technology, Nomi, Japan*

Keywords: Question Answering, Multi-hop QA, Two-step Tuning, Transfer Learning, Multi-task Learning.

Abstract: Multi-hop question answering (QA) requires a model to aggregate information from multiple paragraphs to predict the answer. Recent research on multi-hop QA has attempted this task by utilizing graph neural networks (GNNs) with sophisticated graph structures. While such models can achieve good performance, their computation is rather expensive. In this paper, we explore an alternative method that leverages a single-hop QA model to deal with multi-hop questions. Our system called '**A**nswer **M**ulti-hop questions by **S**ingle-hop QA' (**AMS**) consists of three main parts that first filter a document and then conduct prediction using the attention-based single-hop QA model with multi-task learning. Specifically, **AMS** is constructed based on the co-attention and self-attention architecture. Lastly, consider that BERT-based model is pre-trained in a general domain and the data distribution can be different from multi-hop QA task. We propose two-step tuning mechanism to enhance the model's performance, which is based on transfer learning from other QA datasets. To verify **AMS** effectiveness, we consider the previous state-of-the-art Hierarchical Graph Network (HGN) with the same document filter as our baseline. Experiments on HotpotQA show that **AMS** can outperform HGN by 1.78 points and 0.56 points for Joint EM and Joint F1, respectively. Meanwhile, it has smaller model's size and uses less computational resource. We also experiment with other GNN-based models and achieve better results.



## 1 INTRODUCTION

As a popular task in Natural Language Processing (NLP), much effort has been made to the development of question answering (QA) systems, due to the release of many large-scale and high-quality datasets such as (Hermann et al., 2015; Rajpurkar et al., 2018a; Joshi et al., 2017). Early on, these datasets mainly concentrate on single-hop questions, in which an answer can be retrieved from a single paragraph and only one fact is involved. With the recent explosion of success of deep learning techniques, QA models such as (Lan et al., 2019; Zhang et al., 2021) have correspondingly improved and have achieved super-human performance, especially in SQuAD 2.0[1]. More recently, multi-hop QA datasets including (Khashabi et al., 2018; Welbl et al., 2018; Yang et al., 2018) have gained increasing attention. These datasets require models to answer a more complicated question by integrating information from multiple paragraphs and facts.

Figure 1 shows an example from HotpotQA (Yang et al., 2018), which is a popular multi-hop QA dataset. Here, given a complex question and a document, the question is the composition of two single-hop sub-questions: (i) 'Who is the author of "Armad"?' (the answer is Ernest Cline) and (ii) 'Which novel by Ernest Cline will be adapted as a feature film by Steven Spielberg?'. The document contains 10 paragraphs but only two paragraphs are related to the question. Models are required to aggregate information from scattered facts across multiple paragraphs, and predict both the answer and supporting facts (i.e., sentences showing evidences of the answer).

Regarding the current research line, there has been a trend of exploiting graph neural network (GNN) for multi-hop QA (Qiu et al., 2019; Fang et al., 2020; Huang and Yang, 2021). Investigation of the graph construction and applying GNN reasoning has been explored. GNN-based models intuitively consider answering multi-hop questions as reasoning process on a document graph. Specifically, the document is first modeled into a graph, and then GNN is applied for information propagation and aggregation. The updated graph state is expected to have the semantics of each

[1]https://rajpurkar.github.io/SQuAD-explorer/



Yuntao, K., Phuong, N., Racharak, T., Le, T. and Minh, N.
An Effective Method to Answer Multi-hop Questions by Single-hop QA System.
DOI: 10.5220/0010824200003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 2*, pages 244-253
ISBN: 978-989-758-547-0; ISSN: 2184-433X

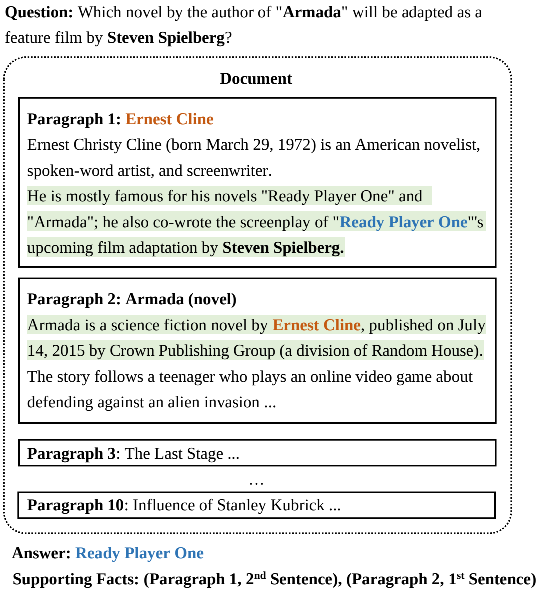


Figure 1: An example from HotpotQA. A document and A compositional question are given. Both the answer and supporting facts ( in green background ) should be predicted.

node with its neighbors, which would be used for the final prediction. However, it has been studied that the computation of GNN is usually expensive and the graph construction strongly depends on prior knowledge (Wu et al., 2021).

