Ph.D. DISSERTATION

Sentence Matching using Deep Learning for Question Answering

질의 응답을 위한 딥러닝 기반 문장 매칭

BY

SEONHOON KIM

AUGUST 2021

Intelligent Systems
Department of Transdisciplinary Studies
Graduate School of Convergence Science and Technology
SEOUL NATIONAL UNIVERSITY

Sentence Matching using Deep Learning for Question Answering

질의 응답을 위한 딥러닝 기반 문장 매칭

지도교수 곽 노 준
이 논문을 공학박사 학위논문으로 제출함

2021년 8월

서울대학교 대학원

융합과학부 지능형융합시스템전공

김 선 훈

김선훈의 공학박사 학위 논문을 인준함

2021년 8월

위 원 장:	서 봉 원
부위원장:	곽 노 준
위 원:	이 원 종
위 원:	최 상 일
위 원:	최 종 현

Abstract

Question Answering is becoming one of the most important applications in natural language processing, thanks to the development of deep neural networks. Improving the performance of answering the questions helps humans acquire more useful information efficiently. In this dissertation, we study sentence matching that understands the relationship between the sentences for better reasoning in various question answering systems.

First, we propose a semantic sentence matching model for question paraphrase identification, natural language inference, and answer sentence selection which can be used in the question answering system. We propose a densely-connected co-attentive recurrent neural network, each layer of which uses concatenated information of attentive features as well as hidden features of all the preceding recurrent layers. It enables preserving the original and the co-attentive feature information from the bottommost word embedding layer to the uppermost recurrent layer. To alleviate the problem of the ever-increasing size of feature vectors due to dense concatenation operations, we also propose to use an autoencoder after dense concatenation.

Second, we propose matching strategies to find the relevant part against the question and the answer option from the textual context. For the word-level matching required in the task which has a number of technical terminologies, We build a dependency tree with Dependency Parser for each sentence of the textual context and designate the words which exist in the question and the answer option as anchor nodes. We can narrow down the scope to answer more precisely by removing the nodes which are far from the anchor nodes. In addi-

tion, we utilize an additional temporal localization classifier as an auxiliary task to find the relevant subtitle sentence from the long subtitle context by calculating the relevance matching score of the subtitle sentences.

Lastly, we propose the training schemes for multiple-choice video question answering in order to enhance the performance with a self-supervised pre-training stage and supervised contrastive learning in the main stage as auxiliary learning. For the pre-training stage, we transform the original problem format to have a better parameter initialization from predicting the correct answer into predicting the corresponding question of the context by building the synthesized pre-training dataset. In the main stage, we propose the supervised contrastive representation learning method as another auxiliary learning to separated the embedding space between the correct answer and the wrong answers to enhance the model performance. Taking the ground truth answer as a positive sample and the rest as negative samples, the contrastive loss confines the positive sample to be mapped in the neighborhood of an anchor, a perturbed ground truth answer, and the negative samples to be away from the anchor. Our model achieves the best performance on the challenging multiple-choice Video QA tasks, TVQA, TVQA+, and DramaQA.

keywords: Question Answering, Deep neural network, Text matching, Sentence matching, Self-supervised learning, Contrastive learning

student number: 2017-30004

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Sentence Matching in Question Answering

Question Answering system, which is the intelligent machine answering the questions from the information-seeking human, has become one of the most important applications in the field of Natural Language Processing tasks. The development of this system is able to not only make humans find more useful information efficiently but also enhance the accessibility to a variety of smart devices and information sources such as a single document, a book, a bunch of documents like web, and even images and videos. And, it has shown great potential to be applied to real-world problems.

With the recent advancements of deep neural networks, question answering has gotten much attention. To keep developing the question answering intelligence, a variety of benchmark tasks and datasets have been created such as answer sentence selection [94], question identification of community-based question answering [17], context-based question answering as a machine reading comprehension [75], open-domain question answering [5], knowledge base

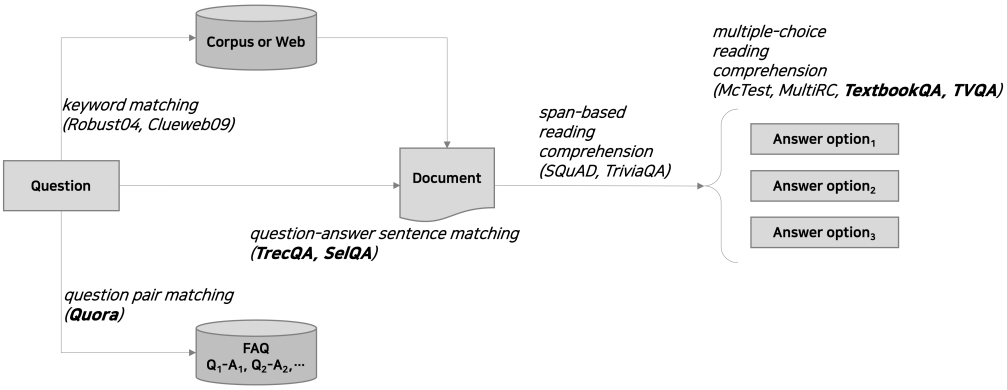


Figure 1.1: One example of the question answering system. Text matching, one of the most important essential components in the question answering system, is required in various parts of the system such as keyword matching for retrieving relevant documents, question pair matching for retrieving questions from FAQs, and answer sentence matching for selecting the answerable sentences. Furthermore, we also require a text matching strategy for context-based question answering such as span-based or multiple-choice reading comprehension tasks.

question answering [18], and image- or video-based visual question answering [2, 55].

Most question answering tasks are relying on text matching as shown in Fig. 1.1. We can discern not only the answerable sentences in answer sentence selection through the sentence matching model between the question and the answer sentences but also semantically similar questions in paraphrase identification for question retrieval tasks between the question pairs as shown in Table 1.1, respectively.

For the context-based question answering such as span-based reading comprehension task as shown in Table 1.2, we can narrow down the context scope to

answer more precisely by matching the text such as words or sentences, between the question/answer and the context. When it comes to the multiple-choice question answering such as the example in Table 1.3, there are multiple answer options, and we can select the correct answer comparing each answer option with the question/context.

Earlier approaches of sentence matching mainly relied on conventional methods such as syntactic features, transformations or relation extraction [79, 94]. Wang *et al*. [94] proposed a statistical syntax-based model that softly aligns a question sentence with a candidate answer sentence with a quasi-synchronous dependency grammar, following the story that questions can be generated from the answers through a series of syntactic and semantic transformations. Wan *et al*. [92] proposed an approach to filter out inconsistent sentences of false paraphases with syntatical features such as differences in sentence length, word overlap based metrics such as BLEU, and dependency tree overlap.

Later on, the sentence pairs are represented by a form of vectorized representation using deep neural networks such as RNN or CNN [87, 97, 16, 88, 27, 26]. And, also neural network-based models calculate the similarity score as a matching degree between two sentences, and the attention mechanism can improve the matching performance with an interaction between two sentences referencing the word alignments. Although these neural models show a good performance on sentence matching tasks, they do not use all features enough and only use the most abstract level of features.

In this thesis, we propose the new RNN framework for matching problems utilizing all layers' output representations with cross attentive features as an aggregation of the information. The experimental results show that our model can achieve competitive performance on multiple sentence matching tasks such

as paraphrase identification, answer sentence selection, and natural language inference, and our proposed model also can be utilized for various tasks using RNN architecture.

There are many benchmark datasets imitating the real-world problems to evaluate the Question Answering system, and most of these datasets require a certain context to reason the answer correctly as shown in Tables 1.2 and 1.3. The context is composed of a number of sentences, and to properly focus on the relevant part to answer the question, we need to use the sentence-level matching between the question and each sentence of the context. Furthermore, if we deal with the multiple-choice question answering as shown in Table 1.3, the answer options are also required in sentence matching with each sentence of the context.

We proposed the strategies for context matching according to the types of domain. Textbook Question Answering, the practical middle school science problems across multiple modalities, is composed of a number of technical terminologies, so we explicitly emphasized the keyword matching for this task even though we used the semantical representation with deep neural networks. We concentrated the matched terminology between the question/answer and the context by utilizing both syntactical and semantical information and with this process, we could narrow down the context scope to answer the question more precisely. For the Video Question Answering such as TVQA and DramaQA, TV show-based datasets containing the subtitle and the video content as a context, we used the subtitle sentence matching with a question and each answer option simultaneously to use fruitful information as an early stage fusion. We encoded the question-answer-subtitle sentences in the early stage while the previous methods [56, 46] used the late stage fusion that they encoded the question-answer and subtitle respectively and fused them in the later stage. Also, we used

the temporal localization loss as an auxiliary learning approach, focusing on the relevant subtitle sentences for answering the question, and we achieved state-of-the-art performance on these tasks.

1.2 Motivation

In this section, we discuss the importance of semantic sentence matching, which is a fundamental technology in natural language processing.

Language is primarily made up of a symbolic system. Humans can communicate using these symbols conveying semantics verbally or in writings. Since there can be various textual appearances expressing a certain semantics, it is significantly important to understand the semantics of the languages. In natural language processing, previous works concentrated on the symbolic and lexical features using word overlap or co-occurrence frequency. In recent, vector-based representation is used to express the semantics, and especially the recurrent network is widely used for learning the sequential information of the language. However, a very deep recurrent network is unstable to learn semantics due to the well-known vanishing gradient problem. We, in this dissertation, propose the deep recurrent network architecture that enables to learn model utilizing every granular semantics in every layer as collective knowledge.

Semantic sentence matching can contribute to the development of natural language processing tasks. In particular, various question answering tasks require matching techniques. For example, humans behave a certain process of finding the correct answer through text matching when taking an exam. To solve the problem, humans try to find the relevant part of the given context comparing each part such as words or sentences with the question. In this case, a matching

technique is required to compare them. And, after finding the relevant part of the context, it is also required to compare each answer option with the question and the relevant part of the context. This dissertation also focuses on proposing text matching methods to be helpful for solving the practical question answering tasks that can be applied to the real-world scenario.

The pre-training method recently has been widely used in the research community of the language domain. This method needs a large-scale unlabeled dataset to pre-train the model using language model objectives. And, this pre-trained language model is utilized for the various natural language understanding tasks. Even though this brings a huge performance improvement, there is room to improve the performance using the given downstream dataset itself as another pre-training research direction. We propose the pre-training strategy to learn better weight initialization using the downstream dataset before fine-tuning the model to that dataset. We also propose a contrastive learning method that allows the matching score for the correct answer to be higher than for the wrong answers to improve the QA system.

Modeling accurate sentence matching is crucial in understanding the language, and our proposed matching strategies can be helpful for the development of the QA applications that make humans being more convenient.

1.3 Outline

The remainder of this dissertation is composed as follows. In chapter 2, we briefly review the background research of the text matching for question answering.

1.3.1 Sentence Pair Matching

In chapter 3, we address sentence pair matching tasks such as paraphrase identification, answer sentence selection, and natural language inference which are one of the sub-tasks for question answering. We require to identify the relationship between two sentences with a sentence pair matching model. For paraphrase identification, we identify whether the two sentences are paraphrase or not. There are two sentences as a question and an answer sentence in the answer sentence selection task, and we identify that the candidate answer sentence can be an answer for the question sentence. For the natural language inference, we need to recognize the relationship between a premise and a hypothesis whether that relationship is entailment, neutral, or contradiction. First, we follow the siamese structure for the sentence pair modeling. Inspired by DenseNet, a densely connected convolutional network, we propose a densely-connected co-attentive recurrent neural network, each layer of which uses concatenated information of attentive features as well as hidden features of all the preceding recurrent layers. It enables preserving the original and the co-attentive feature information from the bottommost word embedding layer to the uppermost recurrent layer without any deformation. These intact features over multiple layers compose a community of semantic knowledge and outperform the previous deep RNN models using residual connections. To alleviate the problem of an ever-increasing size of feature vectors due to dense concatenation operations, we also propose to use an autoencoder after dense concatenation with the property of controllable feature sizes. We evaluate our proposed architecture on highly competitive benchmark datasets related to sentence matching, and the experimental results show that our architecture, which retains recurrent and attentive

features, achieved state-of-the-art performances for most of the tasks at the time our paper was published.

1.3.2 Context based Question Answering

In chapter 4, we try to solve the challenging context-based multiple-choice question answering tasks such as TextbookQA, TVQA, TVQA+, and DramaQA. To properly answer the question, we need to identify the relevant sentences from the context. For TextbookQA which contains a number of technical terminologies, exact term matching is important since those terminologies are hardly paraphrased as different expressions. We first segment the context with multiple sentences and extract the dependency trees from each sentence using a dependency parser. We designate the exact keywords of the context as anchor nodes if those keywords exist in the question or the answer option. Utilizing the relations of the dependency tree, we remove the nodes of the context which are far from the anchor node to narrow down the scope. With building the context graph with an aggregation of the trees from each sentence, we use the graph convolutional neural network to reason the correct answer. We can see that this proposed approach not only reduces the training and inference time but also enhances the performance focusing on a more relevant part. For TVQA, TVQA+, and DramaQA as Video QA, we also split the subtitle into multiple subtitle sentences and align each sentence with the given video frames. And, we make use of timestamp annotation of localized span needed to answer the question provided in the dataset and add temporal localization learning as an auxiliary task. Furthermore, we use subtitle, question, and answer option sentences simultaneously as interaction information in an early stage fusion while the previous studies encode the question and the answer option at first and encode the subti-

tle separately. And our model achieved state-of-the-art performance on TVQA, TVQA+, and DramaQA datasets.

1.3.3 Training Schemes for the Multiple-choice Question Answering

In chapter 5, we propose the training strategies as self-supervised pre-training and contrastive representation learning approach to improve the performance, especially for the multiple-choice question answering. Recently, there are a variety of pre-training strategies requiring a large-scale corpus such as BERT [20]. On the contrary to this, we propose another self-supervised pre-training stage using only the task-specific dataset such as Textbook QA or TVQA. We transform the original problem format of predicting the correct answer into the one that predicts the relevant context or question to provide a model with broader contextual inputs without any further dataset or annotation. For example, the context, question, and answer options are provided. We retrieve the questions/contexts using TF-IDF against the context/question in which those pairs are actually irrelevant. And we designate the original pair from the dataset as the positive sample and the retrieved and irrelevant pairs as the negative samples. Finally, we pre-train the model with these synthesized data to learn a better weight initialization. In the main QA training stage, we also propose contrastive representation learning to separate the correct answer and the negative answers farther. We first add a masking noise to the input corresponding to the ground-truth answer, and consider the original input of the ground-truth answer as a positive sample, while treating the rest as negative samples. For the contrast the positive and negative samples, we calculate the matching scores using the dot product as a matching function. By mapping the positive sample closer to the

masked input, we can see that the model performance is improved.

In chapter 6, we will finally conclude the dissertation.

Answer Sentence Selection	
Question	*How fast does the Concorde fly?*
Answer 1	*The Concorde, which crosses the Atlantic at 1,350 mph, has been considered among the world's safest planes.*
Answer 2	*Except for a handful of astronauts and military pilots, we Concorde passengers are flying higher and faster than any other humans.*
Answer 3	*Some days this year are worse, but the airline pointed to one recent post-crash concorde flight from new york that arrived with 82 passengers, compared to 32 on the same date a year ago.*
Label	***Answer 1***
Paraphrase Identification for Question Retrieval	
Question	*What are the best books on algorithms and data structures?*
Question 1	*What are the best algorithm books for beginners?*
Question 2	*Should there be a book on how to use Kibana?*
Question 3	*Which is the best book for data structure and algorithm using Java?*
Label	***Question 1***

Table 1.1: Examples of answer sentence selection and paraphrase identification tasks.

Machine Reading Comprehension as a Context Question Answering	
Question	*In what country is Normandy located?*
Context	***The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France**. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.*
Answer	***France***

Table 1.2: Example of machine reading comprehension.

Multiple choice Question Answering	
Question	*Why does Sally like walking in the woods?*
Context	*Sally had a very exciting summer vacation. She went to summer camp for the first time. She made friends with a girl named Tina. They shared a bunk bed in their cabin.* ***Sally's favorite activity was walking in the woods because she enjoyed nature****. Tina liked arts and crafts. Together, they made some art using leaves they found in the woods. Even after she fell in the water, Sally still enjoyed canoeing. She was sad when the camp was over, but promised to keep in touch with her new friend. Sally went to the beach with her family in the summer as well. She loves the beach. Sally collected shells and mailed some to her friend, Tina, so she could make some arts and crafts with them. Sally liked fishing with her brothers, cooking on the grill with her dad, and swimming in the ocean with her mother. The summer was fun, but Sally was very excited to go back to school. She missed her friends and teachers. She was excited to tell them about her summer vacation.*
Answer 1	*She likes to climb trees.*
Answer 2	*She likes to play hide and go seek.*
Answer 3	*She likes to swim.*
Answer 4	***She likes nature.***
Label	***Answer 4***

Table 1.3: Example of multiple-choice question answering.

Chapter 2

Background

In this chapter, we address the background of the text matching problem for Question Answering. First, we briefly review the text representation learning approaches which have been the base component for neural networks in the natural language domain. Second, we address the various sentence matching tasks, one of the fundamental tasks of natural language processing. Next, we investigate various question answering tasks, and lastly, we introduce further techniques such as contrastive learning and self-supervised learning approaches to learn the better representation which is related to chapter 5.

2.1 Learning Text Representation

To match the text, most models need to acquire the textual representations in advance using the word embeddings to reflect the semantics of the text.

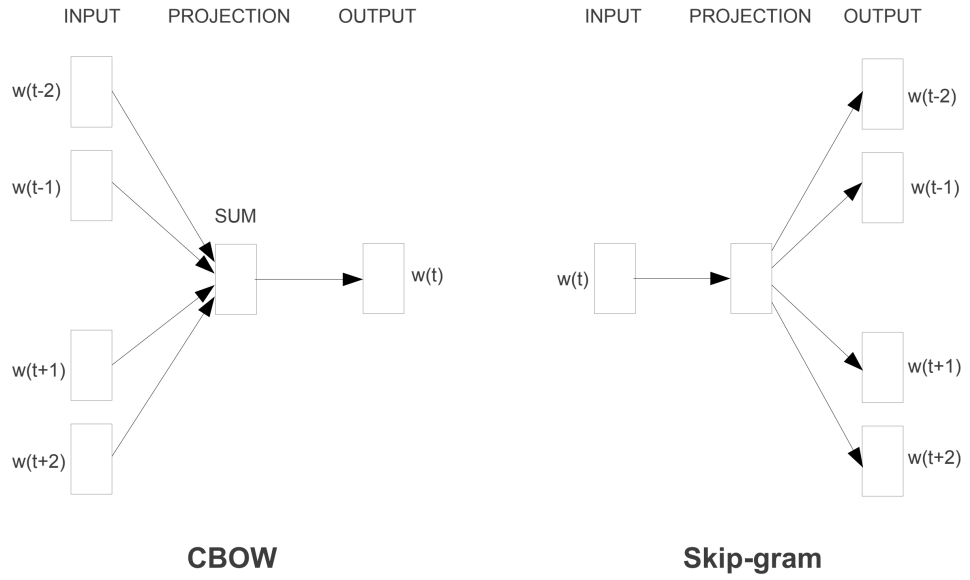


Figure 2.1: The CBOW architecture predicts the current word based on the surrounding words, and the Skip-gram predicts surrounding words given the current word [64].

2.1.1 Distributed Word Representations

Mikolov *et al.* [64] proposed two model architectures for learning distributed representations of words, continuous bag-of-words (CBOW) and continuous skip-gram models. For CBOW, the word embeddings are trained by predicting the current word based on the surrounding context words. And, the word embeddings are trained by predicting the surrounding words given the current word for skip-gram model as shown in Fig. 2.1. Pennington *et al.* [70] proposed a specific weighted least squares model that trains on global word-word co-occurrence counts and thus makes efficient use of statistics. The model produces a word vector space with meaningful substructure. They called this model as Global Vectors (GloVe) since the global corpus statistics are captured directly by the model. These word embedding approaches, mapping the word to a spe-

cific vector, have brought a huge impact as a fundamental component in natural language processing tasks. However, these embeddings can not distinguish the homonym since the embedding vector of the same word is always the same regardless of the context.