Recently, document filters (Qiu et al., 2019; Fang et al., 2020; Tu et al., 2020) are proposed to denoise any document by selecting the most relevant paragraphs inside it. Table 1 shows promising performance of the filter from Hierarchical Graph Network (HGN) (Fang et al., 2020). For 2-paragraph selection, both precision and recall can achieve around 95%. For 4-paragraph selection, recall is nearly 99%. We observe that such performance can effectively neglect irrelevant information while keeping necessary evidences, making it acceptable to utilize single-hop QA model for multi-hop QA.

Table 1: Performance of HGN's document filter.

| Filter | Precision | Recall |
|---|---|---|
| 2-paragraph selection | 94.53 | 94.53 |
| 4-paragraph selection | 49.45 | 98.74 |

Inspired by this, our work proposes an effective method to **A**nswer **M**ulti-hop questions by **S**ingle-hop QA system (**AMS**). We consider HGN (Fang et al., 2020), one of state-of-the-art (SOTA) models, with its document filter as our baseline. Our AMS exploits existing single-hop QA models based on the attention mechanism and integrates with the HGN's document filter. Since the prediction of supporting facts is also required, additional layers are incorporated for related sub-tasks to adapt multi-task learning. Besides, two-step tuning is proposed to enhance model's performance, which is based on transfer learning from other QA datasets. We conduct comprehensive experiments on five datasets to study how two-step tuning impacts on the model's performance. To validate our method, we focus on the HotpotQA dataset distractor setting (Yang et al., 2018). The result shows that AMS can outperform the strong baseline model, and decrease both model's size and computational resource by around 80% and 23%, respectively. Moreover, AMS also outperforms other sophisticated GNN-based models.

To conclude, our contributions are threefolds. First, we propose an effective method (AMS) to answer multi-hop questions, which incorporates single-hop QA models with a document filter. Second, the proposed model outperforms the strong baseline and other sophisticated GNN-based models, while it requires less computational resource. Lastly, we propose a new two-step fine-tuning scheme that can improve the overall performance. We experimentally study its effectiveness with diverse datasets to analyze their effect on the model's performance.

## 2 RELATED WORK

**GNN-based Multi-hop QA.** GNN-based models attempt to construct a graph based on entities or other levels of granularity in text, which could bridge scattered information in different paragraphs. For instance, MHQA-GRN (Song et al., 2018) integrates evidence by constructing an entity-based graph and investigates two GNNs to update graph state. Entity-GCN (De Cao et al., 2019) refines entity-based graphs with different edges representing different relations. HDE-Graph (Tu et al., 2019) constructs a heterogeneous graphs by introducing the entity and document nodes. CogQA (Ding et al., 2019) imitates human reasoning to construct a cognitive graph and predicts both possible answer spans and next-hop answer spans. DFGN (Qiu et al., 2019) proposes a RoBERTa-based document filter to select the most relevant paragraphs and develops a dynamic entity-based graph interacting with context. SAE (Tu et al., 2020) improves the document filter by considering information between paragraphs. HGN (Fang et al., 2020) utilizes Wikipedia's hyperlinks to retrieve more paragraphs and proposes a hierarchical graph consisting of entity, sentence, paragraph and question nodes. BFR-Graph (Huang and Yang, 2021) constructs a weighted graph by relational information and poses restrictions on information propagation to improve the efficiency of graph reasoning.

**No-GNN-based Multi-Hop QA.** There are also attempts to address multi-hop QA by exploiting the existing NLP methods. For instance, Coref-GRU (Dhingra et al., 2018) extracts entities and their coreference from different paragraphs, and aggregates the information by using multi-GRU layers with a gated-attention reader. CFC (Zhong et al., 2019) employs the hierarchical attention to construct the coarse and fine module for two-stage scoring. QFE (Nishida et al., 2019) follows an extractive summarization work and incorporates an additional sentence prediction layer for multi-task learning. C2F Reader (Shao et al., 2020) considers the graph-attention as a special kind of self-attention, and argues that GNN may be unnecessary for multi-hop reasoning. Compared with the above methods, our work takes a step forward to effectively utilize existing single-hop QA models, and shows better performance than sophisticated GNN-based models.

**Fine-tuning for NLP Tasks.** ULMFiT (Howard and Ruder, 2018) proposes the discriminative fine-tuning that employs layer-wise learning rates, and slanted triangular learning rates with a sharp increase and a gradual decrease of the learning rates. Peters et al. (2019) compare the performance of feature extraction and fine-tuning, and demonstrates that the distance between pre-training and the target task can impact on their relative performance. Sun et al. (2019) explores a general scheme to fine-tune BERT for text classification, ranging from in-domain tuning, multi-task learning, to the fine-tuning in the target task. Houlsby et al. (2019) proposes compact adapter modules for the text Transformer. Above works explore general fine-tunning schemes or study on a specific task. However, to the best of our knowledge, there is no work focusing on multi-hop QA.

# 3 PROPOSED MODEL

We select HGN (Fang et al., 2020), which is the SOTA approach for HotpotQA, as our strong baseline. Inspired from HGN, our model is the integration of its document filter and single-hop QA models. In our approach, the document is first denoised by the filter and then is fed into the attention-based single-hop QA model for the sub-tasks prediction and multi-task learning. Figure 2 shows an overview of our model.