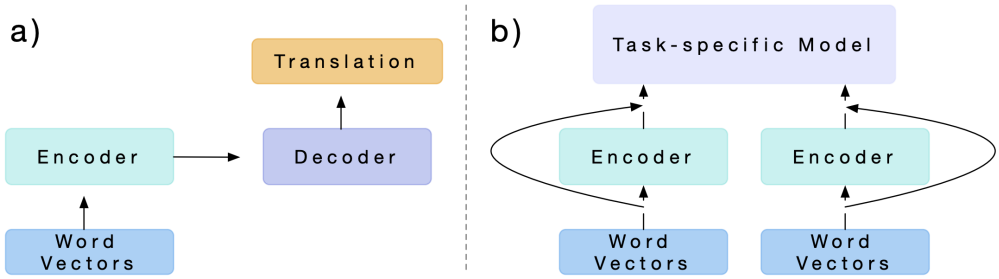


Figure 2.2: A two-layer, bidirectional LSTM is trained as the encoder of an attentional sequence-to-sequence model for machine translation and b) the trained encoder, providing contextual information, can be used for other NLP models [63].

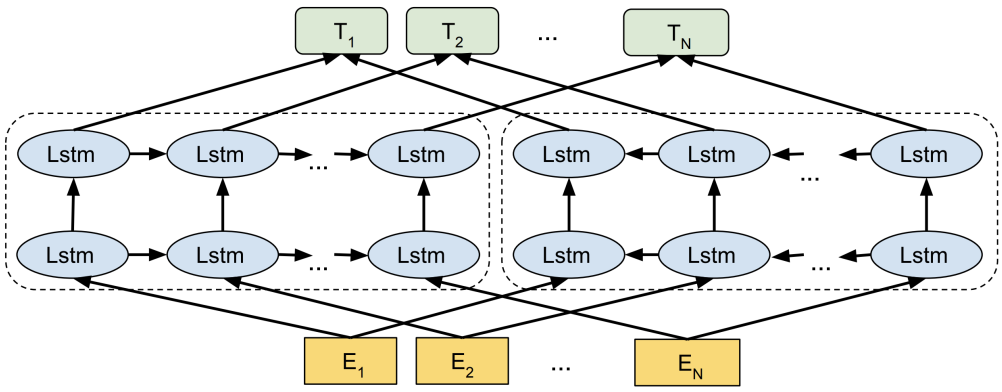


Figure 2.3: ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTMs to generate features for downstream tasks [20].

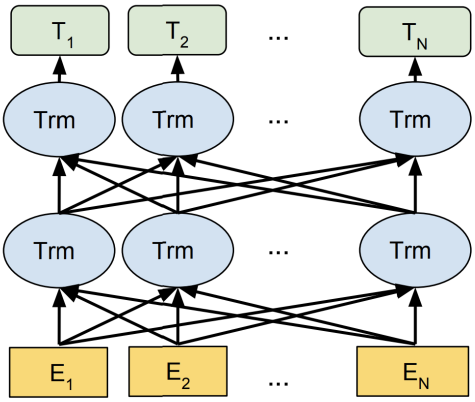


Figure 2.4: BERT uses a bidirectional Transformer. It's representations are jointly conditioned on both left and right context in all layers [20].

2.1.2 Contextualized Word Representations

Recently, contextualized word embeddings were introduced for the purpose of understanding the context beyond the single word embedding, adding more component such as RNN with a simple distributed word embeddings to inject more contextual information [63, 71]. The embedding vector of the same word can be different in the different context. McCann *et al*. [63] used a deep LSTM encoder, called CoVe, from an attentional sequence-to-sequence model trained for machine translation (MT) to contextualize word vectors (Fig. 2.2). Peters *et al*. [71] proposed a deep bidirectional language model (biLM) as a contextualized word embeddings, ELMo, using a bi-directional LSTM, which is pretrained on a large text corpus as shown in Fig. 2.3.

More recently, Devlin *et al*. [20] proposed BERT (Bidirectional Encoder Representations from Transformers) which is designed to pre-train deep bidirectional representations from the unlabeled text by jointly conditioning on both left and right context in all layers. It is trained by using a masked language model

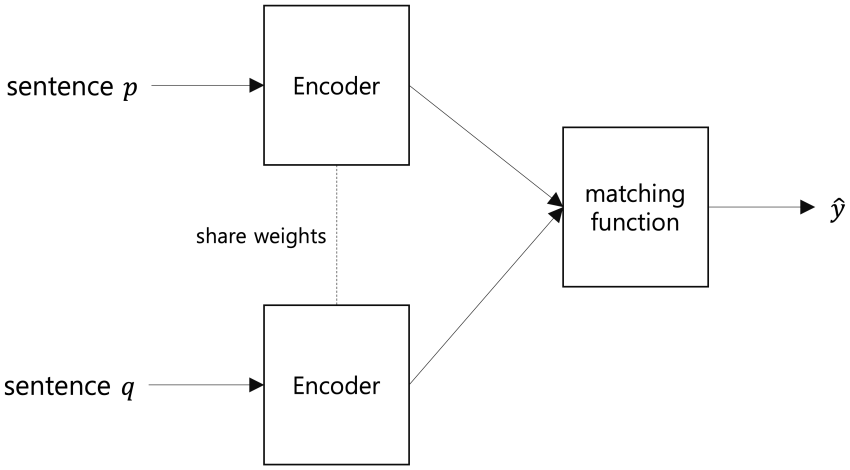


Figure 2.5: Siamese structure for the sentence pair matching architecture. Sentence p and q are encoded by the encoder and the encoders can have the same weights. The matching function determines the matching degree of the two sentences whether they have a relationship or not.

(MLM) pre-training objective. The masked language model randomly masks out some of the input tokens, and the objective is to predict the original token of the masked word based only on its contextual information. And the stream using contextualized embeddings [20, 61, 15] has shown a significant improvement on a variety of the downstream tasks including question answering.

2.2 Sentence Matching

Sentence matching is a fundamental task of natural language processing aiming to determine the relationship between two sentences. For example, paraphrase identification [17] identifies whether the two sentences have an identical meaning or not. This can be used in a community-based question answering task [93, 86] finding the paraphrased questions from the previously answered ques-

tions. In the answer sentence selection task, it is used to find the answerable sentences among a number of the answer sentence candidates through the matching between the question and the answer [40, 105]. In the natural language inference task, the matching models recognize the textual entailment between a premise and a hypothesis whether a hypothesis sentence can reasonably be inferred from a premise sentence [4, 97]. However determining the logical and semantic relationship between two sentences is not easy due to the diverse expressions of the sentences, the problem of the semantic gap [59]. Thanks to the development of deep neural networks, there was big progress on matching approaches. We can divide these into two widely studied approaches, encoding-based methods and joint methods. In the encoding-based methods, the sentences are encoded by the separated models to extract their own representations without any cross-interaction between the sentences. Then, a matching function determines the relationship using these independent representations. For the structure of the sentence pair matching model, the siamese network, as shown in Fig. 2.5, using the same structure such as RNN, CNN, or Transformer for each text has been widely used [37, 16, 12, 67, 38]. On the other hand, there are joint methods utilizing cross information between two sentences with an attention mechanism [84, 3, 95, 83, 88, 27]. The attention mechanism endows richer information in sentence matching by providing alignment and dependency relationship between two sentences, and this cross information across two sentences brings much better performance than the encoding-based methods. In this dissertation, we propose the sentence pair matching model in a form of a joint-method to learn better textual semantics by utilizing all the layers' outputs intact as collective knowledge in the deep RNNs.

2.3 Question Answering

Question Answering is the task of answering the questions and there are a variety of types of question answering. As investigated in the previous section, there are sentence matching based question answering tasks such as community-based question answering or answer sentence selection. Community-based question answering [86, 23, 1] is a crowdsourcing service that enables users to post the questions on the system and also enables users to answer those questions. The question answering system can answer the question automatically using sentence matching as a paraphrase identification task identifying whether the user question is a paraphrase of the previously asked questions. Answer sentence selection [105, 94, 40] tasks can focus on the more answerable sentences from the user questions in the document by calculating the matching degree between the question and the answer.

In another stream, machine reading comprehension tasks have gained much attention recently. An extractive reading comprehension extracts the correct answer span from a context paragraph or document given a question. A variety of large-scale datasets make it possible to solve extractive reading comprehension. The SQuAD dataset [75] was created by crowdworkers on a set of Wikipedia articles, and the answers are composed of a segment of text or span from the corresponding reading passages. The questions might be unanswerable so that the machine has to predict that the given question can be answerable or not from the corresponding context. Yu *et al*. [66] created MS MARCO, consisting of a number of anonymized questions sampled from Bing's search query logs and corresponding passages extracted from the web documents retrieved by the Bing search system. A question in the MS MARCO dataset may have multiple

answers or no answers at all.

Long-form question answering is a system producing paragraph-length answers. This task generates elaborate and in-depth answers to open-ended questions. This involves retrieving documents relevant to the question and generating a paragraph-length answer using retrieved documents. Fan *et al*. created ELI5 [21] from the Reddit forum "Explain Like I'm Five" where an online community provides answers to questions that are comprehensible by five-year-olds. GooAQ [43] contains over 5 million questions and 3 million answers collected from the answer boxes in the Google search results. These datasets contain *why* or *how* questions in a large proportion, unlike the previously mentioned datasets. However, there is a lack of auto evaluation metrics for these generative QA datasets so far. While ROUGE-L is a common metric for assessing the quality of models for text generation tasks, it is quite unreliable to evaluate the generative QA systems yet. More sophisticated evaluation metrics should be devised to evaluate the models more correctly.

Also, there is a multiple-choice reading comprehension task that requires machines to select the correct option from a set of answer options using the given context and question. To reason the answer more precisely, we need to identify the relevant sentences from the context by matching the context with a question and answer options. A number of datasets were created for multiple-choice reading comprehension. Richardson *et al*. [78] created the text based MCTest dataset about fictional stories. Kembhavi *et al*. [42] created the multimodal multiple-choice machine comprehension dataset which aims at answering multimodal questions given the context of text and diagrams. This dataset was built from middle school science curricula as imitating a real-world setting, and it is required to understand and reason long lesson of the text and the di-

agrams simultaneously. TVQA [55, 56] and DramaQA [13] are created from popular TV shows, and they require comprehending subtitle-based dialogue as a realistic natural language. In this dissertation, we deal with multiple-choice question answering tasks requiring the matching techniques between the question, answer options, and context sentences.

2.4 Contrastive Learning and Self-supervised learning for Better Representation

Recently, self-supervised pre-training on large corpus brings significant performance improvement in language domains such as BERT [20], one of the contextualized embedding models in the previous section. The models are trained with variants of the language modeling objectives before they are fine-tuned for the specific tasks. BERT [20] used masked LM (MLM) and next sentence prediction (NSP) tasks. For MLM, the input words are randomly masked in a certain percentage and the model is trained by predicting the masked token using its context only. For NSP, the model uses an input consisting of two sentences and is trained to predict whether the two sentences are consecutive or not. ALBERT [53] used a sentence-order prediction (SOP) loss which focuses on modeling inter-sentence coherence. The SOP uses as positive examples the same technique as NSP, and as negative examples the same two consecutive sentences but with their order swapped. In ELECTRA [15], replaced token detection is used for the pre-training objective that the tokens are corrupted by replacing some tokens with plausible alternatives sampled from a small generator network. And a discriminative model is trained to predict whether each token in the corrupted input was replaced by a generator sample or not. However, these approaches

require a large-scale external corpus to train the model even though it has a generalization capability across various downstream tasks. On the other hand, we propose the new pre-training approach with only the given downstream dataset that we have to solve. We transform the problem format from predicting the answer to predicting the relevant question or context without any ground-truth annotations. We pre-train the model to have a better parameter initialization and the performance of the downstream tasks is improved using this pre-training strategy.

Contrastive representation learning is another research direction that has shown success in the computer vision domain [96, 81, 8, 44, 29, 34, 22, 30]. The goal of contrastive learning is to learn the embedding space in which similar samples stay close to each other while dissimilar ones are far apart based on metric distance learning approaches. Chen *et al*. [8] proposed the contrastive learning framework for visual representations without requiring specialized architectures or a memory bank. Khosla *et al*. [44] proposed a fully-supervised contrastive loss that the clusters of points belonging to the same class are pulled together in embedding space, while simultaneously pushing apart clusters of samples from different classes. In the language domain, Fang *et al*. [22] proposed Contrastive self-supervised Encoder Representations from Transformers (CERT) which pre-trains language representation models using contrastive self-supervised learning at the sentence level using back-translation. Gunel *et al*. [30] proposed fine-tuning objective leads to models that are more robust to different levels of noise in the fine-tuning training data and can generalize better to related tasks with limited labeled data. We also contrast the positive representation and the negative representations using given annotations in a supervised manner to separate the correct answer and the negative answers farther. We calculate the

matching scores between the noisy ground truth answer as an anchor representation and the answer options, and by mapping the positive sample closer to the anchor, we improve the model performance in chapter 5.

Chapter 3

Sentence Pair Matching

Semantic sentence matching, a fundamental technology in natural language processing, is a task to apprehend the logical and semantic relationship between two sentences. In paraphrase identification, sentence matching is utilized to identify whether two sentences have identical meaning or not (e.g. a query and a question for question retrieval). In natural language inference also known as recognizing textual entailment, it determines whether a hypothesis sentence can reasonably be inferred from a given premise sentence. For answer sentence selection task, sentence matching is required to determine the degree of matching between a question and an answer. However identifying logical and semantic relationship between two sentences is not trivial due to the problem of the semantic gap [59]. In this chapter, we propose the sentence pair matching model to learn better textual semantics by utilizing all the layers' outputs intact as collective knowledge.

3.1 Motivation

Recent advances of deep neural network enable to learn textual semantics for sentence matching. Large amount of annotated data such as Quora [17], TrecQA [94], SNLI [4], and MultiNLI [97] have contributed significantly to learning semantics as well. In the conventional methods, a matching model can be trained in two different ways [27]. The first methods are sentence-encoding-based ones where each sentence is encoded to a fixed-sized vector in a complete isolated manner and the two vectors for the corresponding sentences are used in predicting the degree of matching. In this paradigm, because two sentences have no interaction, they can not utilize interactive information during the encoding procedure. On the other hand, the others are joint methods that allow to capture interactive features like attentive information between the sentences for performance improvements.

Among the deep neural networks, recurrent networks are widely used to model the textual sequences. Deeper recurrent models such as GNMT [98] are more advantageous for learning long sequences and outperform the shallower architectures. However, the attention mechanism is unstable in deeper models with the well-known vanishing gradient problem. Though [98] uses residual connection between recurrent layers to allow better information and gradient flow, there are some limitations. The recurrent hidden or attentive features are not preserved intact through residual connection because the summation operation may impede the information flow in deep networks.

Inspired by Densenet [36], we, in this chapter, propose a densely-connected recurrent network where the recurrent hidden features are retained to the uppermost layer. In addition, instead of the conventional summation operation, the

concatenation operation is used in combination with the attention mechanism to preserve co-attentive information better.

The proposed architecture shown in Figure 3.1 is called DRCN which is an abbreviation for *Densely-connected Recurrent and Co-attentive neural Network*. The proposed DRCN can utilize the increased representational power of deeper recurrent networks and attentive information. This approach makes the features getting bigger because of the concatenation function. DenseNet, from which we inspired, used convolutional layers as its bottleneck component to alleviate this problem of increasing vector size. However, this method could not preserve the lower layers' features intact, and it could not reuse the lower layers' features at the higher layers since this approach could not accomplish the real dense connections in all layers. In this chapther, to alleviate the problem of an ever-increasing feature vector size due to concatenation operations, we adopted an autoencoder and forwarded a fixed-length vector to the higher layer recurrent module as shown in the figure, and it allows the model to reuse all the layers' outputs without any deformation. DRCN is, to our best knowledge, the first generalized version of DenseRNN which is expandable to deeper layers with the property of controllable feature sizes by the use of an autoencoder.

The contributions of this work are as follows. First, we design the simple yet powerful DenseRNN architecture for sentence pair matching to learn better semantics with reusable information by the concatenation operation. Second, we propose an autoencoder as our bottleneck component to mitigate the problem of the ever-increasing feature vector size without any deformation. Finally, our model achieves the competitive performance on three highly challenging sentence matching tasks.

The rest of this chapter is organized as follows. We address the related works

in section 3.1. And, we present the proposed method and experimental results with an analysis in sections 3.2 and 3.3 respectively. Finally, we conclude the proposed method in section 3.4.

3.2 Related Work

Earlier approaches of sentence pair matching mainly relied on conventional methods such as syntactic features, transformations or relation extraction [79, 94]. These are restrictive in that they work only on very specific tasks.

The developments of large-scale annotated datasets [4, 97] and deep learning algorithms have led a big progress on matching natural language sentences. Furthermore, the well-established attention mechanisms endowed richer information for sentence matching by providing alignment and dependency relationship between two sentences. The release of the large-scale datasets also has encouraged the developments of the learning-centered approaches to semantic representation. The first type of these approaches is sentence-encoding-based methods [16, 12, 67, 85] where sentences are encoded into their own sentence representation without any cross-interaction. Then, a classifier such as a neural network is applied to decide the relationship based on these independent sentence representations. These sentence-encoding-based methods are simple to extract sentence representation and are able to be used for transfer learning to other natural language tasks [16]. On the other hand, the joint methods, which make up for the lack of interaction in the former methods, use cross-features as an attention mechanism to express the word- or phrase-level alignments for performance improvements [95, 7, 27, 101].

Recently, the architectural developments using deeper layers have led more

progress in performance. The residual connection is widely and commonly used to increase the depth of a network stably [35, 98]. Huang *et al*. [36] enable the features to be connected from lower to upper layers using the concatenation operation without any loss of information on lower-layer features. More recently, transformer-based pre-trained language models [20, 61] trained from the large-scale unlabeled corpus are used for natural language understanding tasks including sentence matching tasks. However, these models are too heavy with over hundreds of millions of parameters while they show a powerful performance.

External resources are also used for sentence matching. Chen *et al*. [6, 7] used syntactic parse trees or lexical databases like WordNet to measure the semantic relationship among the words and Pavlick *et al*. [69] added interpretable semantics to the paraphrase database. Unlike these, in this study, we do not use any such external resources. Our work belongs to the joint approaches which uses densely-connected recurrent and co-attentive information to enhance representation power for semantic sentence matching.

3.3 Method

In this section, we describe our sentence matching architecture DRCN which is composed of the following three components: (1) word representation layer, (2) attentively connected recurrent neural network and (3) interaction and prediction layer. We denote two input sentences as $P = \{p_1, p_2, \cdots, p_I\}$ and $Q = \{q_1, q_2, \cdots, q_J\}$ where p_i/q_j is the i^{th}/j^{th} word of the sentence P/Q and I/J is the word length of P/Q. The overall architecture of the proposed DRCN is shown in Fig. 3.1.

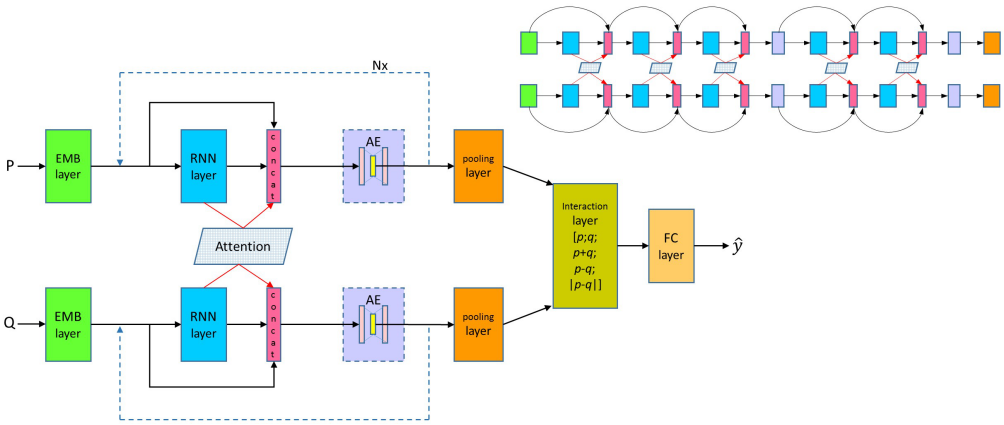


Figure 3.1: General architecture of our Densely-connected Recurrent and Co-attentive neural Network (DRCN). Dashed arrows indicate that a group of RNN-layer, concatenation and AE can be repeated multiple (N) times (like a repeat mark in a music score). The bottleneck component denoted as AE, inserted to prevent the ever-growing size of a feature vector, is optional for each repetition. The upper right diagram is our specific architecture for experiments with 5 RNN layers ($N = 4$).

3.3.1 Word Representation Layer

To construct the word representation layer, we concatenate word embedding, character representation and the exact matched flag which was used in [27].