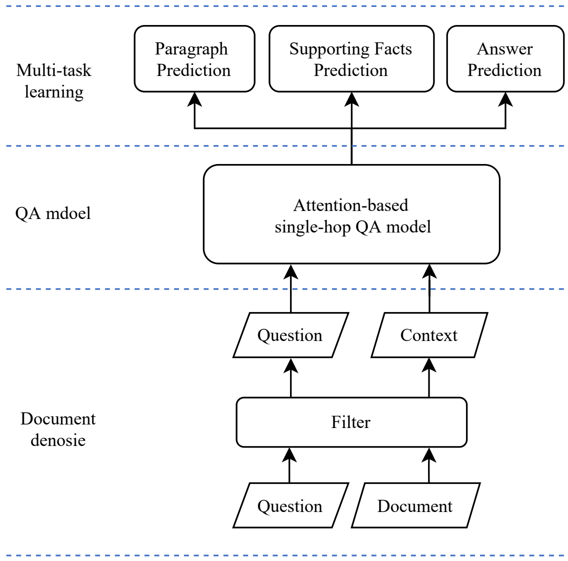


Figure 2: Overview of our model. Answer prediction includes answer span prediction and answer type prediction.

## 3.1 Document Denoise

The filter plays a crucial role in our work and we follow HGN's filter consisting of three components:

- Paragraph Ranker: It is trained based on RoBERTa and followed by a binary classification layer to calculate the probability of whether each paragraph contains supporting facts.
- Title Matching: It searches for paragraphs whose title exactly match any phrase with the question.
- Entity Matching: It searches for paragraphs which contain any entity exactly that appears in the question.

HGN's filter selects paragraphs within two steps. In the first step, it retrieves paragraphs by Title Matching. If multiple paragraphs are returned, two paragraphs yielding the highest score from Paragraph Ranker are selected. If it fails to retrieve any paragraphs, it further searches for paragraphs by Entity Matching. If it also fails, the paragraph yielding the highest score from the Paragraph Ranker is thus selected. In the second step, the filter retrieves additional paragraphs by Wikipedia's hyperlinks from the paragraphs identified by first step.

Table 1 show the performance of the adopted filter. According to the table, we select four paragraphs from the total ten paragraphs since it achieves high recall (98.74%). The retrieved paragraphs are concatenated and used as context. Figure 3 shows the distribution of token length of the context, indicating that around 94% token length is within 500. Such performance can effectively reduce the input length and keep necessary information. At this stage, the output is the question and context denoised from the filter:

$$\mathit{Question}, \mathit{Context} = \mathrm{Filter}(\mathit{Question}, \mathit{Document}) \tag{1}$$

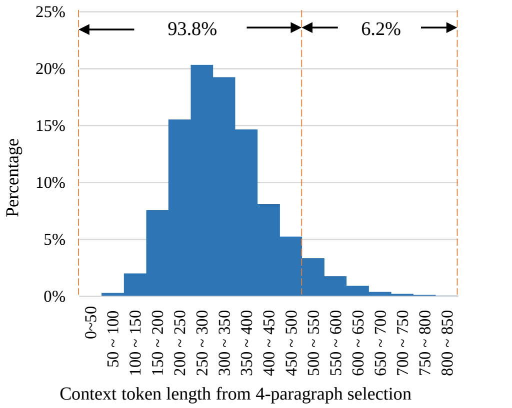


Figure 3: Distribution of context token length from 4-paragraph selection.

## 3.2 QA Model

With the promising performance of the document filter, we propose a single-hop QA model to eliminate the burden of GNN in the multi-hop QA task. Figure 4 illustrates the proposed single-hop QA model architecture, which performs the following steps.

First, it feeds the question and the context into the RoBERTa-large model to obtain question embeddings $\mathbf{E}_q \in \mathbb{R}^{l_q \times d}$ and context embedding $\mathbf{E}_c \in \mathbb{R}^{l_c \times d}$, where $l_c$ and $l_q$ are the length of context and question. $d$ denotes the size of RoBERTa-large embedding.

After the representation of each context and question is extracted, the context embedding needs to be intensified by the question embedding. For this purpose, we apply the attention mechanism to learn the relationship between them. To show the generality of our single-hop QA model's effectiveness, we conduct experiments with two kinds of attention mechanisms: co-attention (Subsubsection 3.2.1) and self-attention (Subsubsection 3.2.2). As a result, context can be updated by either of them:

$$\mathbf{C}' = \mathrm{attention}(\mathbf{E}_q, \mathbf{E}_c) \in \mathbb{R}^{l_c \times h} \tag{2}$$

where $h$ denotes the hidden dimension. The detail is explained in the subsequent sections.