In word embedding, each word is represented as a d-dimensional vector by using a pre-trained word embedding method such as GloVe [70] or Word2vec [64]. In our model, a word embedding vector can be updated or fixed during training. The strategy whether to make the pre-trained word embedding be trainable or not is heavily task-dependent. Trainable word embeddings capture the characteristics of the training data well but can result in overfitting. On the other hand, fixed (non-trainable) word embeddings lack flexibility on task-specific data, while it can be robust for overfitting, especially for less frequent words. We use both the trainable embedding $e_{p_i}^{tr}$ and the fixed (non-trainable) embedding $e_{p_i}^{fix}$ to let them play complementary roles in enhancing the performance of our model. This technique of mixing trainable and non-trainable word embeddings is simple but yet effective.

The character representation c_{p_i} is calculated by feeding randomly initialized character embeddings into a convolutional neural network with the max-pooling operation. The character embeddings and convolutional weights are jointly learned during training.

Like [27], the exact match flag f_{p_i} is activated if the same word is found in the other sentence.

Our final word representational feature p_i^w for the word p_i is composed of

four components as follows:

$$
\begin{aligned}
e_{p_i}^{tr} = E^{tr}(p_i), \quad e_{p_i}^{fix} = E^{fix}(p_i) \\
c_{p_i} = \text{Char-Conv}(p_i) \\
p_i^w = [e_{p_i}^{tr}; e_{p_i}^{fix}; c_{p_i}; f_{p_i}].
\end{aligned}
\tag{3.1}
$$

Here, E^{tr} and E^{fix} are the trainable and non-trainable (fixed) word embeddings respectively. Char-Conv is the character-level convolutional operation and $[\cdot\ ;\ \cdot]$ is the concatenation operator. For each word in both sentences, the same above procedure is used to extract word features.

3.3.2 Densely connected Recurrent Networks

The ordinal stacked RNNs (Recurrent Neural Networks) are composed of multiple RNN layers on top of each other, with the output sequence of previous layer forming the input sequence for the next. More concretely, let H_l be the l^{th} RNN layer in a stacked RNN. Note that in our implementation, we employ the bidirectional LSTM (BiLSTM) as a base block of H_l. At the time step t, an ordinal stacked RNN is expressed as follows:

$$
h_t^l = H_l(x_t^l, h_{t-1}^l), \quad x_t^l = h_t^{l-1}. \tag{3.2}
$$

While this architecture enables us to build up higher level representation, deeper networks have difficulties in training due to the exploding or vanishing gradient problem.

To encourage gradient to flow in the backward pass, residual connection [35] is introduced which bypasses the non-linear transformations with an identity mapping. Incorporating this into (3.2), it becomes

$$
h_t^l = H_l(x_t^l, h_{t-1}^l), \quad x_t^l = h_t^{l-1} + x_t^{l-1}. \tag{3.3}
$$

However, the summation operation in the residual connection may impede the information flow in the network [36]. Motivated by Densenet [36], we employ direct connections using the concatenation operation from any layer to all the subsequent layers so that the features of previous layers are not to be modified but to be retained as they are as depicted in Figure 3.1. The densely connected recurrent neural networks can be described as

$$h_t^l = H_l(x_t^l, h_{t-1}^l), \quad x_t^l = [h_t^{l-1}; x_t^{l-1}]. \tag{3.4}$$

The concatenation operation enables the hidden features to be preserved until they reach to the uppermost layer and all the previous features work for prediction as collective knowledge [36].

3.3.3 Densely-connected Co-attentive networks

Attention mechanism, which has largely succeeded in many domains [98, 91], is a technique to learn effectively where a context vector is matched conditioned on a specific sequence.

Given two sentences, a context vector is calculated based on an attention mechanism focusing on the relevant part of the two sentences at each RNN layer. The calculated attentive information represents soft-alignment between two sentences. In this dissertation, we also use an attention mechanism. We incorporate co-attentive information into densely connected recurrent features using the concatenation operation, so as not to lose any information (Fig. 3.1). This concatenated recurrent and co-attentive features which are obtained by densely connecting the features from the undermost to the uppermost layers, enrich the collective knowledge for lexical and compositional semantics.

The attentive information a_{p_i} of the i^{th} word $p_i \in P$ against the sentence Q is calculated as a weighted sum of h_{q_j}'s which are weighted by the softmax weights as follows:

$$\begin{aligned} a_{p_i} &= \sum_{j=1}^{J} \alpha_{i,j} h_{q_j} \\ \alpha_{i,j} &= \frac{exp(e_{i,j})}{\sum_{k=1}^{J} exp(e_{i,k})} \\ e_{i,j} &= \cos(h_{p_i}, h_{q_j}) \end{aligned} \tag{3.5}$$

Similar to the densely connected RNN hidden features, we concatenate the attentive context vector a_{p_i} with triggered vector h_{p_i} so as to retain attentive information as an input to the next layer:

$$h_t^l = H_l(x_t^l, h_{t-1}^l), \quad x_t^l = [h_t^{l-1}; a_t^{l-1}; x_t^{l-1}]. \tag{3.6}$$

3.3.4 Bottleneck Component

Our network uses all layers' outputs as a community of semantic knowledge. However, this network is a structure with increasing input features as layers get deeper, and has a large number of parameters especially in the fully-connected layer. To address this issue, we employ an autoencoder as a bottleneck component. Autoencoder is a compression technique that reduces the number of features while retaining the original information, which can be used as a distilled semantic knowledge in our model. Furthermore, this component increased the test performance by working as a regularizer in our experiments.

3.3.5 Interaction and Prediction Layer

To extract a proper representation for each sentence, we apply the step-wise max-pooling operation over densely connected recurrent and co-attentive fea-

tures (pooling in Fig. 3.1). More specifically, if the output of the final RNN layer is a 100d vector for a sentence with 30 words, a 30×100 matrix is obtained which is max-pooled column-wise such that the size of the resultant vector p or q is 100. Then, we aggregate these representations p and q for the two sentences P and Q in various ways in the interaction layer and the final feature vector v for semantic sentence matching is obtained as follows:

$$v = [p; q; p + q; p - q; |p - q|]. \tag{3.7}$$

Here, the operations $+$, $-$ and $|\cdot|$ are performed element-wise to infer the relationship between two sentences. The element-wise subtraction $p - q$ is an asymmetric operator for one-way type tasks such as natural language inference or answer sentence selection.

Finally, based on previously aggregated features v, we use two fully-connected layers with ReLU activation followed by one fully-connected output layer. Then, the softmax function is applied to obtain a probability distribution of each class. The model is trained end-to-end by minimizing the multi-class cross entropy loss and the reconstruction loss of autoencoders.

3.4 Experiment

In this section, we evaluate our matching model on five popular and well-studied benchmark datasets for three challenging sentence matching tasks: (i) SNLI and MultiNLI for natural language inference; (ii) Quora Question Pair for paraphrase identification of questions; and (iii) TrecQA and SelQA for answer sentence selection in question answering. Details about the above datasets are shown in the following section.

3.4.1 Datasets

A. SNLI is a collection of 570k human written sentence pairs based on image captioning, supporting the task of natural language inference [4]. The labels are composed of entailment, neutral and contradiction. The data splits are provided in [4]. Examples of SNLI dataset are shown in Table 3.1.

B. MultiNLI, also known as Multi-Genre NLI, has 433k sentence pairs whose size and mode of collection are modeled closely like SNLI. MultiNLI offers ten distinct genres (FACE-TO-FACE, TELEPHONE, 9/11, TRAVEL, LETTERS, OUP, SLATE, VERBATIM, GOVERNMENT and FICTION) of written and spoken English data. Also, there are matched dev/test sets which are derived from the same sources as those in the training set, and mismatched sets which do not closely resemble any seen at training time. The data splits are provided in [97].

C. Quora Question Pair consists of over 400k question pairs based on actual `quora.com` questions. Each pair contains a binary value indicating whether the two questions are paraphrase or not. The training/dev/test splits for this dataset are provided in [95].

D. TrecQA provided in [94] was collected from TREC Question Answering tracks 8-13. There are two versions of data due to different pre-processing methods, namely clean and raw [76]. We evaluate our model on both data and follow the same data split as provided in [94]. We use official evaluation metrics of MAP (Mean Average Precision) and MRR (Mean Reciprocal Rank), which are standard metrics in information retrieval and learning to rank tasks.

E. SelQA consists of questions generated through crowdsourcing and the answer senteces are extracted from the ten most prevalent topics (Arts, Country,

Premise *two bicyclists in spandex and helmets in a race pedaling uphill.*
Hypothesis *A pair of humans are riding their bicycle with tight clothing, competing with each other.*
Label {***entailment***; *neutral; contradiction*}
Premise *Several men in front of a white building.*
Hypothesis *Several people in front of a gray building.*
Label {*entailment; neutral;* ***contradiction***}

Table 3.1: Examples of *natural language inference*.

Food, Historical Events, Movies, Music, Science, Sports, Travel and TV) in the English Wikipedia. We also use MAP and MRR for our evaluation metrics, and the data splits are provided in [40].

3.4.2 Implementation Details

We initialized word embedding with 300d GloVe vectors [70] pre-trained from the 840B Common Crawl corpus [70], while the word embeddings for the out-of-vocabulary words were initialized randomly. We also randomly initialized character embedding with a 16d vector and extracted 32d character representation with a convolutional network. For the densely-connected recurrent layers, we stacked 5 layers each of which have 100 hidden units. We set 1000 hidden units with respect to the fully-connected layers. The dropout was applied after the word and character embedding layers with a keep rate of 0.5. It was also applied before the fully-connected layers with a keep rate of 0.8. For the bottleneck component, we set 200 hidden units as encoded features of the autoencoder with a dropout rate of 0.2. The batch normalization was applied on

the fully-connected layers, only for the one-way type datasets. The RMSProp optimizer with an initial learning rate of 0.001 was applied. The learning rate was decreased by a factor of 0.85 when the dev accuracy does not improve. All weights except embedding matrices are constrained by L2 regularization with a regularization constant $\lambda = 10^{-6}$. The sequence lengths of the sentence are all different for each dataset: 35 for SNLI, 55 for MultiNLI, 25 for Quora question pair and 50 for TrecQA. The learning parameters were selected based on the best performance on the dev set. We employed 8 different randomly initialized sets of parameters with the same model for our ensemble approach.

3.4.3 Experimental Results

SNLI and MultiNLI

We evaluated our model on the natural language inference task over SNLI and MultiNLI datasets. Table 3.2 and 3.3 show the results on SNLI dataset of our model with other published models. Among them, ESIM+ELMo and LM-Transformer are the previous state-of-the-art models. However, they use additional contextualized word representations from language models as an externel knowledge. The proposed DRCN obtains an accuracy of 88.9% which is a competitive score although we do not use any external knowledge like ESIM+ELMo and LM-Transformer. The ensemble model achieves an accuracy of 90.1%, which sets the state-of-the-art performance at the time our paper was published. The development of the large-scale pre-trained language model, BERT [20] shows the best performance of this task as 91%, however, the number of parameters is 110m which is larger than our proposed model.

Our ensemble model with 53m parameters (6.7m$\times$8) outperforms the LM-

Sentence encoding-based method		
Models	**Acc.**	$\|\theta\|$
BiLSTM-Max [16]	84.5	40m
Gumbel TreeLSTM [12]	85.6	2.9m
CAFE [88]	85.9	3.7m
Gumbel TreeLSTM [12]	86.0	10m
Residual stacked [67]	86.0	29m
Reinforced SAN [85]	86.3	3.1m
Distance SAN [38]	86.3	3.1m
DRCN (- Attn, - Flag)	**86.5**	5.6m

Table 3.2: Classification accuracy (%) of encoding-based method for natural language inference on SNLI test set. $|\theta|$ denotes the number of parameters in each model.

Transformer whose the number of parameters is 85m. Furthermore, in case of the encoding-based method, we obtain the best performance of 86.5% without the co-attention and exact match flag.

Table 3.4 shows the results on MATCHED and MISMATCHED problems of MultiNLI dataset. Our plain DRCN has a competitive performance without any contextualized knowledge. And, by combining DRCN with the ELMo, one of the contextualized embeddings from language models, our model outperforms the LM-Transformer which has 85m parameters with fewer parameters of 61m. Like the case of SNLI, BERT, which has 110m parameters, shows the best re-

Joint method (cross-features available)		
Models	**Acc.**	$\|\theta\|$
DIIN [27]	88.0 / 88.9	4.4m
ESIM [7]	88.0 / 88.6	4.3m
BCN+CoVe+Char [63]	88.1 / -	22m
DR-BiLSTM [26]	88.5 / 89.3	7.5m
CAFE [88]	88.5 / 89.3	4.7m
KIM [6]	88.6 / 89.1	4.3m
ESIM+ELMo [71]	88.7 / 89.3	8.0m
LM-Transformer [73]	**89.9** / -	85m
BERT [60]	**91.0** / -	110m
DRCN (- AE)	88.7 / -	20m
DRCN	88.9 / **90.1**	6.7m

Table 3.3: Classification accuracy (%) of joint method for natural language inference on SNLI test set. $|\theta|$ denotes the number of parameters in each model.

sults on the MultiNLI task. From this point of view, the combination of our model with a contextualized knowledge is a good option to enhance the performance.

Quora Question Pair

Table 3.5 shows our results on the Quora question pair dataset. BiMPM using the multi-perspective matching technique between two sentences reports baseline performance of a L.D.C. network [95]. We obtained accuracies of 90.15% and 91.30% in single and ensemble methods, respectively, surpassing the previous

Models	Accuracy (%)	
	MATCHED	MISMATCHED
ESIM [97]	72.3	72.1
DIIN [27]	78.8	77.8
CAFE [88]	78.7	77.9
LM-Transformer [73]	**82.1**	**81.4**
BERT [73]	**84.6**	**83.4**
DRCN	79.1	78.4
DIIN* [27]	80.0	78.7
CAFE* [88]	80.2	79.0
DRCN*	**80.6**	**79.5**
DRCN+ELMo*	**82.3**	**81.4**

Table 3.4: Classification accuracy for natural language inference on MultiNLI test set. * denotes ensemble methods.

state-of-the-art model of DIIN.

TrecQA and SelQA

Table 3.6 and Table 3.7 show the performance of different models on TrecQA and SelQA datasets for answer sentence selection task that aims to select a set of candidate answer sentences given a question. Most competitive models [84, 3, 95, 83] also use attention methods for words alignment between question and candidate answer sentences. However, the proposed DRCN using collective attentions over multiple layers, achieves the competitive performance, exceeding the previous state-of-the-art performance significantly on both datasets.

Models	Accuracy (%)
Siamese-LSTM [95]	82.58
MP LSTM [95]	83.21
L.D.C. [95]	85.55
BiMPM [95]	88.17
pt-DecAttchar.c [90]	88.40
DIIN [27]	89.06
DRCN	**90.15**
DIIN* [27]	89.84
DRCN*	**91.30**

Table 3.5: Classification accuracy for paraphrase identification on Quora question pair test set. * denotes ensemble methods.

3.4.4 Analysis

Ablation study

We conducted an ablation study on the SNLI dev set as shown in Table 3.8, where we aim to examine the effectiveness of our word embedding technique as well as the proposed densely-connected recurrent and co-attentive features. First, we verified the effectiveness of the autoencoder as a bottleneck component in (2). Although the number of parameters in the DRCN significantly decreased as shown in Table 3.3, we could see that the performance was rather higher because of the regularization effect. Second, we study how the technique of mixing trainable and fixed word embeddings contributes to the performance in models (3-4). After removing E^{tr} or E^{fix} in eq. (3.1), the performance degraded, slightly. The trainable embedding E^{tr} seems more effective than the

Models	MAP	MRR
PWIM [33]	0.758	0.822
MP CNN [32]	0.762	0.830
HyperQA [87]	0.770	0.825
PR+CNN [76]	0.780	0.834
DRCN	**0.804**	**0.862**

(a) TrecQA: *raw version*

Models	MAP	MRR
HyperQA [87]	0.801	0.877
BiMPM [95]	0.802	0.875
Comp.-Aggr. [3]	0.821	0.899
IWAN [84]	0.822	0.889
DRCN	**0.830**	**0.908**

(b) TrecQA: *clean version*

Table 3.6: Performance for answer sentence selection on TrecQA.

Models	MAP	MRR
CNN-DAN [80]	0.866	0.873
CNN-hinge [80]	0.876	0.881
ACNN [83]	0.874	0.880
AdaQA [83]	0.891	0.898
DRCN	**0.925**	**0.930**

Table 3.7: Performance for answer sentence selection on selQA test set.

fixed embedding E^{fix}. Next, the effectiveness of dense connections was tested in models (5-9). In (5-6), we removed dense connections only over co-attentive or recurrent features, respectively. The result shows that the dense connections over attentive features are more effective. In (7), we removed dense connections over both co-attentive and recurrent features, and the performance degraded to 88.5%. In (8), we replace dense connection with residual connection only over recurrent and co-attentive features. It means that only the word embedding features are densely connected to the uppermost layer while recurrent and atten-

Models	**Accuracy (%)**
(1) DRCN	89.4
(2) − autoencoder	89.1
(3) $-\ E^{tr}$	88.7
(4) $-\ E^{fix}$	88.9
(5) − dense(Attn.)	88.7
(6) − dense(Rec.)	88.8
(7) − dense(Rec. & Attn.)	88.5
(8) − dense(Rec. & Attn.) + res(Rec. & Attn.)	88.7
(9) − dense(Rec. & Attn. & Emb) + res(Rec. & Attn.)	88.4
(10) − dense(Rec. & Attn. & Emb)	87.8
(11) − dense(Rec. & Attn. & Emb) - Attn.	85.3

Table 3.8: Ablation study results on the SNLI dev sets.

tive features are connected to the upper layer using the residual connection. In (9), we removed additional dense connection over word embedding features from (8). The results of (8-9) demonstrate that the dense connection using concatenation operation over deeper layers, has more powerful capability retaining collective knowledge to learn textual semantics. The model (10) is the basic 5-layer RNN with attention and (11) is the one without attention. The result of (10) shows that the connections among the layers are important to help gradient flow. And, the result of (11) shows that the attentive information functioning as a soft-alignment is significantly effective in semantic sentence matching.

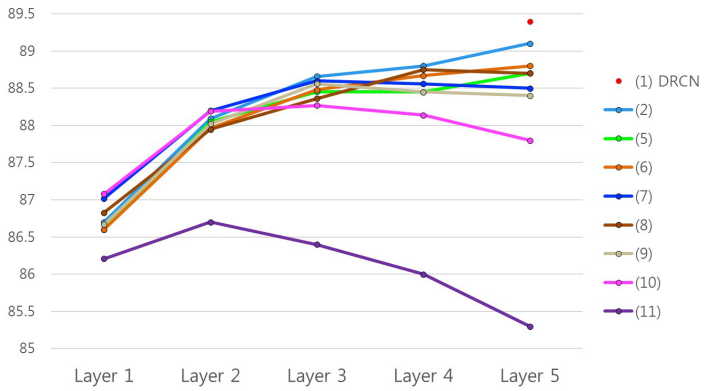
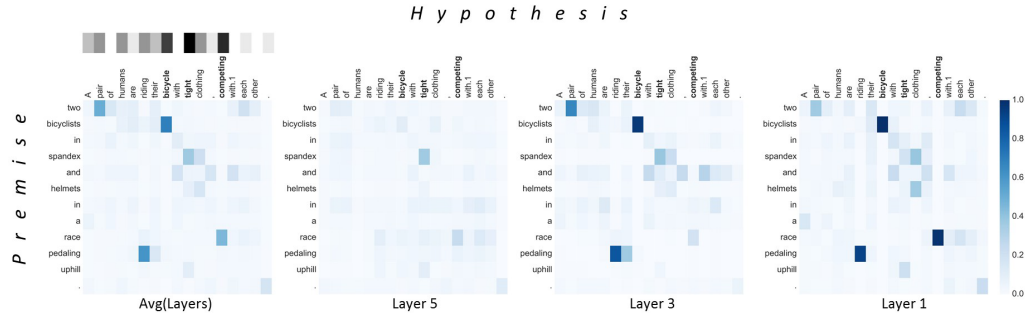


Figure 3.2: Comparison of models on every layer in ablation study. (best viewed in color)

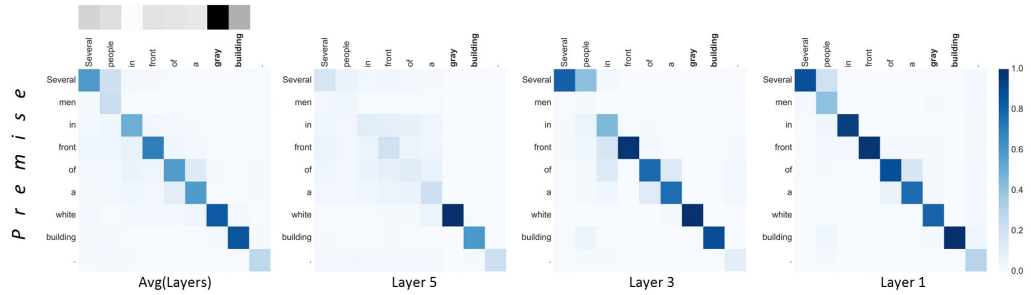
The performances of models having different number of recurrent layers are also reported in Fig. 3.2. The models (5-9) which have connections between layers, are more robust to the increased depth of network, however, the performances of (10-11) tend to degrade as layers get deeper. In addition, the models with dense connections rather than residual connections, have higher performance in general. Figure 3.2 shows that the connection between layers is essential, especially in deep models, endowing more representational power, and the dense connection is more effective than the residual connection.