### 3.2.1 Co-attention

Co-attention (Xiong et al., 2016) is a vital model for single-hop QA. It enables the question and context to attend mutually, and also learns the question-aware context representation iteratively. We implement it

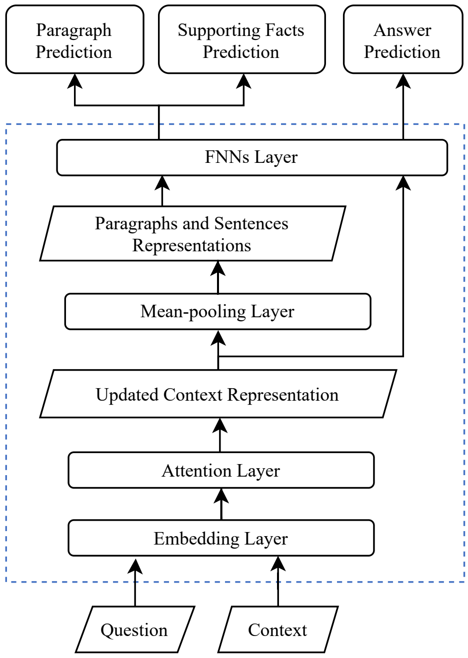


Figure 4: Architecture of proposed attention-based single-hop QA model.

as follows: Embedding $\mathbf{E}_c$ and $\mathbf{E}_q$ is mapped into a hidden dimension by two-layer feed-forward networks (FFNs[2]). Affinity matrix $\boldsymbol{A}$ is the product of context representation $\mathbf{C}$ and question representation $\mathbf{Q}$. In matrix $\boldsymbol{A}$, each value is the related score of one word from the question and one word from the context:

$$\mathbf{C} = \mathrm{FFN}_c(\mathbf{E}_c) \in \mathbb{R}^{l_c \times h} \tag{3}$$

$$\mathbf{Q} = \mathrm{FFN}_q(\mathbf{E}_q) \in \mathbb{R}^{l_q \times h} \tag{4}$$

$$\boldsymbol{A} = \mathbf{C}\mathbf{Q}^\top \in \mathbb{R}^{l_c \times l_q} \tag{5}$$

We normalize matrix $\boldsymbol{A}$ row-wise by softmax, so that each row indicates how much one word from the context is attended by all words from the question. By multiplying it with context representation $\mathbf{C}$, we can obtain the question representation $\mathbf{S}_q$ attended by the context. Similarly, we derive the context representation $\mathbf{S}_c$ attended by the question as follows:

$$\mathbf{S}_q = \mathrm{softmax}(\boldsymbol{A}^\top) \times \mathbf{C} \in \mathbb{R}^{l_q \times h} \tag{6}$$

$$\mathbf{S}_c = \mathrm{softmax}(\boldsymbol{A}) \times \mathbf{Q} \in \mathbb{R}^{l_c \times h} \tag{7}$$

where $\mathrm{softmax}(\cdot)$ denotes the normalization column-wise and $\top$ denotes the matrix transpose.

Let the updated question $\mathbf{S}_q$ attend context again with the matrix $\boldsymbol{A}$. In addition, the attended context is

[2] All FFNs in this work includes two linear transformations with ReLU, Layer Normalization and Dropout in between.

further fed into a BiGRU as follows:

$$\mathbf{D}_c = \text{BiGRU}(\text{softmax}(\boldsymbol{A}) \times \mathbf{S}_q) \in \mathbb{R}^{l_c \times h} \quad (8)$$

$\mathbf{D}_c$ and $\mathbf{S}_c$ are context representations intensified by the question. Finally, they are concatenated and further applied with the $\text{FFN}_d$ to transform into the original document's length:

$$\mathbf{C}' = \text{FFN}_d([\mathbf{D}_c; \mathbf{S}_c]) \in \mathbb{R}^{l_c \times h} \quad (9)$$

where $[\cdot;\cdot]$ denotes the concatenation function.

#### 3.2.2 Self-attention

We use a Transformer encoder (Vaswani et al., 2017) for defining self-attention, including a linear layer that maps the representation into the hidden dimension. It can capture relations between each pair of words from the query and the context. We set 8-head attention and stack two encoder layers to keep the model's size smaller than HGN.

$$\mathbf{C}' = \text{TransformerEncoder}([\mathbf{E}_q; \mathbf{E}_c]) \in \mathbb{R}^{l_c \times h} \quad (10)$$

#### 3.2.3 Prediction

After the attention module, updated context $\mathbf{C}'$ is sent to a mean-pooling layer to extract the representations of paragraphs and sentences:

$$\mathbf{P} = \text{Mean-pooling}(\mathbf{C}', start_p, end_p) \quad (11)$$

$$\mathbf{S} = \text{Mean-pooling}(\mathbf{C}', start_s, end_s) \quad (12)$$

where $start_p$ and $start_s$ denote the starting positions of each paragraph and each sentence, respectively. Similarly, $end_p$ and $end_s$ denote the ending positions.

Unlike the conventional single-hop QA, additional layers are employed for sub-tasks. the paragraphs' representation $\mathbf{P}$ is sent to a FFN for binary classification to calculate the probability that they contain supporting facts or not. Similarly, the sentences' representation $\mathbf{S}$ is sent to a FNN to calculate the probability that they are supporting facts or not.

$$\boldsymbol{o}_{para} = \text{FFN}_1(\mathbf{P}) \quad (13)$$

$$\boldsymbol{o}_{sent} = \text{FFN}_2(\mathbf{S}) \quad (14)$$

On the other hand, updated context $\mathbf{C}'$ is directly sent to other FFNs to predict the starting and ending positions of the answer span:

$$\boldsymbol{o}_{start} = \text{FFN}_4(\mathbf{C}') \quad (15)$$

$$\boldsymbol{o}_{end} = \text{FFN}_5(\mathbf{C}') \quad (16)$$

Since the answer type could be "yes", "no" or an answer span, 3-way classification is conducted. If the prediction is "yes" or "no", the answer is directly returned. Otherwise, the answer span is returned. Similar with HGN, we also use the first hidden representation for answer type classification.