Word Alignment and Importance

Our densely-connected recurrent and co-attentive features are connected to the classification layer through the max pooling operation such that all max-valued features of every layer affect the loss function and perform a kind of deep supervision [36]. Thus, we could cautiously interpret the classification results using our attentive weights and max-pooled positions. The attentive weights contain information on how two sentences are aligned and the numbers of max-pooled



(a) entailment



(b) contradiction

Figure 3.3: Visualization of attentive weights and the rate of max-pooled position. The darker, the higher.

positions in each dimension play an important role in classification.

Figure 3.3 shows the attention map ($\alpha_{i,j}$ in eq. (3.5)) on each layer of the samples in Table 3.1. The Avg(Layers) is the average of attentive weights over 5 layers and the gray heatmap right above the Avg(Layers) is the rate of max-pooled positions. The darker indicates the higher importance in classification. In the figure, we can see that *tight*, *competing* and *bicycle* are more important words than others in classifying the label. The word *tight clothing* in the hypothesis can be inferred from *spandex* in the premise. And *competing* is also inferred from *race*. Other than that, the *riding* is matched with *pedaling*, and *pair* is matched with *two*. Judging by the matched terms, the model is undoubt-

edly able to classify the label as an entailment, correctly.

In Figure 3.3 (b), most of words in both the premise and the hypothesis coexist except *white* and *gray*. In attention map of layer 1, the same or similar words in each sentence have a high correspondence (*gray* and *white* are not exactly matched but have a linguistic relevance). However, as the layers get deeper, the relevance between *white building* and *gray building* is only maintained as a clue of classification (See layer 5). Because *white* is clearly different from *gray*, our model determines the label as a contradiction.

The densely connected recurrent and co-attentive features are well-semanticized over multiple layers as collective knowledge. And the max pooling operation selects the soft-positions that may extract the clues on inference correctly.

3.4.5 Visualization on the Comparable Models

We study how the attentive weights flow as layers get deeper in each model using the dense or residual connection. We used the samples of the SNLI dev set in Table 3.1.

Figure 3.4 and 3.5 show the attention map on each layer of the models of DRCN, Table 3.8 (8), and Table 3.8 (9). In the model of Table 3.8 (8), we replaced the dense connection with the residual connection only over recurrent and co-attentive features. And, in the model of Table 3.8 (9), we removed additional dense connection over word embedding features from Table 3.8 (8). We denote the model of Table 3.8 (9) as Res1 and the model of Table 3.8 (8) as Res2 for convenience.

In Figure 3.4, DRCN does not try to find the right alignments at the upper layer if it already finds the rationale for the prediction at the relatively lower layer. This is expected that the DRCN use the features of all the preceding layers

as a collective knowledge. While Res1 and Res2 have to find correct alignments at the top layer, however, there are some misalignments such as *competing* and *bicyclists* rather than *competing* and *race* in Res2 model.

In the second example in Figure 3.5, although the DRCN couldn't find the clues at the lower layer, it gradually finds the alignments, which can be a rationale for the prediction. At the 5th layer of DRCN, the attentive weights of *gray building* and *white building* are significantly higher than others. On the other hand, the attentive weights are spread in several positions in both Res1 and Res2 which use residual connection.

Linguistic Error Analysis

We conducted a linguistic error analysis on MultiNLI, and compared DRCN with the ESIM, DIIN and CAFE. We used annotated subset provided by the MultiNLI dataset, and each sample belongs to one of the 13 linguistic categories. The results in Table 3.9 and Table 3.10 show that our model generally has a good performance than others on most categories. Especially, we can see that ours outperforms much better on the Quantity/Time category which is one of the most difficult problems. Furthermore, our DRCN shows the highest mean and the lowest stddev for both MATCHED and MISMATCHED problems, which indicates that it not only results in a competitive performance but also has a consistent performance.

3.5 Summary and Discussion

In this chapter, we introduce a densely-connected recurrent and co-attentive network (DRCN) for semantic sentence pair matching. We connect the recurrent

Category	ESIM	DIIN	CAFE	DRCN
Matched				
Conditional	**100**	57	70	65
Word overlap	50	79	82	**89**
Negation	76	78	76	**80**
Antonym	67	**82**	**82**	**82**
Long Sentence	75	81	79	**83**
Tense Difference	73	**84**	82	82
Active/Passive	88	93	**100**	87
Paraphrase	89	88	88	**92**
Quantity/Time	33	53	53	**73**
Coreference	**83**	77	80	80
Quantifier	69	74	75	**78**
Modal	78	**84**	81	81
Belief	65	**77**	**77**	76
Mean	72.8	77.46	78.9	**80.6**
Stddev	16.6	10.75	10.2	**6.7**

Table 3.9: Accuracy (%) of Linguistic correctness on MultiNLI dev sets:matched.

and co-attentive features from the bottom to the top layer without any deformation. These intact features over multiple layers compose a community of semantic knowledge and outperform the previous deep RNN models using residual connections. In doing so, bottleneck components are inserted to reduce the feature size of the network.

Category	ESIM	DIIN	CAFE	DRCN
Mismatched				
Conditional	60	69	85	**89**
Word overlap	62	**92**	87	89
Negation	71	77	**80**	78
Antonym	58	**80**	**80**	**80**
Long Sentence	69	73	77	**84**
Tense Difference	79	78	**89**	83
Active/Passive	91	70	90	**100**
Paraphrase	84	**100**	95	90
Quantity/Time	54	69	62	**80**
Coreference	75	79	83	**87**
Quantifier	72	78	80	**82**
Modal	76	75	81	**87**
Belief	67	81	83	**85**
Mean	70.6	78.53	82.5	**85.7**
Stddev	10.2	8.55	7.6	**5.5**

Table 3.10: Accuracy (%) of Linguistic correctness on MultiNLI dev sets:mismatched.

Our proposed model is the first generalized version of DenseRNN which can be expanded to deeper layers with the property of controllable feature sizes by the use of an autoencoder. We additionally show the interpretability of our model using the attentive weights and the rate of max-pooled positions. Our model achieves the competitive performance on most of the datasets of three

highly challenging sentence matching tasks.

Our proposed method using the collective semantic knowledge is expected to be applied to not only the various other natural language tasks including question answering but also the recent neural architectures such as Transformers. For example, there is a residual connection followed by a feed-forward layer in each transformer block. We could replace these operations with a dense connection followed by an autoencoder. The encoder part of the autoencoder is expected to play a role of a feed-forward layer. It'd be an interesting future work whether our proposed approach can be applied to the recent architectures and can bring the performance improvement.

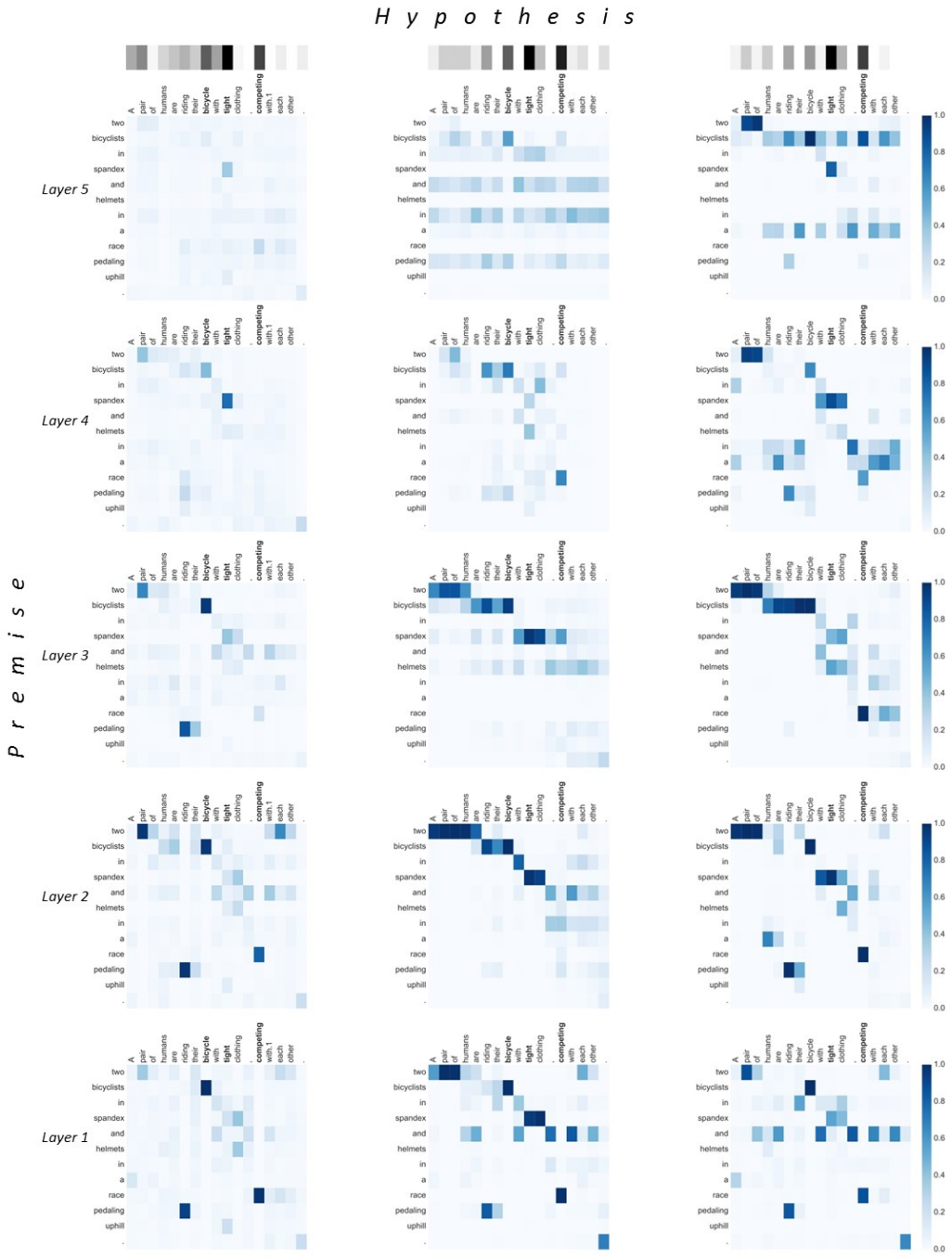


Figure 3.4: Visualization of attentive weights on the *entailment* example. The premise is "*two bicyclists in spandex and helmets in a race pedaling uphill.*" and the hypothesis is "*A pair of humans are riding their bicycle with tight clothing, competing with each other.*". The attentive weights of DRCN, Res1, and Res2 are presented from left to right.

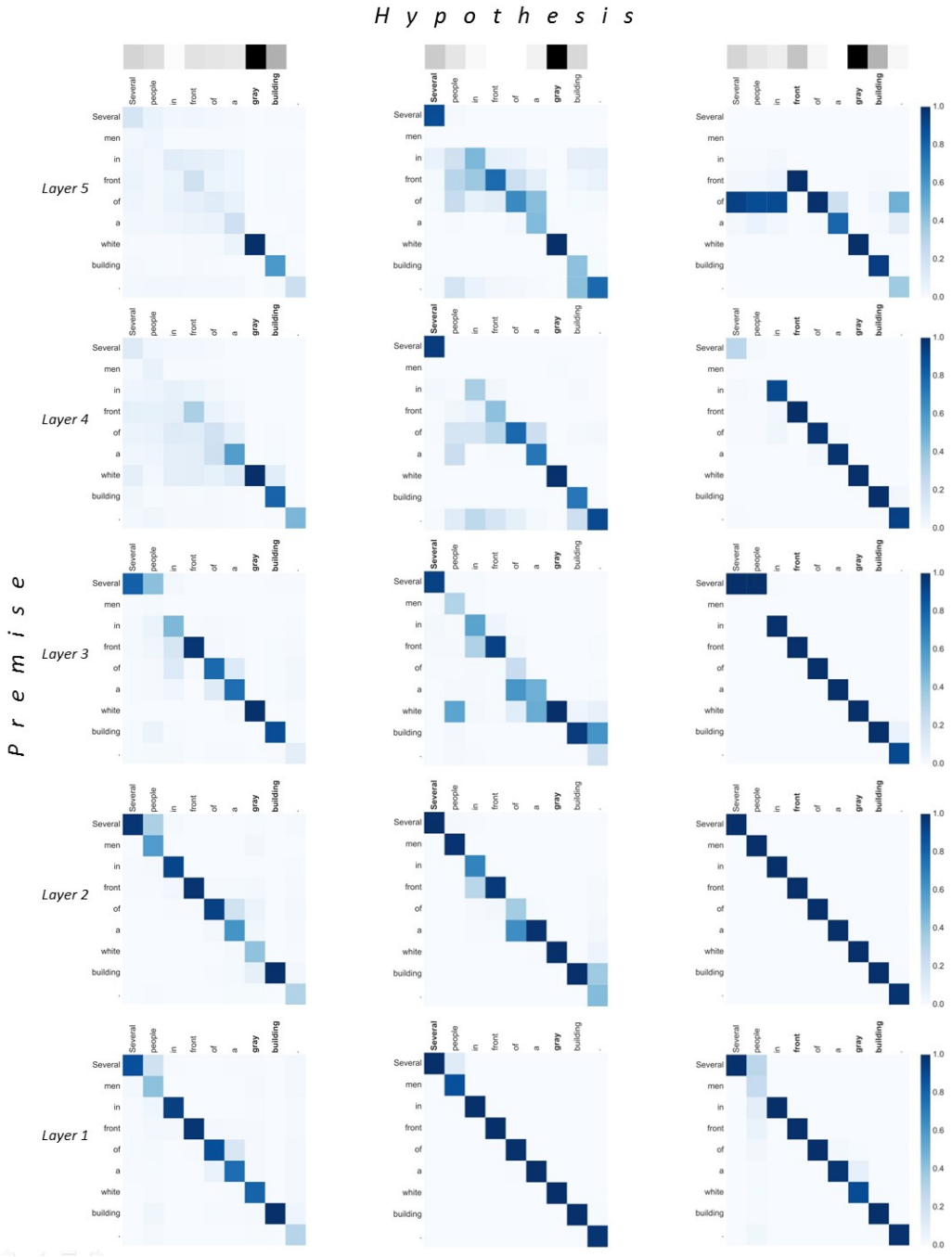


Figure 3.5: Visualization of attentive weights on the *contradiction* example. The premise is "*Several men in front of a white building.*" and the hypothesis is "*Several people in front of a gray building.*". The attentive weights of DRCN, Res1, and Res2 are presented from left to right.

Chapter 4

Context based Question Answering with Sentence Matching

In this chapter, we introduce our context-based question answering models especially for multiple-choice machine reading comprehension using sentence matching. Machine reading comprehension is a challenging task, requiring a machine to reason the answer to the given question based on the context such as a paragraph, document, or even a bunch of documents. Among them, the multiple-choice machine reading comprehension task has a few answer candidates. We, in this chapter, propose context matching approaches for four highly competitive QA datasets: Textbook QA, TVQA, TVQA+, and DramaQA. The Textbook QA (TQA) task [42] can describe the real-life process of a student who learns new knowledge from books and practices to solve related problems (Figure 4.1). It also has several novel characteristics as a realistic dataset. Since the TQA contains visual contents as well as textual contents, it requires to solve multi-modal QA. Moreover, the formats of questions are various which include both text-related questions and diagram-related questions. Compared to other

QA datasets, the context part of TQA has more complexity in the aspect of the context length. It requires comprehending long lessons to obtain knowledge. Therefore, it is important to extract exact knowledge from long texts. Since TQA is a science textbook of the middle school, there are a number of technical terminologies. And, it tells that we have to concern not only with semantic matching but also keyword matching considerably regarding the technical terminologies. We establish a context graph by matching the question and answer keywords to the terms of the context and propose a novel module based on graph convolution networks (GCN) [51] to extract proper knowledge for solving questions.

On the other hand, TV show-based video QA datasets such as TVQA, TVQA+, and DramaQA are built upon the TV shows. Since these TV shows closely reflect our daily life, comprehending these contents of TV shows helps to learn the real-world life such as human behaviors or conversations. Recent years have witnessed significant improvements in vision and language communities, which have consequently led to substantial attention in vision-language multi-modality tasks such as visual grounding [72], image captioning [11], and visual question answering [2, 28]. Furthermore, as video becomes ubiquitous, as a daily source of information and communication, video-language tasks such as video captioning [106], video moment retrieval [58], and video question answering (video QA) [55, 56] are emerging as important topics. Among these topics, video QA is especially challenging, as it requires fine-grained understanding of both video and language. Figure 4.2 shows an example of multiple-choice video QA from the TVQA dataset. The multiple-choice video QA task requires the model to select the correct answer given a question, corresponding video frames, and subtitles.

To address video QA, several works have utilized early stage fusion method

[50, 65] to merge two different modalities, while other recent works [55, 56] have employed late stage fusion method, which extracts representation from language and vision independently during the early stage of framework, and combines them in QA-aware manners. To obtain further fine-grained information from videos, Kim *et al*. [46] has generated captions using a dense caption model [102] to translate vision modality to that of language. Furthermore, Geng *et al*. [25] has reckoned that explicitly replacing the predicted regions corresponding to a person from object detection with the name of protagonists helps the model to answer the questions.

In addition, we present a locally aligned attention mechanism to selectively extract video representations corresponding to the given subtitles. Previous works [55, 56, 46] have utilized attention mechanisms on video sequences and subtitles with either question and answer pairs respectively or with the subtitles in a globally aligned manner. In contrast, we hypothesize that it is desirable to apply a direct attention mechanism that computes attention score between two modalities in locally aligned fashion. Performing attention in locally aligned fashion is beneficial to the model's performance, since it prevents the model from reasoning with unnecessary information.

In this task, we also utilize sentence matching to solve the video QA which has a context consisting of the subtitle. The subtitle is composed of multiple utterances with a sentence format. We match each sentence with the question and the answer option to focus on the more relevant part to answer correctly by localizing the relevant part as auxiliary learning.

Our contributions are summarized as follows. First, we propose the approach that can narrow down the context scope with word matching between the context and the question/answer by concentrating on the essential keywords (or

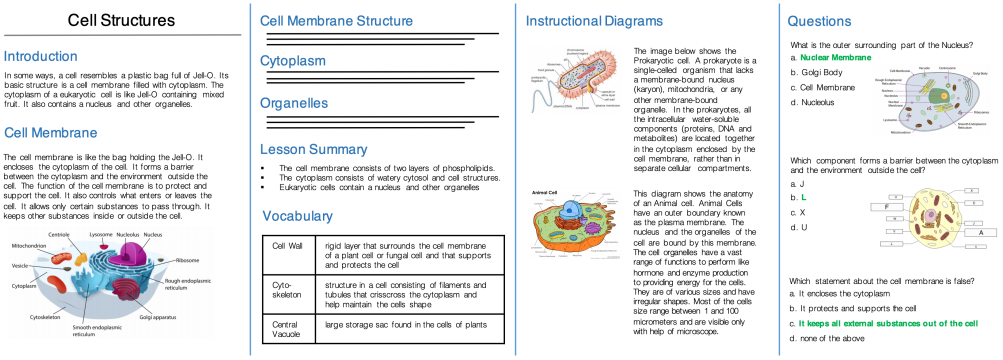


Figure 4.1: Examples of the textbook question answering task. In this figure, we can see lessons which contain long essays and diagrams in the TQA [42]. Related questions are also illustrated.

anchor words) in Textbook QA. We eliminate unnecessary parts of the context using the dependency links from the anchor words and we have the advantage of not having to read all the context to solve the problem. Our method is more efficient than the previous works that read all the context and also brings better performance focusing on the more relevant parts of the context.

Second, for the Video QA, we utilize an auxiliary temporal localization loss to focus on a more relevant part of the context. And we also present a locally aligned attention mechanism to selectively focus on corresponding video sequences of given subtitles. The previous works, however, have utilized attention mechanisms on video sequences and subtitles with either question and answer pairs respectively or with the subtitles in a globally aligned manner.

Finally, our approaches narrowing down the scope and focusing on more relevant parts achieve competitive performance in Textbook QA and Video QA such as TVQA, TVQA+, and DramaQA respectively.

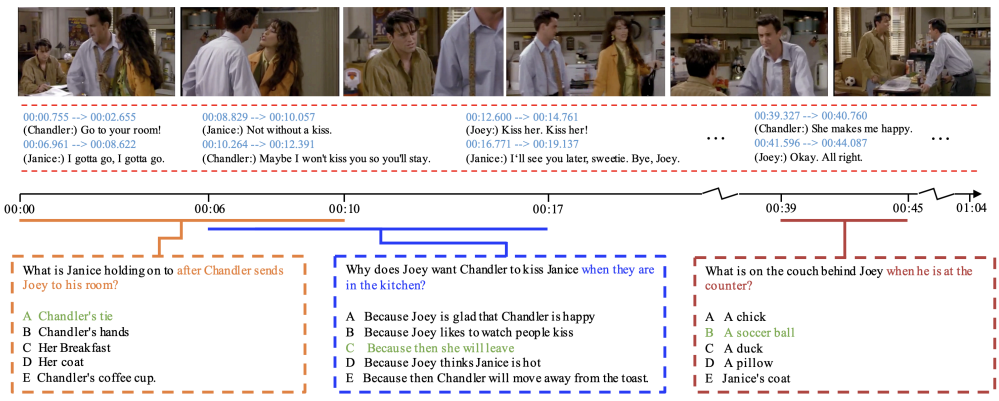


Figure 4.2: Examples of the TVQA dataset. All questions and answers are attached to 60-90 seconds long clips. There are questions requiring subtitles or videos alone to answer, while some require information of both modalities [55].. Related questions are also illustrated.

4.1 Related Work

Context question answering, also known as machine reading comprehension, is a challenging task which requires a machine not only to comprehend natural language but also to reason how to answer the asked question correctly. Large amount of datasets such as MCTest [78], SQuAD [75] or MS Marco [66] have contributed significantly to the textual reasoning via deep learning approaches. MCTest [78] is a smaller-scale datasets focusing on common sense reasoning and SQuAD [75] is a QA dataset where the answer is a span in the paragraph. MS MARCO dataset [66] is composed of the query, web documents and crowd-sourced answers. These datasets, however, are restricted to a small set of textual content. In this study, we tackle Textbook QA, the practical middle school science problems across multiple modalities, by transforming long essays into customized graphs for solving the questions on a textbook.

We also deal with a multi-modal context as a Video QA, large-scale TV

shows-based question answering datasets, composed of a 60-90 second long video clip with corresponding subtitles. Visual and video question answering requires the fine-grained interplay of vision and language to understand multi-modal contents. In the last few years, most of the pioneering works used a single image as a visual content with a joint image-question embedding and a spatial attention to predict the correct answer [2, 99, 104, 24]. More recently, beyond question answering on a single image, as video has become an important source of information, video QA has emerged as a key topic in the vision-language community [55, 56, 48, 54, 46, 49, 25]. In this work, we focus on learning the multi-modal representations by utilizing the transcripted subtitles as a context in order to solve the given questions.

4.2 Method

In this section, we describe our context matching approaches for two challenging multiple-choice question answering datasets, Textbook QA and Video QA.

4.2.1 Textbook QA

Figure 4.3 shows our entire framework. First, we retrieve one paragraph which is the most relevant to the given question or answer options by TF-IDF scores to narrow down the scope. Since 80% of the questions can be answered with only one paragraph according to the [42], we only concentrated on those questions. There might be a corresponding diagram for each paragraph. We convert the paragraph and the corresponding diagram into two types of context graphs for text and diagram, respectively. We incorporate Graph Convolutional Network (GCN) to extract graph features from both the visual and the textual context

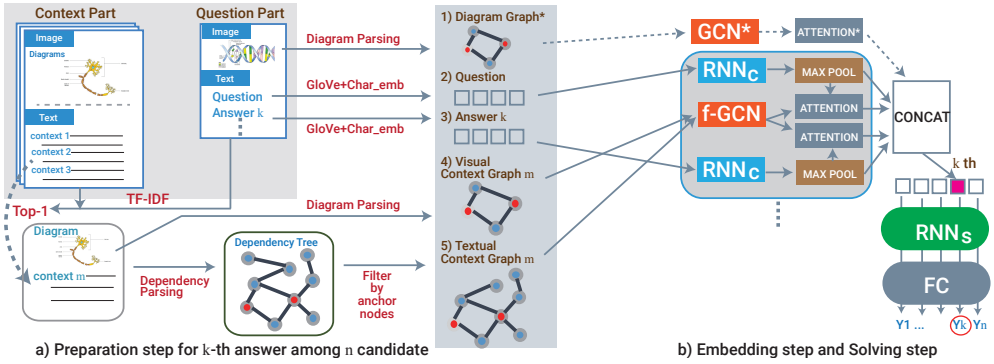


Figure 4.3: Overall framework of our model: (a) The preparation step for the k-th answer among n answer options. The context m is determined by TF-IDF score with the question and the k-th answer. Then, the context m is converted to a context graph m. The question and the k-th answer are also embedded by GloVe and character embedding. This step is repeated for n options. (b) The embedding step uses RNN_C as a sequence embedding module and f-GCN as a graph embedding module. With attention methods, we can obtain combined features. After concatenation, RNN_S and the fully connected module predict final distribution in the solving step.

graphs. Then, we encode textual inputs, a question and an answer option, utilizing an RNN (denoted as RNN_C in the figure). For the sake of our text embeddings, we use GloVe [70] vectors. After repeating the previous process for each answer option, we stack each of the concatenated representations. And, we exploit another RNN (RNN_S) to cope with the variable number of answer options which varies from 2 to 7 that can have sequential relations such as "none of the above" or "all of the above" in the last answer option. We predict the final answer with fully connected layers by deciding the probabilities of the answer options.

Context Graph Building

For the diagrams, we build a visual context graph using UDPnet [45]. We obtain the name of the entities with OCR information and the number of the entities as our node of the graph. Then we can establish edges between related entities.

We build the textual context graphs using relevant parts of the lesson where the questions can focus on solving problems as follows. Each lesson can be divided into multiple paragraphs and we extract one paragraph which has the highest TF-IDF matching score using a concatenation of the question and one of the candidate answers (leftmost of Figure 4.3(a)).

Then, we build the dependency trees regarding each sentence of the retrieved paragraph utilizing the Stanford dependency parser [62], and designate the words which are matched with the question and the answer option as anchor nodes. The nodes which have more than two levels of depth difference with anchor nodes are removed and we build the final textual context graphs using the remaining nodes and edges as shown in Process 1. We construct the adjacency matrix $\mathcal{A}$ using the remaining nodes and edges.

Graph Understanding

Next, we combine the visual and texual context graphs, f-GCN, as shown in Figure 4.4. Each of context graphs has its own graph matrix C containing node representations and a normalized adjacency matrix which are used as inputs of each GCN to comprehend the contexts. The graph matrix C is comprised of the word embeddings and the character representation. First, we extract graph features after the forward pass from both of the context graphs based on one-

Process 1 Build textual context and adjacency matrices C, $\mathcal{A}$

Input: a paragraph, a set of *anchor nodes* V

Construct a dependency tree on each sentence of the given paragraph
Split the tree into multiple units each of which represents two nodes and one edge $u = \{v_1, v_2\}$
$U \leftarrow$ a set of units
$E \leftarrow$ an empty set of edges
for $depth \leftarrow 1$ to 2 **do**
    for all nodes $v \in V$ **do**
        for all units $u \in U$ **do**
            if $v \in u$ **then**
                $E \leftarrow E \cup \{u\}$
            end if
        end for
    end for
    $V \leftarrow$ a set of all nodes in E
end for

Output: context matrix C from V with embedding matrices, adjacency matrix $\mathcal{A}$ from E

layer GCN as

$$\begin{aligned} H_c^t &= f(C^t, \mathcal{A}^t) = \sigma(\mathcal{A}^t C^t W^t) \\ H_c^d &= f(C^d, \mathcal{A}^d) = \sigma(\mathcal{A}^d C^d W^d), \end{aligned} \tag{4.1}$$

where $\mathcal{A}^t$ and $\mathcal{A}^d$ are the adjacency matrices for the text and visual contexts, W^t and W^d are learning parameters of linear layer for the text and visual contexts, and σ is the $\tanh$ activation function.

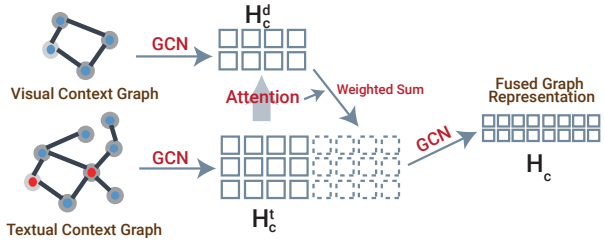


Figure 4.4: Illustration of f-GCN. Both textual and visual contexts are converted into H_c^d and H_c^t. We concatenate H_c^t and H_c^d to obtain combined features (f-GCN1). Finally, we use another GCN to get fused graph representation as f-GCN2.

We calculate the attention matrix Z of visual context H_c^d against textual context H_c^t as a main context with a dot product. Then we concatenate features of textual context H_c^t and weighted sum $Z^T H_c^d$ to get fused context features,

$$H_c^1 = [H_c^t; Z^T H_c^d], \tag{4.2}$$

where $[\cdot\ ;\ \cdot]$ is the concatenation operator. In order to fuse them again, We use another GCN to propagate over entire features of context graphs:

$$H_c^2 = f(H_c^1, \mathcal{A}^t) = \sigma(\mathcal{A}^t H_c^1 W^c). \tag{4.3}$$

We denote this module except the last GCN as f-GCN1 (eq. (4.2)) and the whole module including the last GCN as f-GCN2 (eq. (4.3)).

Multi-modal Problem Solving

We used the f-GCN and RNNs to embed the contexts and to answer the questions as shown in Figure 4.3(b). Two different RNNs are used in our architecture. One is the *comprehending* RNN (RNN_C) which understands questions

and answer options and the other is the *solving* RNN (RNN$_S$) which answers the questions.

The input of the RNN$_C$ is composed of the word embedding, character embedding, and the occurrence flag for both questions and candidate answers. In word embedding, each word can be represented as e_{q_i}/e_{a_i} by using a pre-trained word embedding method such as GloVe [70]. The character representation c_{q_i}/c_{a_i} is calculated by feeding randomly initialized character embeddings into a CNN with the max-pooling operation. The occurrence flag f_{q_i}/f_{a_i} indicates whether the word exists in the contexts or not. Our final input representation q_i^w for the question word q_i in RNN$_C$ is composed of three components as follows:

$$\begin{aligned} e_{q_i} =& Emb(q_i), \quad c_{q_i} = \textit{Char-CNN}(q_i) \\ & q_i^w = [e_{q_i}; c_{q_i}; f_{q_i}]. \end{aligned} \tag{4.4}$$

The input representations for the answer options are also obtained in the same way as the one for the question. Here, Emb is the trainable word embeddings and *Char-CNN* is the character-level convolutional network. For each representations of the questions and answer options, we apply the step-wise max-pooling operation over the RNN$_C$ hidden features.

Given each representation of the question and the answer option, we use an attention mechanism to acquire the attentive information Att_q which is calculated as follows:

$$\begin{aligned} Att_q = \sum_{k=1}^{K} \alpha_k H_{c_k}, \quad \alpha_k = \frac{exp(g_k)}{\sum_{i=1}^{K} exp(g_i)}, \\ g_k = h_q^T \mathbf{M} H_{c_k}. \end{aligned} \tag{4.5}$$

Here, K is the number of words in the context C which equals the dimension of the square adjacency matrix $\mathcal{A}$. $\mathbf{M}$ is the matrix that converts the question into

the context space. The attentive information of the candidate answers Att_a is calculated similar to Att_q.

RNN$_S$ can solve the problems and its input consists of the representations of the question and the answer option with their attentive information on the contexts as:

$$\begin{aligned} I^t_{RNN_S} &= [h_q; h_a; Att^c_q; Att^c_a], \\ I^d_{RNN_S} &= [h_q; h_a; Att^c_q; Att^c_a; Att^{qd}_q; Att^{qd}_a] \end{aligned} \tag{4.6}$$

where $I^t_{RNN_S}$ is for the text questions and $I^d_{RNN_S}$ is for the diagram questions. Finally, based on the outputs of RNN$_S$, we use one fully-connected layer followed by a softmax function to obtain a probability distribution of each answer option and optimize those with cross-entropy loss.

4.2.2 Video QA

In this section, we describe our architecture for multiple-choice video QA which has long video sequence with subtitles. For our problem setting, the inputs are composed of the following: (1) question q, (2) answer options $\Omega_a = \{a_n | n = 1, ..., N\}$, (3) subtitle sentences $\{S_t | t = 1, ..., T\}$ as a text context, and (4) video frames $\{V^i_t | t = 1, ..., T, i = 1, ..., I\}$ as a visual context where a_n is the n^{th} answer option, S_t is the t^{th} subtitle sentence, and V^i_t is the i^{th} image frame in the t^{th} video segment connected to the t^{th} subtitle sentence. Our goal is to predict the correct answer given a question and text/visual contexts.

$$\hat{a} = \underset{a \in \Omega_a}{\operatorname{argmax}}\, p(a | q, S, V; \theta) \tag{4.7}$$

Visual Representation

We first separate each video into T segments using the provided subtitle timestamp in the dataset, and further separate each segment into I image frames. Then, for the visual representation, we use ResNet-101 [35] trained on ImageNet [19] to extract global image features $v_t^i \in \mathbb{R}^{2048}$ as the i^{th} image feature in the t^{th} video segment. In addition, using Faster R-CNN [77] trained on Visual Genome [52], we extract objects $o_t^{i_j}$ as j^{th} object in the i^{th} image frame, which can be used as one of the text inputs described in the next subsection.

Text Representation

We use four types of text inputs: a question, answer options, subtitle sentences, and objects. For the objects o_t, extracted from each image frame in the t^{th} video segment, we use the following as the objects input:

$$o_t = [o_t^{1_1}; ...; o_t^{1_{J_1}}; ...; o_t^{I_1}; ...; o_t^{I_{J_I}}] \tag{4.8}$$

where $[\cdot;\cdot]$ is the concatenation operator. To encode entire textual inputs, we use BERT [20] which achieves state-of-the-art performance on a wide range of NLP tasks. While only one or two types of text inputs are used with [SEP] tokens in the standard practices of BERT, since we use four different types of text inputs, we separate them with [SEP] tokens as follows:

$$[\texttt{CLS}]\ q\ [\texttt{SEP}]\ a_n\ [\texttt{SEP}]\ S_t\ [\texttt{SEP}]\ o_t\ [\texttt{SEP}].$$

To properly distinguish multiple text inputs (four in our case) in the model, we modify the token type embedding method to explicitly accommodate different token type embeddings as types of text inputs vary. For the first input, we keep

Conceptual text input
[CLS] *question* [SEP] *answer option* [SEP] *subtitle sentence* [SEP] *objects* [SEP]
Original text input
[CLS] *Where does Ted go after leaving the bar ?* [SEP] *Ted goes to Marshall's apartment to tell him about the trip* [SEP] *Marshall : In fact, take my car .* [SEP] *necklace brown shirt woman ...* [SEP]

Table 4.1: The text input of BERT. We use four types of text input as a question, answer option, subtitle sentence, and the objects as shown in the conceptual text input. The example of the text input is shown in the original text input.

the token type embedding of 0 as it is. For the second and third inputs, we use the output of the token type embedding of 1 but multiplied by $\frac{1}{3}$ and $\frac{2}{3}$, respectively, to distinguish them. Lastly, we keep the token type embedding of 1 for the fourth text input.

Network for Multiple-choice Video QA

For the video QA, in addition to predicting the answer as our main task, we make use of timestamp annotation of localized span needed to answer the question given in the dataset and add temporal localization learning as an auxiliary task. We use visual and text inputs for our video QA network as shown in Fig. 5.2(a). For the visual representation $H_v \in \mathbb{R}^{T \times I \times d_v}$, we extract $d_v = 2048$ features of the last block of ResNet-101 which was used in Lei *et al*. [55]. We set the number of images I as 4, extracted from the video segment connected to each subtitle sentence. In our implementation, we repeated H_v N times to match

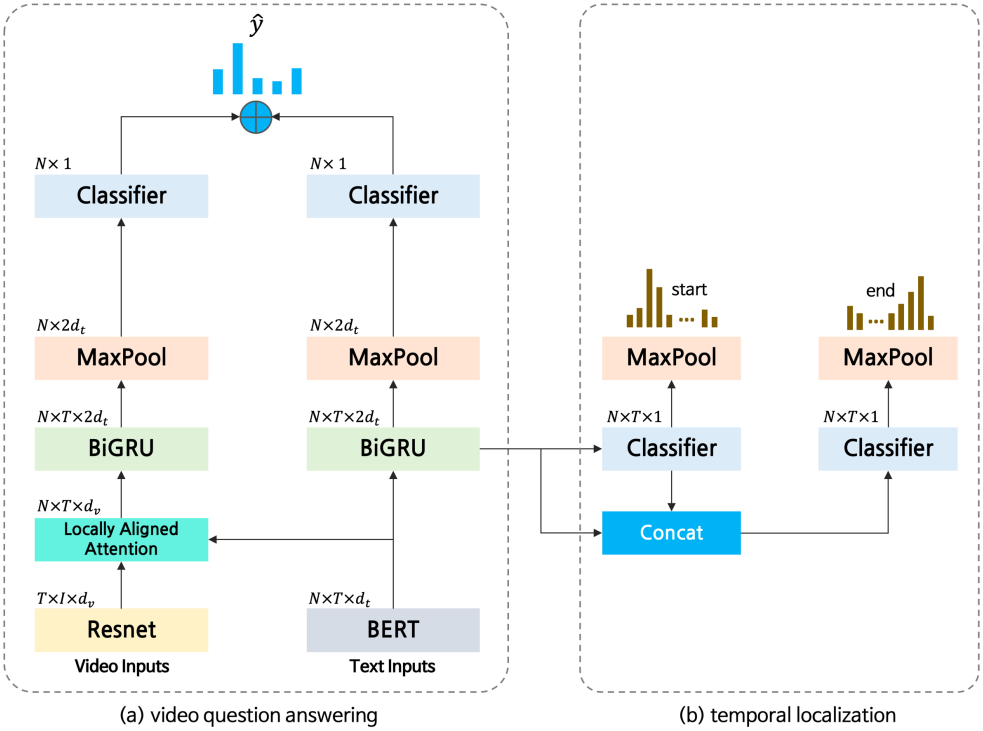


Figure 4.5: Overall architecture of our model: (a) For a video QA part, we use ResNet and BERT to extract video and text representations. A locally aligned attention mechanism is introduced to match each subtitle sentence with the corresponding images. Then, we use RNNs to learn sequential information of subtitle sentences. We predict the final answer distribution on both modalities. At inference time, we use this video QA part only. (b) Temporal localization, one of our auxiliary tasks, is used to predict the necessary part to answer the question.

the dimension with the text representation. For the text representation $H_t \in \mathbb{R}^{N\times T\times d_t}$, we extract $d_t = 768$ features of the hidden state of the [CLS] token from the last layer of 12-layer BERT-base model.

To extract attentive information between a text context from a subtitle and a visual context from the corresponding video frames, we calculate the locally aligned attention to focus on particularly relevant images regarding each subtitle

sentence. This prevents the model from reasoning with unnecessary information. Our locally aligned attention mechanism, used only in the image side, is calculated between image frames and the subtitle sentence that share the timestamp with the image frames.

$$H_v^{Att} = \sum_{i=1}^{I} \alpha_i H_{v_i}^T \mathbf{M},\ \alpha_i = \frac{e^{g_i}}{\sum_{k=1}^{I} e^{g_k}},\ g_i = H_{v_i}^T \mathbf{M} H_t. \tag{4.9}$$

Here, I is the number of image frames from the video segment matching to each subtitle sentence by the timestamp information given in the dataset, and $\mathbf{M}$ is the projection matrix that converts the text representation into the visual representation space.