$$\boldsymbol{o}_{type} = \text{FFN}_6(\mathbf{C}'[0]) \quad (17)$$

### 3.3 Multi-task Learning

Finally, an answer type, an answer span with the starting and ending positions, gold paragraphs, and support facts are jointly predicted for multi-task learning. The cross-entropy loss is used for each task. Thus, the total loss ($\mathcal{L}_{total}$) is a weighted sum of each loss and each weight $\lambda_i$ is our hyper-parameter:

$$\begin{aligned}\mathcal{L}_{total} = &\lambda_1 \mathcal{L}_{type} + \lambda_2 \mathcal{L}_{start} + \lambda_3 \mathcal{L}_{end} \\ &+ \lambda_4 \mathcal{L}_{para} + \lambda_5 \mathcal{L}_{sent}\end{aligned} \quad (18)$$

## 4 TWO-STEP TUNING

BERT-based language models (Devlin et al., 2019; Liu et al., 2019) are pre-trained on the large-scale corpora to learn universal semantics. But for a specific task, such as multi-hop QA, the data distribution can be different. More tuning on a related domain is expected to bring improvement as also investigated in (Houlsby et al., 2019; Sun et al., 2019). Therefore, we propose two-step tuning with an in-task distribution and a cross-task distribution for enhancing the model's performance. To study its effectiveness based on diverse datasets, we experiment with five datasets: SQuAD (Rajpurkar et al., 2018b), NewsQA (Trischler et al., 2017), TweetQA (Xiong et al., 2019), CoLA (Warstadt et al., 2019), IMDB (Maas et al., 2011).

**In-task Tuning:** In this scenario, language model is first tuned in a QA dataset[3], including SQuAD, NewsQA and TweetQA[4]. Then, we use the tuned language model as an embedding in our proposed AMS and perform the second tuning in HotpotQA.

**Cross-task Tuning:** In this scenario, the first tuning dataset is not a QA dataset. Specifically, CoLA is a grammatical classification dataset and IMDB is a sentimental classification dataset. The second tuning process is the same as the in-task tunning.

## 5 EXPERIMENT

### 5.1 Dataset

HotpotQA (Yang et al., 2018) is a popular multi-hop QA dataset, which is constructed from Wikipedia.

[3]We only tune the the language model, instead of the entire model, in first tuning. It enables us to study its effectiveness from cross-task datasets.

[4]There is no annotated answer span in TweetQA. We retrieve the span with the best BLUE-1 score for training.

Table 2: Comparison between HGN and AMS on dev set. The upper part is based on original RoBERTa-large embedding, which means the RoBERTa-large embedding from HuggingFace without two-step tuning. The lower part is based on SQuAD tuning embedding, which means two-step tuning based on SQuAD. 'Ans' indicates 'Answer' and 'Sup' indicates 'Supporting facts'. $\Delta$ = model's performance - HGN (reproduced) performance with original RoBERTa-large.

| Embedding | Model | Ans | | Sup | | Joint | |
|---|---|---|---|---|---|---|---|
| | | EM | F1 | EM | F1 | EM | F1 |
| Original RoBERTa-large | HGN (reproduced) | 68.33 | 82.04 | 62.89 | 88.53 | 45.78 | 74.06 |
| | $AMS_{\text{co-attention}}$ | 67.85 | 81.55 | 63.28 | 87.7 | 46.35 | 73.58 |
| | $\Delta$ | -0.48 | -0.49 | 0.39 | -0.83 | 0.57 | -0.48 |
| | $AMS_{\text{self-attention}}$ | 68.87 | 82.14 | 63.20 | 88.45 | 46.67 | 74.21 |
| | $\Delta$ | 0.54 | 0.10 | 0.31 | -0.08 | 0.89 | 0.15 |
| SQuAD tuning | HGN (reproduced) | 69.14 | 82.55 | 63.24 | 88.82 | 46.75 | 74.75 |
| | $\Delta$ | 0.81 | 0.51 | 0.35 | 0.29 | 0.97 | 0.69 |
| | $AMS_{\text{co-attention}}$ | 69.21 | 82.48 | 63.7 | 88.62 | 47.33 | 74.41 |
| | $\Delta$ | 0.88 | 0.44 | 0.81 | 0.09 | 1.55 | 0.35 |
| | $AMS_{\text{self-attention}}$ | 69.26 | 82.51 | 64.4 | 88.63 | 47.56 | 74.62 |
| | $\Delta$ | 0.93 | 0.47 | 1.51 | 0.1 | 1.78 | 0.56 |

Table 3: Comparison between different embeddings.

| Embedding | Ans | | Sup | | Joint | |
|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 |
| Original RoBERTa-large | 67.85 | 81.55 | 63.28 | 87.7 | 46.35 | 73.58 |
| SQuAD tuning | **69.21** | **82.48** | **63.7** | 88.62 | **47.33** | **74.41** |
| TweetQA tuning | 67.87 | 81.79 | 63.52 | 88.62 | 46.84 | 73.93 |
| NewsQA tuning | 68.28 | 82.09 | 63.65 | **88.77** | 47.24 | 74.21 |
| CoLA tuning | 67.86 | 81.44 | 63.59 | 87.29 | 46.84 | 73.29 |
| IMDB tuning | 67.56 | 81.43 | 63.66 | 87.31 | 46.65 | 73.15 |