To reflect the sequence information between multiple subtitle sentences, we use BiGRU on both text and video respectively. Then, we apply the max-pooling operation across the sequence of the subtitle sentences, to get a global representation of each answer, called hypothesis:

$$\begin{aligned} \mathcal{H}_v^{Att} &= \text{Max}(\text{BiGRU}(H_v^{Att})), \\ \mathcal{H}_t &= \text{Max}(\text{BiGRU}(H_t)). \end{aligned} \tag{4.10}$$

Given the max-pooled hypothesis representations, we use two fully-connected layers as classifiers to obtain the logits s_t and s_v for the answer options on both sides of text and video respectively.

$$s_v = \text{classifier}(\mathcal{H}_v^{Att}), \quad s_t = \text{classifier}(\mathcal{H}_t). \tag{4.11}$$

Then, we add those logits followed by a softmax function to obtain a probability distribution of each answer option and apply cross-entropy loss as our question answering loss:

$$\hat{y} = \text{softmax}(s_v + s_t), \quad \mathcal{L}_{qa} = -\sum_{i=1}^{N} y_i \log \hat{y}_i. \tag{4.12}$$

Temporal Localization As a Subtitle Context Matching

We use temporal localization network as a context matching, shown in Figure 5.2(b), which localizes relevant moments from a long video sequence given a question, and assign the ground truth start/end sentence position in the subtitle sequence using the given start/end time annotations. We utilize the BiGRU output $\mathcal{H}_t$ from the text input, reflecting the sequence information of the text context and a question. Then, we predict the start/end position using span predicting classifiers calculating the temporal matching scores followed by a max-pooling operation across the five hypotheses, and train them with cross-entropy loss as follows:

$$\mathcal{L}_{span} = -\frac{1}{2}(\log p_{start} + \log p_{end}) \tag{4.13}$$

where p_{start} and p_{end} are the span probabilities of the start and end ground truth positions respectively. Since we use this temporal localization part as one of our auxiliary tasks, we do not need start/end time annotations as well as temporal localization network in the inference time.

We can define the total loss using a linear combination of the previous two losses with scale parameters; λ_{qa} and λ_{span} as follows:

$$\mathcal{L} = \lambda_{qa} * \mathcal{L}_{qa} + \lambda_{span} * \mathcal{L}_{span} \tag{4.14}$$

4.3 Experiment

We evaluate our approaches using context matching on four benchmark datasets: Textbook QA and Video QA as TVQA, TVQA+, and DramaQA.

4.3.1 Datasets

Textbook QA

TQA is a textbook based question answering dataset which consists of 1,076 lessons from Life Science, Earth Science and Physical Science textbooks. While the dataset contains 78,338 sentences and 3,455 images including diagrams, it also has 26,260 questions with 12,567 of them having an accompanying diagram, split into training, validation and test at a lesson level. The training set consists of 666 lessons and 15,154 questions, the validation set consists of 200 lessons and 5,309 questions and the test set consists of 210 lessons and 5,797 questions. Since evaluation for test is hidden, we only use the validation set to evaluate our methods.

Video QA

TVQA is a large scale multiple-choice video QA dataset based on 6 popular TV shows: *The Big Bang Theory, How I Met Your Mother, Friends, Grey's Anatomy, House, Castle*, and consists of 152,545 QA pairs from 21,793 clips, spanning over 460 hours of video. The training, validation, and test-public set consist of 122,039, 15,253, and 7,623 questions, respectively.

TVQA+ is a subset (*The Big Bang Theory*) of TVQA, but TVQA+ adds frame-level bounding box annotations for visual concept words and modifies its timestamp information for better annotations. TVQA+ contains 29,383 QA pairs from 4,198 video clips, with 148,468 images annotated with 310,826 bounding boxes. The training, validation, and test-public set consist of 23,545, 3,017, and 2,821 questions, respectively. Note that we do not use bounding box information on TVQA+ to match the problem format to that of TVQA.

DramaQA is built upon the TV drama (*Another Miss Oh*) and it contains 16,191 QA pairs from 23,928 various length video clips. The QA pairs belong to one of four difficulty levels and the dataset provides the character-centered annotations, including visual bounding boxes, behaviors, and emotions of main characters. As in TVQA+, we do not use bounding box information at all. However, we use textual information regarding behaviors and emotions as objects. The number of examples for training, validation, and test datasets is 10,098, 3,071, and 3,022, respectively.

4.3.2 Implementation Details

Textbook QA

We initialized word embedding with 300d GloVe vectors pre-trained from the 840B Common Crawl corpus, while the word embeddings for the out-of-vocabulary words were initialized randomly. We also randomly initialized character embedding with a 16d vector and extracted 32d character representation with a 1D convolutional network. And the 1D convolution kernel size is 5. We used 200 hidden units of Bi-LSTM for the RNN_c whose weights are shared between the question and the candidate answers. The maximum sequence length of them is set to 30. Likewise, the number of hidden units of the RNN_s is the same as the RNN_c and the maximum sequence length is 7 which is the same as the number of the maximum candidate answers. We employed 200d one layer GCN for all types of graphs, and the number of maximum nodes is 75 for the textual context graph, 35 for the diagrammatic context graph, and 25 for the diagrammatic question graph, respectively. We use $\tanh$ for the activation function of the GCN. The dropout was applied after all of the word embeddings with a

keep rate of 0.5. The Adam optimizer with an initial learning rate of 0.001 was applied, and the learning rate was decreased by a factor of 0.9 after each epoch.

Video QA

We use pre-extracted 2048-dimensional hidden features (d_v in Fig. 5.2) from the Imagenet-pretrained ResNet-101 and object information from the modified Faster R-CNN trained on Visual Genome [55]. We use the BERT-base uncased model, which has 12 layers with hidden size of 768 and fine-tuned only top-6 layers due to the limitation of resources. We set the hidden sizes of all the remaining layers as 768 (d_t in Fig. 5.2). The total video context sequence T is 40, the number of images I corresponding to each subtitle sentence is set to 4, and the number of answer options N is 5 in all datasets, as shown in Figure **??**. The maximum number of tokens of the text input is set to 80 for TVQA/TVQA+ and 170 for DramaQA. The probability of masking out the tokens used in our contrastive learning is 0.2. The weights of each loss λ_{qa}, λ_{span}, and λ_{cont} are set to 1, 0.2, and 0.1 based on TVQA+ validation performance. We set the learning rate to 1e-5 for the self-supervised pre-training stage and 5e-5 for the main QA stage. Likewise, the total number of epochs is set to 1 for the pre-training stage and 3 for the main QA stage. We use the batch size of 8 for the entire experiment settings.

4.3.3 Experimental Results: Textbook QA

Overall Performance

Overall performances on the Textbook QA dataset are shown in Table 4.2 including the previous models. As the ones of the baselines, MemN+VQA and

MemN+DPG utilize memory networks to embed text data such as the lessons and questions. MemN+VQA uses VQA approaches for diagram questions, while MemN+DPG exploits Diagram Parse Graph (DPG) as a context graph on diagrams built by DsDP-net [41]. BiDAF+DPG incorporates BiDAF (Bi-directional Attention Flow Network) [82], one of the machine reading comprehension models, for the textual data. It exploits bidirectional attention to capture dependencies between the question and the corresponding context paragraph. Challenge denotes the top result in the TQA competition [42] and IGMN uses the Instructor Guidance with Memory Networks (IGMN) based on Contradiction Entity Relationship Graph (CERG).

The performances of our models are at the bottom of the Table 4.2. The result shows that our model outperforms previous baseline models in all types of questions. Our model shows about 2% higher than the previous best model, IGMN, in the accuracy of all questions. We believe that our context graph understanding method works well on this TQA problem since our models achieve significant margins compared to the previous researches. IGMN also exploits a graph module of contraction, but ours outperforms especially in both text problems, T/F and MC with over 2.5% and 6.5% margin, respectively. We believe that our graph building strategy can represent the feature of context for problem solving and the GCN also plays an important role in encoding the features of our graph.

Ablation Study

First, we replace our GCN model with a LSTM model for processing the context (w/o f-GCN in Table 4.2). We can see that the accuracy of the model without GCN decreased over 1.5% compared with the proposed GCN model. It tells that

Model	Text T/F	Text MC	Text All	Diagram	All
Random	50.10	22.88	33.62	24.96	29.08
MemN+VQA [42]	50.50	31.05	38.73	31.82	35.11
MemN+DPG [42]	50.50	30.98	38.69	32.83	35.62
BiDAF+DPG [42]	50.40	30.46	38.33	32.72	35.39
Challenge	-	-	45.57	35.85	40.48
IGMN [57]	57.41	40.00	46.88	36.35	41.36
Our model	60.02	46.86	52.06	36.61	43.97
w/o f-GCN	58.72	45.16	50.51	35.67	42.74

Table 4.2: Comparison of performance with previous methods (Top). We describe the accuracies of each type of questions, Text T/F (true-false in text only), Text MC (multiple-choices in text only), Text all (all in text only), Diagram and All.

our proposed approach, which concentrates on the co-exist keyword between the context and the question/answer, is effective than the RNN dealing with the whole paragraph. Furthermore, the inference time of our proposed model is much lower than the RNN model since our proposed graph structure does not require all words in the context. Thus, the context graph we built for each lesson could give proper representations with the f-GCN module.

Second, Table 4.3 shows the results of the ablation study about occurrence flag representing our keyword matching strategy. In Eq. (4.4), we concatenate three components including the occurrence flag to create question or answer representation. We found that the occurrence flag which explicitly indicates the existence of a corresponding word in the contexts is considerably effective. Results of all types degrade significantly as ablating occurrence flags. Especially, eliminating a-flag (answer flag) drops accuracy about 7% which is almost 4 times higher than the decrease due to eliminating q-flag (question flag). We be-

Model	Text	Diagram	All
Our model	**52.06**	**36.61**	**43.97**
w/o q-flag	49.29	35.78	42.21
w/o a-flag	43.24	31.50	37.09
w/o q & a-flag	42.64	31.72	36.92

Table 4.3: Results of ablation study about the occurrence flags. We demonstrate the accuracies of Text only, Diagram, and total questions.

lieve that disentangled features of answer candidates can mainly determine the results while a question feature equally affects all features of candidates. And, in this dataset containing many technical keywords like Textbook QA, simple but effective keyword matching has also a significant role to solve the problem. Our model without both flags shows the lowest results due to the loss of representational power.

Qualitative Results

Figure 4.6 shows the qualitative results of the three text-type questions. We can see the textual context, questions, answer options, and the constructed graph from the sentences of the context. The gray and blue nodes in the graph are anchor nodes that are designated by the keyword matching between the question/answers and the context sentences. Green rectangles represent the relation types in the dependency graph.

The first example is a boolean question that the answer options are True or False. The question is “convection currents occur in the inner core”, and three words, “currents”, “core” and “convection” are set as anchor nodes as shown in the first example of Figure 4.6. We can find the “outer” node from the graph

which has the opposite meaning with the "inner" word in the question sentence. As a result, our model predicts the true and false probabilities of this question as 0.464 and 0.536, respectively, and correctly solves this problem as a false statement. The second example is a multiple choice question (4 options in this case) which is more complicated than the T/F problem. In the neighbor with the anchor nodes of the question such as "causes", "erosion" and "soil", we can easily find the "running" and "water" nodes which exist in the answer option (d) among the 4 options. Therefore, our model can estimate (d) as the correct answer with the highest probability of 0.455. The last example shows a more complicated multiple choice question (7 options in this case). In the context graph, we set "organelle", "recycles", "molecules" and "unneeded" as question anchor nodes with each anchor word of the answer options. Then we can easily find an important keyword, "lysosome" of the answer option (a) in the neighbors of the anchor nodes of the question.

These examples demonstrate abstraction ability and relationship expressiveness which can be huge advantages of graphs. Moreover, those results could support that our keyword matching strategy in building context graph works well for the context understanding in solving textbook question answering.

4.3.4 Experimental Results: Video QA

TVQA

We evaluate our model on TVQA dataset as shown in Table 4.4. Since the ground truth answers of the test set are not provided, we present the performance via the online evaluation server system. Our model achieves 76.15% of accuracy on the test set, outperforming the previous state-of-the-art models, MSAN [49]

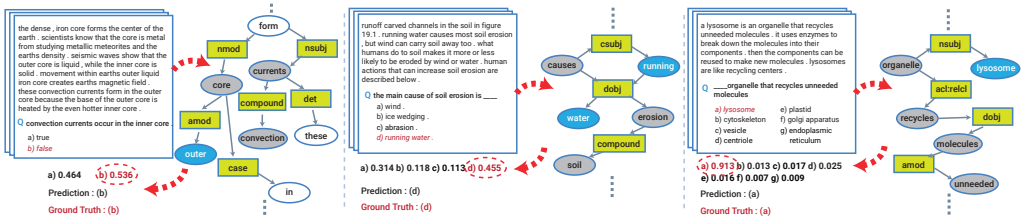


Figure 4.6: Qualitative results of text-type questions without visual context. Each example shows all items for a question in the textbook and a textual context subgraph to solve a question. And our predicted distribution for answers and ground truths are also displayed. In the subgraph, gray circles represent words in questions and blue circles represent words related to answers. Green rectangles represent relation types of the dependency graph.

using modality importance with BERT and DenseCap [46] using captions and frame-selection with RoBERTa, with over 5% and 2% margins, respectively. Our model also achieves the best performance in all 6 individual TV shows. Note that, to improve the model performance we utilized the proposed training strategy which will be introduced in the next chapter, and the performance is already reflected in Table 4.4.

TVQA+

Table 4.5 shows the performance on TVQA+ dataset. To measure the performance of our model, we use QA classification accuracy just like in TVQA, and additionally, temporal mean Intersection-over-Union (mIOU) and Answer-Span joint Accuracy (ASA) provided by Lei *et al.* [56] are used. mIOU measures temporal localization and ASA jointly evaluates the performance of both QA classification and temporal localization. For the ASA metric, we regard a prediction to be correct if the predicted temporal localized span has an IoU ≥ 0.5

Models	Val (Acc.)	Test-public (Acc.)						
	All	bbt	friends	himym	grey	house	castle	All
multi-stream [55]	65.85	70.25	65.78	64.02	67.20	66.84	63.96	66.46
PAMN [48]	66.38	-	-	-	-	-	-	66.77
Multi-task [47]	66.22	-	-	-	-	-	-	67.05
CA-RN [25]	68.90	71.43	65.78	67.20	70.62	69.10	69.14	68.77
STAGE [56]	70.50	-	-	-	-	-	-	70.23
akalsdnr (anonymous)	71.13	71.49	67.43	72.22	70.42	70.83	72.30	70.52
MSAN [49]	70.79	-	-	-	-	-	-	71.13
DenseCap [46]	74.20	74.04	73.03	74.34	73.44	74.68	74.86	74.09
Ours	**76.23**	**77.43**	**73.24**	**76.72**	**74.04**	**76.94**	**77.86**	**76.15**

Table 4.4: Comparison of QA performance with previous methods on TVQA validation and test sets. All results are from the models that do not use timestamp annotations (w/o ts version). We also compare the performance on the 6 individual TV shows.

with the ground-truth span and the answer is correctly predicted. We obtained the accuracy of 76.21% in QA classification and the mIoU of 39.03% in temporal localization. For ASA, we achieved 31.05%, outperforming the previous state-of-the-art model, STAGE [56] which used BERT with grounding spatial regions and temporal moments, with about 9% margin.

DramaQA

Table 4.6 shows our result on DramaQA dataset, consisting of four levels of difficulty. The first five lines of Table 4.6 show the top-5 resulting models of the DramaQA challenge, evaluated on the test set. Since the challenge is no longer ongoing and the test set is yet inaccessible, we evaluate our model only on the available validation set and report ours for future benchmark comparison.

Models	QA (Acc.)	TempLocal (mIOU)	ASA
ST-VQA [39]	48.2	-	-
two-stream [55]	68.13	-	-
STAGE (video) [56]	52.75	10.90	2.76
STAGE (sub) [56]	67.99	30.16	20.13
STAGE [56]	74.83	32.49	22.23
Ours	**76.21**	**39.03**	**31.05**

Table 4.5: Comparison on TVQA+ test set. We evaluate QA accuracy, mIoU for temporal localization, and Answer-Span joint Accuracy (ASA) as the overall performance indicators.

Models	Difficulty 1	Difficulty 2	Difficulty 3	Difficulty 4	Overall
IITDrama	76	72	55	60	71
bjorn	77	74	57	57	71
HARD KAERI	76	73	56	59	71
Sudoku	78	74	68	67	75
GGANG	81	79	64	70	77
Ours (validation)	84	85	70	70	81

Table 4.6: QA accuracy on DramaQA dataset with four difficulty levels. Task becomes more difficult as the level increases. We report top-5 results from the competition leaderboard, evaluated on the test set. Note that, we only evaluate on the validation set since the challenge is no longer ongoing and the test set is yet inaccessible.

Although direct comparison is difficult, our model shows competitive performances among others.

Ablation study

We conduct an ablation study on the TVQA+ validation set as shown in Table 4.7. For an ablation experiment, we define base models where the token type embedding and temporal localization. The base models consist of the globally aligned attention model (1) and the proposed locally aligned attention model (4). First, we observe that the models with locally aligned attention outperformed all the other models that are trained with a globally aligned attention in (1-3 vs. 4-6). It implies that misalignment between subtitle sentences and the image frames from other sentences can be prevented by utilizing the time sequence information. Second, (2,5) shows the effectiveness of the temporal context matching. We can see that predicting the relevant subtitle sentences helps answer the question correctly, and to calculate the relevance, sentence matching against the subtitle/question/answer option is required. Lastly, the multiple token type embedding technique (3,6) improves the performance and it can be extensively applied when working with various types of text inputs.

4.4 Summary

In this chapter, we introduce the multiple-choice question answering models with the aid of context matching. First, when it comes to Textbook QA, we propose the building strategy for the textual context graph. We split the context into multiple sentences and convert each sentence into the dependency parse trees using a dependency parser. Then we designated the words of the context as an-

Models	QA (Acc.)
(1) base model (GA)	71.62 ± 0.45
(2) + TL	73.45 ± 0.31
(3) + TL + MT	73.98 ± 0.27
(4) base model (LA)	72.29 ± 0.31
(5) + TL	73.53 ± 0.29
(6) + TL + MT	74.54 ± 0.21

Table 4.7: Results of the ablation study of our model on TVQA+ validation set. We ablate our model with globally aligned attention (GA), locally aligned attention (LA), multiple token type embeddings (MT), and Temporal localization span loss (TL).

chor nodes if the words exist in the question or the answer option by a keyword matching. We remove the nodes which have more than two levels of depth difference with anchor nodes to narrow down the scope. Since the Textbook QA dataset deals with a number of technical terminologies, context matching by keywords works well and brings us a comparable performance.

Since we assumed that each question can be answered using only one relevant paragraph among multiple paragraphs, we relied on the paragraph retrieval to select our context paragraph. We retrieved the paragraph using the question and each answer option as our query for retrieval. However, we need to expand our context scope from using only one paragraph to the use of whole paragraphs since using one paragraph has two limitations. The first limitation is that TF-IDF based paragraph retrieval might retrieve the paragraph which is not relevant to the question and the answer options. In that case, the solver model could

not refer to the relevant part of the context so that it cannot predict the answer correctly. The second one is that there are some questions requiring multiple paragraphs to reason the answer such as multi-hop reasoning. In that case, it is impossible to answer the question using only one paragraph. As our future work, we expand our context-building strategy to multiple paragraphs and designate the anchor node from whole paragraphs using keyword-based matching. Furthermore, we can incorporate the sentence matching between each context sentence and the question/answer sentences with keyword matching for narrowing down the scope.

Second, for the TV show-based Video QA, we use pre-trained BERT as our contextual embeddings. Also, we use sentence-level matching to utilize the contextual cross information using each subtitle sentence, question, answer option, and the objects in the early fusion of our four textual inputs of BERT. To deal with multiple types of text inputs, we propose multiple token types embedding approach to expand the original two textual inputs of BERT to the multiple types (4 in our case) by an interpolation between 0 and 1 embeddings. Also, we borrow the temporal localization, calculating the relevant score at every subtitle sentence, as our auxiliary learning method. When it comes to composing multi-modal representations, we introduce a locally aligned attention method along with each subtitle sentence to selectively focus on corresponding video frames. We demonstrate that the temporal localization method as our auxiliary task, locally aligned attention mechanism, and the proposed multiple token type embeddings work well in our ablation study. Finally, We achieved competitive performance for the Video QA datasets such as TVQA, TVQA+, and DramaQA.

For the future work of Video QA, we can expand the temporal localization from using only textual inputs into incorporating both text inputs and visual

frames. Since there are multiple questions requiring understanding the visual inputs, we expect that the performance can be improved when we combine the textual and visual inputs to calculate the relevant part of the context as temporal localization.

Chapter 5

Training Schemes for Context-based Question Answering

We addressed the sentence matching model in chapter 3 and the multiple-choice question answering tasks with a context matching strategy in chapter 4, respectively. We, in this chapter, shift our focus to the training procedure that could possibly take the most advantage out of the given dataset.