There are two sub-datasets: the distractor setting and the fullwiki setting. For each case in the distractor setting, a compositional question and a document containing 10 paragraphs are given. In the document, only 2 paragraphs are related with the question and other 8 paragraphs are distractions. The gold paragraphs, supporting facts and ground-truth answers are annotated. The QA system is required to predict both an answer and supporting facts. In the fullwiki setting, the answer should be retrieved from the whole Wikipedia. In this work, we focus on the distractor setting. Official evaluation metrics are considered, i.e., EM (exact match) and the F1 score for the individual and joint evaluations of both the answer and supporting facts.

## 5.2 Experimental Setting

We conduct experiments based on a Quadro RTX 8000 GPU. We train the model for 8 epochs, and set learning rate as 1e-5 with batch size 8. For the hyperparameters in our multi-task learning, we search $\lambda_1$, $\lambda_2$, $\lambda_3$ and $\lambda_4$ from $\{\mathbf{1},3,5\}$ and $\lambda_5$ from $\{5, 10, 15, \mathbf{20}\}$, in which the boldface indicates the best setting.

## 5.3 Experimental Result

### 5.3.1 Comparison with Baseline

We reproduce HGN with its source code and the result is based on RoBERTa-large. The upper part of Table 2 shows the comparison between our proposed AMS and HGN on the development set. According the table, the co-attention based model ($AMS_{\text{co-attention}}$) underperforms HGN within 1.0 point. The self-attention based model ($AMS_{\text{self-attention}}$) yields the better performance and especially outperforms HGN by 0.89 points for Joint EM.

### 5.3.2 Comparison based on Two-step Tuning

Table 3 shows the comparison between the original RoBERTa-large embedding and our two-step tuning embedding. This result is based on $AMS_{\text{co-attention}}$, demonstrating the following information:

- In-task tuning can improve overall performance.
- SQuAD tuning yields the best improvement and TweetQA yields the smallest improvement. Potential reasons could be: (i) SQuAD and HotpotQA are all constructed from Wikipedia; thus,

Table 4: Comparison with related work on dev set. AMS result is based on SQuAD tuning and HGN result is without SQuAD tuning.

| Embedding | Model | Ans | | Sup | | Joint | |
|---|---|---|---|---|---|---|---|
| | | EM | F1 | EM | F1 | EM | F1 |
| Bert-base | DFGN (Xiao et al., 2019) | 55.66 | 69.34 | 53.10 | 82.24 | 33.68 | 59.86 |
| | HGN (Fang et al., 2020) | 60.23 | 74.49 | 56.62 | 85.91 | 38.16 | 66.20 |
| | $\text{AMS}_{\text{co-attention}}$ | 61.39 | 75.39 | 58.78 | **85.93** | 40.04 | 67.03 |
| | $\text{AMS}_{\text{self-attention}}$ | **62.11** | **75.76** | **59.20** | 85.78 | **40.73** | **67.39** |
| RoBERTa-large | SAE (Tu et al., 2020) | 67.70 | 80.75 | 63.30 | 87.38 | 46.81 | 72.75 |
| | HGN (Fang et al., 2020) | 68.33 | 82.04 | 62.89 | 88.53 | 45.78 | 74.06 |
| | $\text{AMS}_{\text{co-attention}}$ | 69.21 | 82.48 | 63.70 | 88.22 | 47.33 | 74.41 |
| | $\text{AMS}_{\text{self-attention}}$ | **69.26** | **82.51** | **64.40** | **88.63** | **47.56** | **74.62** |

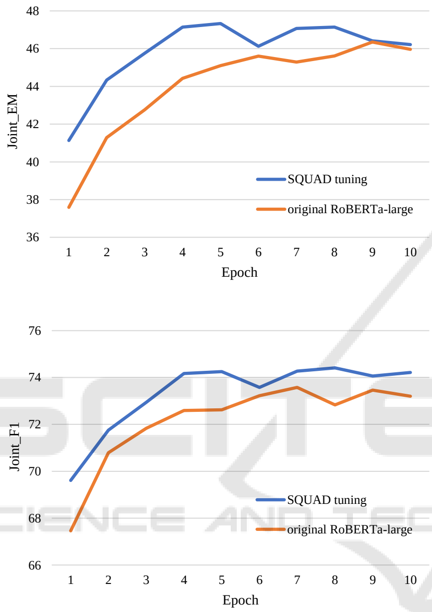


Figure 5: Comparison between original RoBERTa-large and SQuAD tuning on Joint EM (upper) and Joint F1 (lower).

they may share the same resource and most relevant data distribution. (ii) TweetQA is more oral-style than other datasets. And the retrieved answer for training in TweetQA could be incomplete.

- Cross-task tuning can improve Sup EM but cannot benefit the answer prediction. We hypothesize that this is because supporting facts prediction is closely aligned with the classification task.