5.1 Motivation

Most researches concentrated on solving the problems itself in multiple-choice question answering [57, 42, 46, 56, 49, 25]. However, there are more rooms to improve the performance beyond the training datasets, model architectures, or reasoning functions. On the other hand, some researches dealt with the training strategies such as transfer learning [14, 100], however, those approaches require additional external datasets and suffer from domain adaptation in transferring the knowledge from one domain to others. We propose the resource-efficient pre-training strategy which does not require any additional datasets but brings

the performance improvement enabling better parameter initialization.

Besides, in recent years, contrastive learning has led to a state-of-the-art performance in the unsupervised training of deep image models [96, 81, 8, 44, 29, 34, 22, 30] however there is a lack of research in the language domain. Even more, to our best knowledge, there has been no research applying contrastive learning against the answer options in multiple-choice question answering since it is hard to claim that the correct and the wrong answer options are types of the classes. Even though those answer options are not classes, we regard them as classes and push apart from each other between the correct and the wrong answer options by supervised contrastive learning to get a further performance improvement. Our proposed supervised contrastive learning contrasting the correct and the wrong answer options is, to our best knowledge, the first attempt applying contrastive learning into the multiple-choice question answering.

In this chapter, we propose two training schemes for multiple-choice video question answering in order to enhance the performance with 1) a self-supervised pre-training stage and 2) supervised contrastive learning in the main stage as auxiliary learning.

In the self-supervised pre-training stage, we transform the original problem format of predicting the correct answer into the one that predicts the relevant question to provide a model with broader contextual inputs without any further dataset or annotation. We originally used three types of input as follows: question, answer options, and the context as shown in Figure 5.1. For this example, the context is comprised of the subtitles and the video frames. We randomly select the negative questions from the given context for negative sampling using the metadata such as the TV show names, the episode, or season information. We set the questions from other contexts as negative questions. Hence, we can

come up with the one positive question which is the original question corresponding to the context, and multiple negative questions (4 in our case).

During this self-supervised pre-training stage, instead of predicting the correct answer, our model is expected to predict the positive question from the corresponding context which is comprised of the video clips and subtitles. And this training scheme lets the model learn a better weight initialization in advance before fine-tuning the model. This procedure does not require any additional data or human annotation.

For contrastive learning in the main fine-tuning stage, in addition to the main QA loss and the temporal localization loss which are introduced in chapter 4, we propose an additional contrastive loss that can be applied for the multiple-choice video QA tasks. Taking the ground-truth answer as a positive sample and the rest as negative samples, the contrastive loss confines the positive sample to be mapped in the neighborhood of an anchor, a perturbed ground truth answer, and the negative samples to be away from the anchor. For a perturbed ground truth answer, we add a masking noise to the input corresponding to the ground-truth answer. By mapping the positive sample closer to the masked input, we show that the model performance is improved.

We further show the effectiveness of the contrastive loss by investigating how the distance between the positive sample and negative samples changes as the training continues. Furthermore, we cluster the representations to see how well the positive representation and the four negative ones are clustered among themselves, respectively.

We evaluate our two proposed training schemes on the multiple-choice video question answering tasks: TVQA, TVQA+, and DramaQA utilizing the subtitles as a textual context with additional visual information like in Figure 5.1.

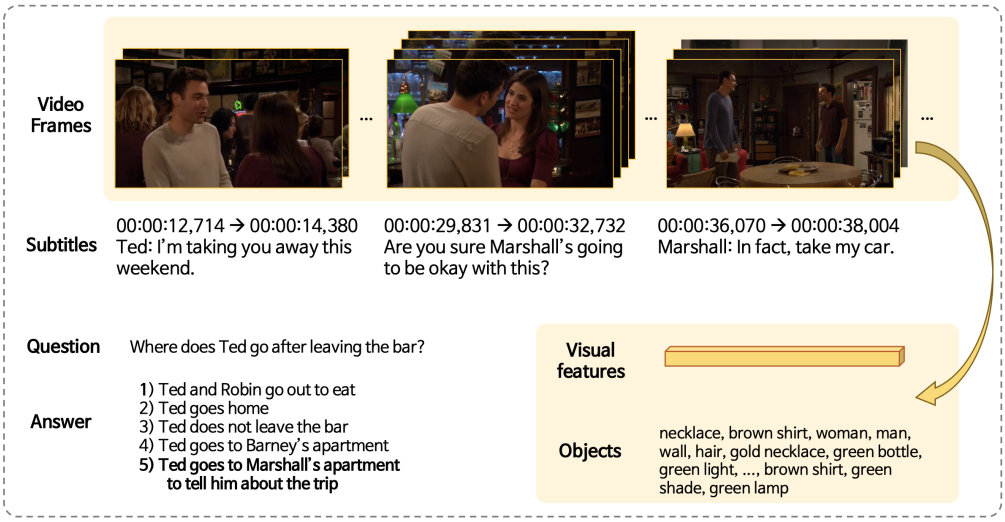


Figure 5.1: Multiple-choice Video QA example of TVQA dataset, composed of a 60-90 second long video clip, question, and the answer options. A video clip consists of video frames and subtitles, and each subtitle is connected to several frames. In our setting, we additionally extract object information and visual features from the video frames using Faster R-CNN and ResNet-101 as in the bottom right yellow box. We use question, answer, subtitles, and objects as our text input and visual features as our visual input.

Empirically, our model takes advantage of the supervised contrastive loss in the main stage and gives further improvements when self-supervised pre-training is preceded. Moreover, our model demonstrates significant performance increase on the test server, outperforming the state-of-the-art scores on all datasets.

5.2 Related Work

On top of utilizing large-scale pre-trained language model and fine-grained object detection results on videos, motivated by recent progress in contrastive

learning [44, 8] and unsupervised pre-training in natural language processing [25], we propose training schemes for multiple-choice video QA that integrates these two perspectives to increase performance gain.

5.2.1 Self-supervised Learning

Self-supervised pre-training is one format of unsupervised training that do not require an annotated labels. It captures the intrinsic information and patterns of the context such as a number of corpus or images. Modern techniques of self-supervised learning are pre-trained on large-scale external and unlabeled datasets [20, 61, 74, 15]. [20] is pre-train the model with a masked language model (MLM) objective, inspired by the Cloze task [89]. The masked language model randomly masks some of the input tokens, and the objective is predicting the original vocabulary of the masked word based on its context. Instead of masking the input, [15] replace to input tokens using small generator model, then the model is pre-trained by a discriminative model to predict whether each token in the corrupted input was replaced by a generator or not. Several studies [56, 103, 46, 49] have taken advantages of these self-supervised pre-trained models and combined with their video QA models to learn representations of the text data such as questions, answers, subtitles, and extracted visual concepts. Likewise, we utilize the pre-trained language model to embed textual information to solve video QA tasks. Besides, we propose a self-supervised learning approach for multiple-choice video QA of predicting a relevant question given contexts, which does not require any additional data or further annotations.

5.2.2 Contrastive Representation Learning

In this subsection, we describe contrastive representation learning [31] which has been explored in numerous literature as a method of extracting powerful feature representation. The main goal of the learning is to, as the name suggests, contrast the semantically nearby points against dissimilar points, in the embedding space.

Contrastive learning has shown success in the computer vision domain with a self-supervised learning algorithm. Contrastive Predictive Coding [68] learns representations by predicting future latent space using the powerful autoregressive model and the contrastive loss. MoCo [10, 34] maintains a memory bank of samples to support a larger dictionary size for calculating the contrastive loss. SimCLR [8, 9] introduces a learnable nonlinear transformation (also known as projection head) between the representation and the contrastive loss. In the language domain, Fang *et al.* [22] proposed Contrastive self-supervised Encoder Representations from Transformers (CERT), which pre-trains language representation models, to capture better sentence-level semantics using contrastive self-supervised learning at the sentence level with back-translation.

Meanwhile, some approaches have focused on leveraging labeled data into contrastive representation learning. [44] extends the self-supervised batch contrastive approach to the fully-supervised setting by pushing apart clusters of the samples from different classes and pulling together the samples belonging to the same class in the embedding space. Gunel *et al.* [30] proposed supervised contrastive learning (SCL) objective for the fine-tuning stage in the language domain, using the intuition that good generalization requires capturing the similarity between the samples in the same class and contrasting them in the different

classes.

In our setting, there are no classes in the multiple-choice video question answering, but it has a correct answer and negative answers. So, we contrast the representation of the correct answer and the representations of the wrong answers using given annotations in a supervised manner to separate the correct and wrong answers farther. For the anchor representation, we make a noise of the input text containing the correct answer by replacing the real token with the [MASK] token with a certain probability.

We calculate the matching scores between the noisy ground truth answer as an anchor representation and the five text representations each of which has a different answer option. We improve the model performance with a contrastive loss in an auxiliary setting on top of the main QA task by mapping the positive sample closer to the anchor.

5.3 Method

Our framework consists of two consecutive stages of training: first is the self-supervised pre-training stage and second is the training stage with a supervised contrastive learning loss in an auxiliary loss setting. During the self-supervised pre-training, instead of predicting the correct answer, our model is expected to predict the relevant question given contexts such as video clips and subtitles to learn a better weight initialization. This procedure does not require any additional data or human annotation. For the fine-tuning stage, in addition to the main QA loss, we propose a contrastive loss that can be applied for the multiple-choice QA tasks and we also make use of a temporal localization loss with text matching in chapter 4. Taking ground truth answer as a positive sample and the

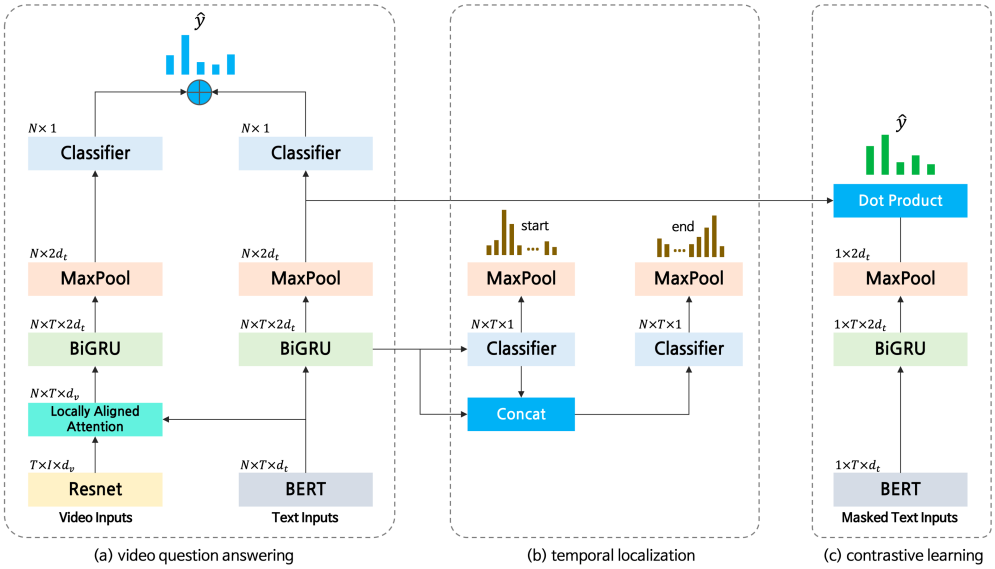


Figure 5.2: Overall architecture of our model: (a) and (b) are explained in chapter 4. (c) We introduce the contrastive loss, which is another component of our auxiliary tasks, to enhance the model's performance. We utilize the identical BERT and RNN, used in a video QA part with the masked text input of the ground-truth and predict the answer distribution by contrasting positive pair against negative pairs.

rest as negative samples, the contrastive loss confines the positive sample to be mapped in the neighborhood of an anchor, a perturbed ground truth answer, and the negative samples to be away from the anchor. We further show the effectiveness of contrastive loss by investigating how the distance between the positive sample and negative samples changes as the training continues.

Contrastive Learning

Figure 5.2 (c) shows a proposed contrastive learning approach, as another auxiliary task, that enhances the model performance in the multiple-choice video

Conceptual text input
[CLS] *question* [SEP] *answer option* [SEP] *subtitle sentence* [SEP] *objects* [SEP]
Original text input
[CLS] *Where does Ted go after leaving the bar ?* [SEP] *Ted goes to Marshall's apartment to tell him about the trip* [SEP] *Marshall : In fact, take my car .* [SEP] *necklace brown shirt woman ...* [SEP]
Masked text input
[CLS] *Where does Ted go* [MASK] *leaving the bar ?* [SEP] *Ted* [MASK] *to Marshall's* [MASK] *to tell him about the trip* [SEP] [MASK] *: In fact, take my car .* [SEP] *necklace brown* [MASK] *woman ...* [SEP]
Answer-removed text input
[CLS] *Where does Ted go after leaving the bar ?* [SEP] [MASK] [SEP] *Marshall : In fact, take my car .* [SEP] *necklace brown shirt woman ...* [SEP]

Table 5.1: Examples of text input of BERT. Original text input is used in a QA network, masked text input is used in a contrastive learning network, and answer-removed text input is used in a self-supervised pre-training stage.

QA. As described by the masked text input example in Table 5.1, we first mask out the tokens of the text input, corresponding to the ground truth answer, with a certain probability using a special token [MASK]. We encode the masked text input using the same BERT and BiGRU, that are used in the video QA section (Figure 5.2 (a) in chapter 4), and denote the encoded representation as an anchor, $\mathcal{H}_{anchor} \in \mathbb{R}^{1\times 2d_t}$. Then, we employ contrastive learning, comparing masked anchor representation and previously extracted text representations,

$\mathcal{H}_t \in \mathbb{R}^{N \times 2d_t}$ in eq. (4.10) from the video QA network. In the representations from the video QA network, we consider the representation corresponding to the ground truth answer as a positive sample, and others as negative samples, and use the dot product to measure the similarity scores between the text representation and the anchor representation.

$$scores = \mathcal{H}_t \mathcal{H}_{anchor}^T \tag{5.1}$$

Then, we apply the softmax to the computed similarity scores and optimize it with the cross-entropy loss that can contrast the positive and negative representations correctly.

$$\begin{aligned} \hat{y}_{con} &= \text{softmax}(scores), \\ \mathcal{L}_{cont} &= -\sum_{i=1}^{N} y_i \log \hat{y}_{con,i} \end{aligned} \tag{5.2}$$

Finally, in addition to the previous scale parameters λ_{qa} of the QA loss and λ_{span} of the temporal localization loss in chapter 4, we introduce a new scale parameter λ_{cont} of the contrastive loss, and the total loss is defined as a linear combination of the above three losses as follows:

$$\mathcal{L} = \lambda_{qa} * \mathcal{L}_{qa} + \lambda_{span} * \mathcal{L}_{span} + \lambda_{cont} * \mathcal{L}_{cont} \tag{5.3}$$

Self-supervised Pre-training

We propose a self-supervised pre-training approach that is applicable to the multiple-choice video question answering task. While the original problem is to predict the answer using a question and text-visual contexts as eq. (4.7), we instead train the model to predict the corresponding question using text-visual

contexts as follows:

$$\hat{q} = \underset{q \in \Omega_q}{\operatorname{argmax}}\, p(q|S, V; \theta) \tag{5.4}$$

where S and V are given contexts which consist of textual and visual contents and q is a given question. θ denotes the trainable parameters. With given S, V, and q, we are to predict the positive question $\hat{q}$ among a set of question candidates.

In this pre-training stage, we randomly sample negative questions for given context to learn the question-context alignment. For each negative training sample, since we previously know which video clips contain which questions, we select questions from other video clips that are not related to the given video clip. In this process, we do not need correct answer annotation, since we replace the part corresponding to the answer option in the input to a single [MASK] token as follows:

$$[\texttt{CLS}]\ q_n\ [\texttt{SEP}]\ [\texttt{MASK}]\ [\texttt{SEP}]\ S_t\ [\texttt{SEP}]\ o_t\ [\texttt{SEP}].$$

The answer-removed text input example in Table 5.1 shows an example used in the self-supervised pre-training stage. In this example, the answer option is "*Ted goes to Marshall's apartment to tell him about the trip*", and we simply mask out this answer option with a [MASK] token not to use the answer information.

And as with the main stage, not only question answering loss but also temporal localization and contrastive losses are also used in the pre-training stage as eq. (5.3). By predicting which question comes from a given context, our proposed network can learn stronger representation with a better parameter initialization to improve the model performance.

5.4 Experiment

We evaluate our proposed approach on multiple-choice question answering dataset: TVQA, TVQA+, and DramaQA. Each video clip is paired with corresponding subtitles and natural language multiple-choice questions. Empirically, our model takes advantage of the supervised contrastive loss in the main stage and gives further improvements when self-supervised pre-training is preceded. Moreover, our model demonstrates significant performance increase on the test server, outperforming the state-of-the-art scores. Our overall performance is shown in Table 4.4, 4.5, and 4.6 in chapter 4.

TVQA+

We conduct an ablation study on the TVQA+ validation set as shown in Table 5.2. For the experiment, we define base models where the token type embedding, temporal localization, contrastive learning, and self-supervised stage are removed. The base models consist of the locally aligned attention model with temporal localization and multiple token type embedding (1-3). In (4), we use the proposed contrastive learning with the masked text input as the auxiliary task. This brings additional performance improvement from 74.54% to 75.16% of accuracy. Lastly, using self-supervised pre-training with a transformed problem format as a prerequisite learning (5), we achieve the best performance of 75.83% accuracy on TVQA+ validation set. It demonstrates that our model takes further advantage of the given dataset using the self-supervised pre-training scheme.

Models	QA (Acc.)
(1) base model (LA)	72.29 ± 0.31
(2) + TL	73.53 ± 0.29
(3) + TL + MT	74.54 ± 0.21
(4) + TL + MT + CL	75.16 ± 0.18
(5) + TL + MT + CL + SS	75.83 ± 0.06

Table 5.2: Results of the ablation study of our model on TVQA+ validation set. We ablate our model with locally aligned attention (LA), multiple token type embeddings (MT), Temporal localization span loss (TL), contrastive loss (CL), and self-supervised pre-training stage (SS).

5.4.1 Analysis

Effectiveness of the proposed contrastive loss

For the contrastive representation learning, as shown in Fig. 5.2(c), among five QA pairs, we contrast a single ground truth answer with the other four negative answers. We investigate how the hidden representations ($\mathcal{H}_t$ in eq. (4.10)) of the five QA pairs (one positive and four negatives) behave depending on whether the contrastive loss is used or not.

We first calculate the distance between the positive and the closest negative representations and report how the distance between them is changing as the epoch continues. For the distance metric, we use the euclidean and cosine distance functions. Figure 5.3 shows that the distance between the positive and the closest negative representations is increasing in both metrics when the contrastive loss is accompanied during the training, while there is no noticeable

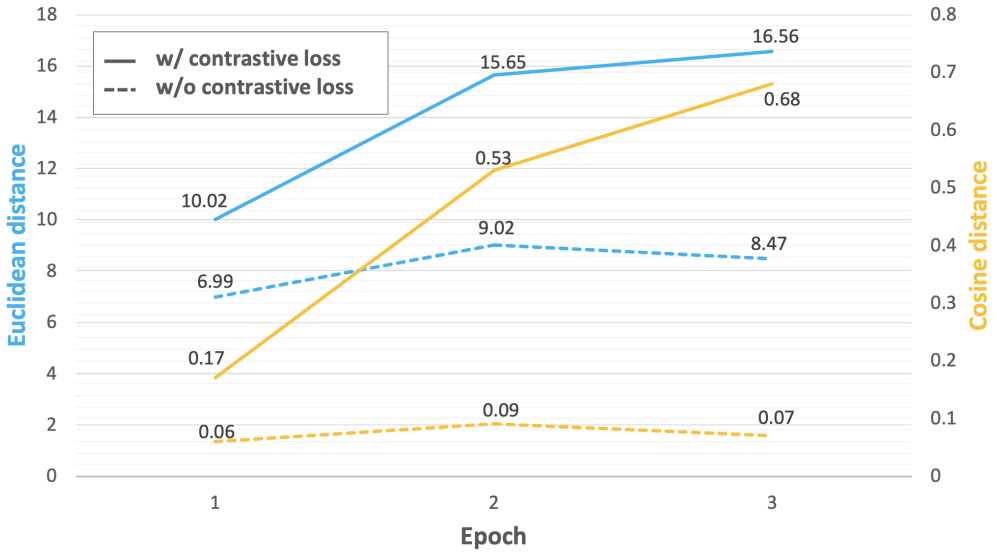


Figure 5.3: Euclidean and Cosine distances between the positive representation and the closest negative representation from the positive one according to whether or not the contrastive loss is used.

increase in distance when the contrastive loss is not used.