The lower part of Table 2 illustrates that both HGN and AMS can be overall enhanced by SQuAD tuning (two-step tuning based on SQuAD). Compared with the reproduced HGN, AMS with SQuAD tuning can outperform it obviously in Sup EM and Joint EM. Furthermore, under the condition of both AMS and HGN using SQuAD tuning, their performances are quite competitive.

Table 5: Comparison of model's size, computational resource and performance.

| | Baseline | Proposed model | |
|---|---|---|---|
| | HGN | $\text{AMS}_{\text{co-attention}}$ | $\text{AMS}_{\text{self-attention}}$ |
| Model's size | 31.61M | **6.30M** | 30.83M |
| RoBERTa-large | 355M | 355M | 355M |
| Training time | 191 min | **148 min** | 160 min |
| Joint EM | 45.78 | 47.33 | **47.56** |
| Joint F1 | 74.06 | 74.41 | **74.62** |

Figure 5 shows curve comparisons between the original RoBERTa-large and the SQuAD tuning based on Joint F1 (bottom) and Joint EM (top). From the figure, the SQuAD tuning curve is initially better than the original RoBERTa-large curve and it converges around $4^{\text{th}}$ epoch. This is faster than the original RoBERTa-large, showing the power of transfer learning in multi-hop reasoning.



### 5.3.3 Comparison with Related Work

We make comparisons with GNN-based models that use the BERT-based language model and the document filter. Table 4 shows the comparison result on the development set. According to the table, our proposed method outperforms GNN-based models with both BERT-base and RoBERTa-large, and $\text{AMS}_{\text{self-attention}}$ yields the best performance.

## 5.4 Comparison of Model's Size and Computational Resource

Table 5 shows the comparison of the model's size, computational resource and performance. The result is based on RoBERTa-large. $\text{AMS}_{\text{co-attention}}$ model's size is only about 20% of HGN and $\text{AMS}_{\text{self-attention}}$ model's size is close to HGN. For computational resource, $\text{AMS}_{\text{co-attention}}$ and $\text{AMS}_{\text{self-attention}}$ is 77.5% and 83.8% of HGN, respectively. Since RoBERTa-large (355M) dominates the total model's size, training time is not reduced significantly. The computational resource is expected to further decrease by incorporating a lighter language model. Generally, both

Table 6: Some examples that supporting facts F1 is 0 but answer F1 is 1.

| ID | Answer | Supporting Facts | Predicted Answer | Predicted Supporting Facts |
|---|---|---|---|---|
| 5ae180195542 9901ffe4aec4 | Battle Creek, Michigan | [['Adventures of Superman (TV series)', 2], [' Kellogg's', 0], ['Kellogg's', 2]] | Battle Creek, Michigan | [['Cocoa Krispies', 0],['Adventures of Superman (TV series)', 0]] |
| 5ae1fa2b5542 997f29b3c1df | Eminem | [['Mack 10 discography', 2], ['Numb (Rihanna song)', 0]] | Eminem | [['The Monster (song)', 0], ['Numb (Rihanna song)', 1]] |
| 5ae18d615542 997283cd2229 | mixed martial arts | [['Liz McCarthy (fighter)', 0], ['Atomweight (MMA)', 0]] | mixed martial arts | [['Atomweight', 0], ['Amber Brown (fighter)', 0]] |

proposed models show better performance and use less computational resource.

### 5.5 Error Analysis

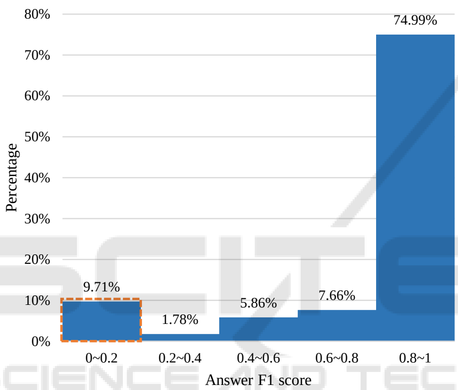


Figure 6: Answer F1 score distribution on dev set. There are almost 10% answer F1 score less than 0.2.

We analyse the answer F1 score on the development set. Figure 6 illustrates its distribution. Almost 10% of the answer F1 score is less than 0.2, in which 9.7% answer F1 score is 0. Further improvement can be considered from this error. Similar with HGN, we randomly sample 100 examples with answer F1 score as 0 and they are categorized as follows:

- **Multi-answer** (12%)**:** There are multiple gold answers and the predicted answer is different from the annotation. For example, the annotation is 'National Broadcasting Company' and the predicted answer is 'NBC'.
- **Multi-hop** (28%)**:** The supporting facts prediction is incorrect, from which the model fails to predict the right answer. For example, the supporting facts are the 1$^{st}$ and the 2$^{nd}$ sentences, but the model predicts the 3$^{rd}$ and the 4$^{th}$ sentences as supporting facts and retrieves answer from them. Accordingly, the answer prediction is incorrect.
- **MRC** (38%)**:** The supporting facts' prediction is right but the answer prediction is wrong. For example, the supporting facts are the 1$^{st}$ and the 2$^{nd}$ sentences. The model predicts them correctly. But the final answer prediction is wrong.
- **Comparison** (22%)**:** The model fails to do numerical operations that involves information aggregation. For example, the question is ' The CEO of Walmart and the CEO of Apple, who is older?'

Multi-hop and MRC account for more than 50%, which indicates that the performance could be further improved by more advanced QA models.

Another tricky error is that there are 1,322 cases, about 17% of the development set, that supporting fact F1 is 0 but answer F1 is 1. This means that the supporting facts prediction is wrong but the answer prediction is right. Table 6 shows some examples of this case. Such interpretable problem may occur when the answer is not directly retrieved from predicted sentences. It could be further studied by considering supporting facts prediction's restrictions for the answer prediction.

## 6 CONCLUSIONS

In this research, we propose a new model, called AMS, for multi-hop QA. AMS is the integration of HGN's document filter and single-hop QA models. We also introduce a new fine-tuning scheme for improving its performance. The result shows that AMS can outperform the strong baseline HGN with less amount of computational resource. Furthermore, AMS can achieve the better performance than other sophisticated GNN-based models. In contrast to GNN-based methods, our method can effectively leverage existing single-hop QA models and does not require any auxiliary tool, such as NER, which should gain more attention in the further research.

According to our analysis, there is still potential for further improvement and interpretable issues to be addressed. In addition, since our method strongly depends on the document filter, the development of filter for other datasets is necessary for its universal application. We leave these studies as our future work.

## ACKNOWLEDGEMENTS

This work was supported by JSPS Kakenhi Grant Number 20H04295, 20K20406, and 20K20625.

## REFERENCES

De Cao, N., Aziz, W., and Titov, I. (2019). Question answering by reasoning across documents with graph convolutional networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2306–2317.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Dhingra, B., Jin, Q., Yang, Z., Cohen, W. W., and Salakhutdinov, R. (2018). Neural models for reasoning over multiple mentions using coreference. *arXiv preprint arXiv:1804.05922*.

Ding, M., Zhou, C., Chen, Q., Yang, H., and Tang, J. (2019). Cognitive graph for multi-hop reading comprehension at scale. *arXiv preprint arXiv:1905.05460*.

Fang, Y., Sun, S., Gan, Z., Pillai, R., Wang, S., and Liu, J. (2020). Hierarchical graph network for multi-hop question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8823–8838.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28:1693–1701.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. (2019). Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Huang, Y. and Yang, M. (2021). Breadth first reasoning graph for multi-hop question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5810–5821.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. (2017). Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Khashabi, D., Chaturvedi, S., Roth, M., Upadhyay, S., and Roth, D. (2018). Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Nishida, K., Nishida, K., Nagata, M., Otsuka, A., Saito, I., Asano, H., and Tomita, J. (2019). Answering while summarizing: Multi-task learning for multi-hop qa with evidence extraction. *arXiv preprint arXiv:1905.08511*.

Peters, M. E., Ruder, S., and Smith, N. A. (2019). To tune or not to tune? adapting pretrained representations to diverse tasks. *arXiv preprint arXiv:1903.05987*.

Qiu, L., Xiao, Y., Qu, Y., Zhou, H., Li, L., Zhang, W., and Yu, Y. (2019). Dynamically fused graph network for multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6140–6150.

Rajpurkar, P., Jia, R., and Liang, P. (2018a). Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Rajpurkar, P., Jia, R., and Liang, P. (2018b). Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Shao, N., Cui, Y., Liu, T., Wang, S., and Hu, G. (2020). Is graph structure necessary for multi-hop question answering? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7187–7192.

Song, L., Wang, Z., Yu, M., Zhang, Y., Florian, R., and Gildea, D. (2018). Exploring graph-structured passage representation for multi-hop reading comprehension with graph neural networks. *arXiv preprint arXiv:1809.02040*.

Sun, C., Qiu, X., Xu, Y., and Huang, X. (2019). How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Trischler, A., Wang, T., Yuan, X., Harris, J., Sordoni, A., Bachman, P., and Suleman, K. (2017). Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200.

Tu, M., Huang, K., Wang, G., Huang, J., He, X., and Zhou, B. (2020). Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9073–9080.

Tu, M., Wang, G., Huang, J., Tang, Y., He, X., and Zhou, B. (2019). Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2704–2713.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Warstadt, A., Singh, A., and Bowman, S. R. (2019). Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

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

Wu, L., Chen, Y., Shen, K., Guo, X., Gao, H., Li, S., Pei, J., and Long, B. (2021). Graph neural networks for natural language processing: A survey. *arXiv preprint arXiv:2106.06090*.

Xiao, Y., Qu, Y., Qiu, L., Zhou, H., Li, L., Zhang, W., and Yu, Y. (2019). Dynamically fused graph network for multi-hop reasoning. *arXiv preprint arXiv:1905.06933*.

Xiong, C., Zhong, V., and Socher, R. (2016). Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*.

Xiong, W., Wu, J., Wang, H., Kulkarni, V., Yu, M., Chang, S., Guo, X., and Wang, W. Y. (2019). Tweetqa: A social media focused question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5020–5031.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., and Manning, C. D. (2018). HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Zhang, Z., Yang, J., and Zhao, H. (2021). Retrospective reader for machine reading comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14506–14514.

Zhong, V., Xiong, C., Keskar, N. S., and Socher, R. (2019). Coarse-grain fine-grain coattention network for multi-evidence question answering. *arXiv preprint arXiv:1901.00603*.