Second, we investigate how well the positive representation and four negative representations are separated in their own clusters. We used K-means (k=2) clustering against five representations and denoted cluster accuracy as our metric. For the cluster accuracy metric, we regard a prediction to be correct if one positive representation forms one cluster and the other four negative representations form another cluster, respectively. Figure 5.4 shows that the cluster accuracy increase as we used the supervised contrastive learning in our model as an auxiliary task, and the cluster accuracy of the model with the contrastive loss is much higher than the one without the contrastive loss.

This tells that applying the proposed contrastive loss helps to separate the representation space between the positive and negative samples and we believe

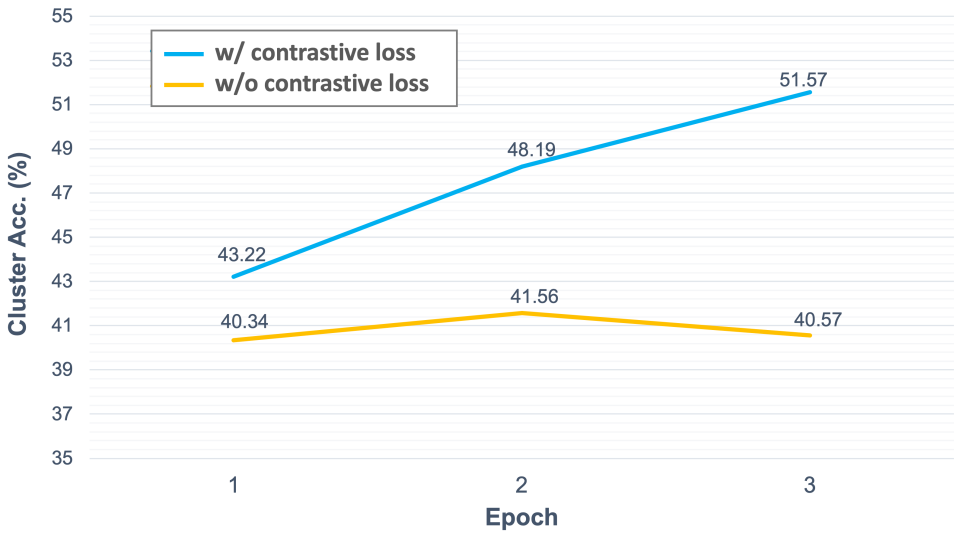


Figure 5.4: Cluster accuracy of the model whether the contrastive loss is used or not. Cluster accuracy is the denoted metric that we regard the prediction is correct if the one positive representation and the four negative representations are well separated in the k-means clustering (k=2).

these separated representations are helpful for predicting the final answer correctly.

Qualitative Results

Figure 5.5 shows two examples of the prediction according to the use of the proposed contrastive representation learning and the self-supervised pre-training schemes (model 3-5 in Table 5.2). In the first example, the proposed model predicts the correct answer by associating Sheldon's dialog, "*officially no longer be roommates*", with the expression "*moving out*" in the correct answer and gives 0.76 of IoU in the temporal localization, while the model without two proposed approaches (model (6)) predicts the wrong answer with only 0.3 of IoU with

the ground truth video span. The second example requires both language and visual understanding to predict the answer and the video span correctly. Our final model localizes the related video span and predicts the answer correctly. However, other models rather pay attention to the word "*door*" which appear in both of the question and the subtitle sentence and fail to predict the correct answer.

5.5 Summary and Discussion

In this chapter, we shift our focus from solving question answering to the training procedure that brings us an additional performance gain. We propose two novel training schemes that can be specialized in multiple-choice question answering.

We first propose the self-supervised pre-training stage. For this stage, we do not require any annotated labels. We transformed the problem format from predicting correct answers to predicting relevant questions. To come up with the pre-training dataset in an unsupervised manner, we randomly sample the negative questions using the context and we can learn better parameter initialization by predicting which questions are from which contexts before fine-tuning the task.

At the fine-tuning as the main stage, we train our model with the original question answering problem which predicts the correct answer with two auxiliary tasks to enhance the model performance. We propose not only the temporal localization that we introduced in chapter 4 but also the contrastive representation learning using the matching scores between the anchor and positive/negative representations.

Our model achieves better performance than the baselines on the challenging multiple-choice video QA datasets. And, we can see that the positive and the negative representations are well separated in the embeddings space by contrastive representation learning in the further analysis. We expect that our proposed method can be applied for various multiple-choice question answering tasks, bringing further performance improvement.

In this work, we showed the possibility of self-supervised pre-training in context-based question answering without using any additional dataset unlike the prevalent pre-training methods using unlabeled large-scale external datasets. This might be a meaningful step especially for low-resource language datasets because there are not many external corpora unlike the case of English.

For our future research, we can utilize more recent fascinating approaches for our pre-training strategy. There is a masked language model loss in pre-training language models to predict the masked token in the sentence. We can develop this loss function into our pre-training method of multiple-choice question answering. We can mask out one subtitle sentence and pre-train to predict the masked sentence using the remaining subtitle context in a format of multiple-choice question answering with negative sentences. Also, we can pre-train the model to predict the order of the subtitle sentences after shuffling the context not to make sense. Through predicting the masked sentence and the right order of the subtitle sentences, we expect that our model might understand the scenario of the video to be helpful for predicting the correct answer in the main QA stage.

For another future research of our contrastive learning across the answer options, we could make our anchor sentence, a noisy correct answer option, in a more sophisticated way. In this work, we randomly masked out the tokens of the

correct answer option. However, we could utilize the entities, the object words which appeared in the video frames, and even the name of the protagonists as our masked tokens to be used as a more challenging auxiliary task. And, we expect that our model can learn the relationship between the modalities or between the entities/protagonists and the events (e.g. behaviors of the protagonists) of the context.

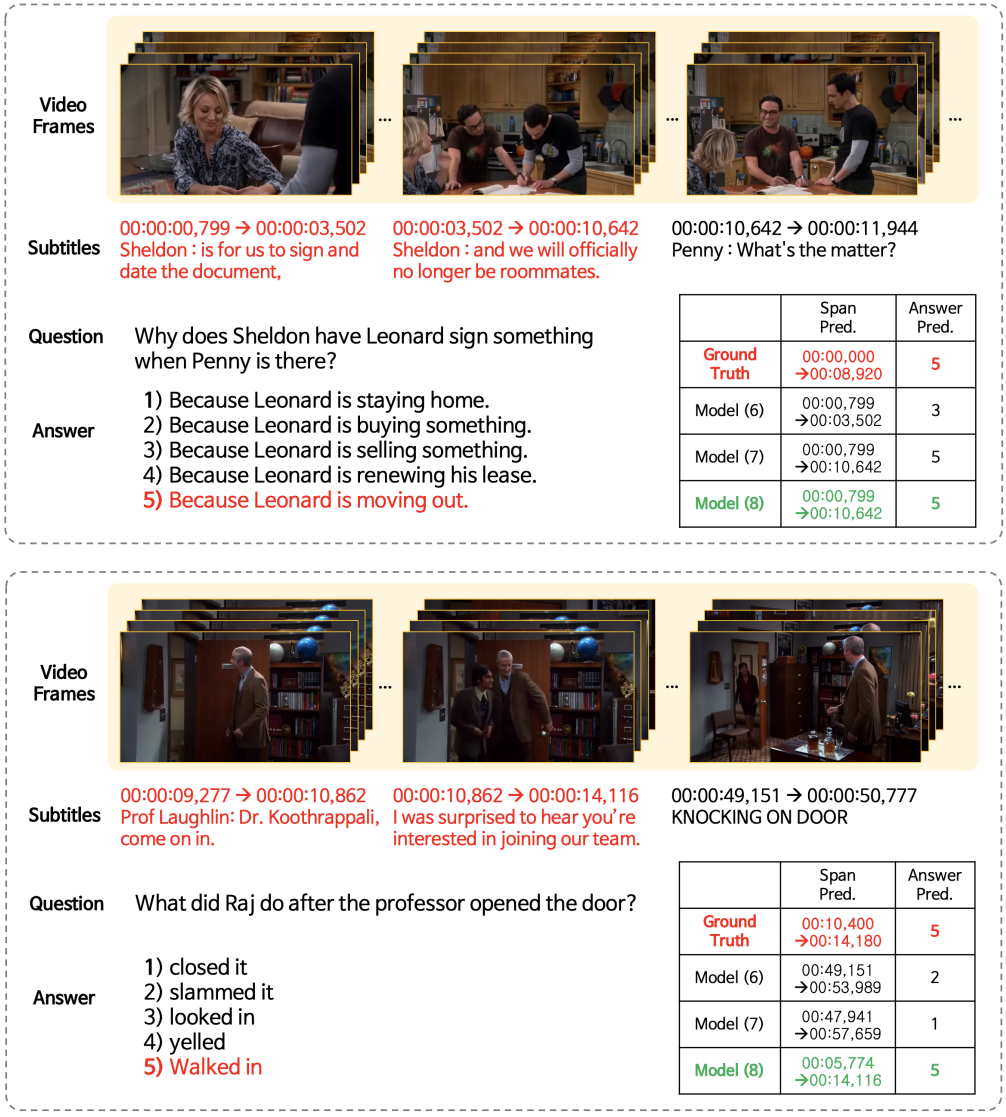


Figure 5.5: Examples of predictions of models with or without the contrastive loss and the self-supervised pre-training scheme. The ground truths are denoted in red, and the predictions of our proposed model are colored in green.

Chapter 6

Conclusion

6.1 Summary

In this dissertation, we addressed text matching approaches for question answering based on the deep neural network. Sentence matching is required to understand the relationship between the sentences, to narrow down the scope of the context to answer the question more precisely, and to calculate the matching score against the question, context, and answer option for ranking the answer especially in the multiple-choice question answering.

In chapter 3, we investigated the sentence pair matching model. We proposed densely connected recurrent neural network architecture to take deep recurrent layers and to utilize all layers' output information as collective knowledge without any deformation. We used the concatenate operation to use the input features of the recurrent layer with the hidden features of all the preceding recurrent layers and the attentive features of the corresponding sentence. It enables preserving the original feature information from the bottommost word embedding layer to the uppermost recurrent layer. However, this ap-

proach causes one issue that the concatenated features are continuously extending and it causes the large size of the model parameters. To mitigate the problem of ever-increasing size of feature vectors due to dense concatenation operations, we also propose to use an autoencoder as a bottleneck and a compression component after dense concatenation with the property of controllable feature sizes. We evaluate our proposed architecture on highly competitive benchmark datasets related to sentence pair matching tasks such as the answer sentence selection, paraphrase identification, and natural language inference. The experimental results show that our architecture, which retains recurrent and attentive features intact, achieved competitive performances for most of the tasks at the time our paper was published. And, when compared to the model using a pre-trained language model like BERT, we can see that the performance still shows the competitive performance, while the number of the model parameters is considerably small than BERT-like models.

In chapter 4, we investigated the context matching for the multiple-choice question answering tasks. In particular, we experimented on the challenging context question answering tasks, Textbook QA, TVQA, TVQA+, and DramaQA, which use the context as textual and visual contents. Textbook QA is a dataset comprised of practical middle school science problems across multiple modalities and it has a number of technical terminologies. Due to this reason, we concentrated on the keyword-based matching between the question/answer and the context after building the dependency trees of the textual context at a sentence level. We designated the matched keyword as anchor nodes. By removing the nodes which have many hops from the anchor nodes, we built the final context graph of the textual context which is used as an input of the graph convolutional network. Also, this process could narrow down the context scope

to answer the question more precisely For the Video QA with TVQA, TVQA+, and DramaQA, we used a temporal localization network as an auxiliary task to focus on the more relevant sentences for answering. We split the subtitle into multiple subtitle sentences and we combined the question, answer option, and the subtitle sentence, all in sentence-level granularity, to be crossly encoded for calculating the matching score.

Lastly, we addressed the training schemes to enhance the performance of the multiple-choice question answering in chapter 5. We proposed the contrastive representation learning that the positive and the negative representations are well separated from each other and learned to be distant. We added the noise to the text input containing the ground-truth answer and designated it as an anchor representation. We calculated the matching score between the anchor representation and the five text representations each of which has a different answer option. By contrasting the positive and the negative representations, we could see that the distance between the positive score and the average of the negative scores gradually increased and the model performance was improved with this contrastive loss. Before fine-tuning the tasks, we also proposed a self-supervised pre-training scheme to learn a better weight initialization. We transformed the original problem format of predicting the correct answer into the one that predicts the corresponding question from the context to provide a model with broader contextual inputs without any further dataset or annotation. By these two training schemes, we achieved state-of-the-art performance in TVQA, TVQA+, and DramaQA.

Text matching such as word or sentence matching is widely used in question answering systems and the development of the matching methods is important to the better QA systems. Also, the training schemes are another important di-

rection to improve the QA systems and they also require the use of the word or sentence matching.

6.2 Future Work

Semantic sentence matching is a fundamental task in question answering. We addressed semantic sentence matching model, multiple-choice question answering tasks with sentence matching, and the training schemes of the multiple-choice QA task with matching scores. However, there is room to improve the question answering system with more sophisticated matching models and strategies, and also we need to broaden the scope of the QA task to the web-based open-domain scale for usability.

We propose the semantic sentence matching model that can be applied for recurrent neural networks, however, it is possible to apply this method to other frameworks such as Transformers recently proposed sophisticated architecture. Every study has used the residual connection as its skip connection, however, it is uncertain that the residual connection is better than the dense concatenated connection. It would be very interesting if the model with a dense connection brings better performance. We could replace the residual connections followed by a feed-forward layer with a dense connection followed by an autoencoder, and the encoder component of the bottleneck autoencoder can play a role of the feed-forward layer of the original architecture of Transformers. By replacing the residual connection with dense connection, we can retain the hidden features intact to be used as a collective knowledge.

For the multiple-choice QA task, Textbook QA, we retrieved one paragraph to compose the context with TF-IDF scores. However, it is difficult to guarantee

that the retrieved paragraph is relevant to the given questions and even some questions require multiple paragraphs as multi-hop reasoning. This approach is limited in that the performance is dependent on the retrieval performance, so if there is no evidence in the retrieved paragraph, it is hardly possible to answer the questions. We need to expand our approach of using one paragraph to the use of whole paragraphs in building the context graph. And for future work, it would be very important to concentrate on retrieving the relevant context, and this can be expanded to the web-scale documents for open-domain question answering. And for context matching, we can use various matching granularities such as document, paragraph, or sentence level for information retrieving.

In the proposed training schemes of multiple-choice QA, we randomly chose the negative samples in the self-supervised pre-training, however, we can extract the harder negative samples by utilizing the matching scores. That is, the model might have a better parameter initialization by solving more difficult problems. Furthermore, for our proposed contrastive learning, we can improve our strategy of adding noise to the correct answer to use it as an anchor sample. Likewise the future work of pre-training strategy, we can mask out the entities or the name of protagonists rather than the random words to make the task being more challenging. We expect that using more challenging auxiliary tasks allows the model to achieve better performance.

We have many chances to use the question answering systems through the search portal or smart devices in our daily lives. And the development of question answering has made people live more efficient, easier, and more convenient. Sentence matching plays an important role in the question answering system. We dealt with three parts, sentence matching model, question answering model, and the training strategies of question answering, and there is very little overlap

between those studies so that they can be incorporated to improve more performance. We hope that our works will help the development of future question answering systems.

Bibliography

[1] A. Abujabal, R. S. Roy, M. Yahya, and G. Weikum. Comqa: A community-sourced dataset for complex factoid question answering with paraphrase clusters. *arXiv preprint arXiv:1809.09528*, 2018.

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[3] W. Bian, S. Li, Z. Yang, G. Chen, and Z. Lin. A compare-aggregate model with dynamic-clip attention for answer selection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1987–1990. ACM, 2017.

[4] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[5] D. Chen, A. Fisch, J. Weston, and A. Bordes. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.

[6] Q. Chen, X. Zhu, Z.-H. Ling, and D. Inkpen. Natural language inference with external knowledge. *arXiv preprint arXiv:1711.04289*, 2017.

[7] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1657–1668, 2017.

[8] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[9] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

[10] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

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

[12] J. Choi, K. M. Yoo, and S. goo Lee. Learning to compose task-specific tree structures. AAAI, 2017.

[13] S. Choi, K.-W. On, Y.-J. Heo, A. Seo, Y. Jang, S. Lee, M. Lee, and B.-T. Zhang. Dramaqa: Character-centered video story understanding with hierarchical qa. *arXiv preprint arXiv:2005.03356*, 2020.

[14] Y.-A. Chung, H.-Y. Lee, and J. Glass. Supervised and unsupervised transfer learning for question answering. *arXiv preprint arXiv:1711.05345*, 2017.

[15] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

[16] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[17] K. Csernai. Quora question pair dataset, 2017.

[18] W. Cui, Y. Xiao, H. Wang, Y. Song, S.-w. Hwang, and W. Wang. Kbqa: learning question answering over qa corpora and knowledge bases. *arXiv preprint arXiv:1903.02419*, 2019.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[21] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*, 2019.

[22] H. Fang, S. Wang, M. Zhou, J. Ding, and P. Xie. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*, 2020.

[23] H. Fang, F. Wu, Z. Zhao, X. Duan, Y. Zhuang, and M. Ester. Community-based question answering via heterogeneous social network learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[24] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016.

[25] S. Geng, J. Zhang, Z. Fu, P. Gao, H. Zhang, and G. de Melo. Character matters: Video story understanding with character-aware relations. *arXiv preprint arXiv:2005.08646*, 2020.

[26] R. Ghaeini, S. A. Hasan, V. Datla, J. Liu, K. Lee, A. Qadir, Y. Ling, A. Prakash, X. Z. Fern, and O. Farri. Dr-bilstm: Dependent reading bidirectional lstm for natural language inference. *arXiv preprint arXiv:1802.05577*, 2018.

[27] Y. Gong, H. Luo, and J. Zhang. Natural language inference over interaction space. In *International Conference on Learning Representations*, 2018.

[28] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017.

[29] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[30] B. Gunel, J. Du, A. Conneau, and V. Stoyanov. Supervised contrastive learning for pre-trained language model fine-tuning. *arXiv preprint arXiv:2011.01403*, 2020.

[31] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[32] H. He, K. Gimpel, and J. Lin. Multi-perspective sentence similarity modeling with convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1576–1586, 2015.

[33] H. He and J. Lin. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 937–948, 2016.

[34] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[35] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[36] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3, 2017.

[37] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data.

In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338, 2013.

[38] J. Im and S. Cho. Distance-based self-attention network for natural language inference. *arXiv preprint arXiv:1712.02047*, 2017.

[39] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2758–2766, 2017.

[40] T. Jurczyk, M. Zhai, and J. D. Choi. Selqa: A new benchmark for selection-based question answering. In *Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on*, pages 820–827. IEEE, 2016.

[41] A. Kembhavi, M. Salvato, E. Kolve, M. Seo, H. Hajishirzi, and A. Farhadi. A diagram is worth a dozen images. In *European Conference on Computer Vision*, pages 235–251. Springer, 2016.

[42] A. Kembhavi, M. Seo, D. Schwenk, J. Choi, A. Farhadi, and H. Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5376–5384. IEEE, 2017.

[43] D. Khashabi, A. Ng, T. Khot, A. Sabharwal, H. Hajishirzi, and C. Callison-Burch. Gooaq: Open question answering with diverse answer types. *arXiv preprint arXiv:2104.08727*, 2021.

[44] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learn-

ing. *arXiv preprint arXiv:2004.11362*, 2020.

[45] D. Kim, Y. Yoo, J.-S. Kim, S. Lee, and N. Kwak. Dynamic graph generation network: Generating relational knowledge from diagrams. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[46] H. Kim, Z. Tang, and M. Bansal. Dense-caption matching and frame-selection gating for temporal localization in videoqa. *arXiv preprint arXiv:2005.06409*, 2020.

[47] J. Kim, M. Ma, K. Kim, S. Kim, and C. D. Yoo. Gaining extra supervision via multi-task learning for multi-modal video question answering. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[48] J. Kim, M. Ma, K. Kim, S. Kim, and C. D. Yoo. Progressive attention memory network for movie story question answering. volume abs/1904.08607, 2019.

[49] J. Kim, M. Ma, T. Pham, K. Kim, and C. D. Yoo. Modality shifting attention network for multi-modal video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10106–10115, 2020.

[50] K.-M. Kim, M.-O. Heo, S.-H. Choi, and B.-T. Zhang. Deepstory: Video story qa by deep embedded memory networks. *arXiv preprint arXiv:1707.00836*, 2017.

[51] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[52] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017.

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

[54] T. M. Le, V. Le, S. Venkatesh, and T. Tran. Learning to reason with relational video representation for question answering. *CoRR*, abs/1907.04553, 2019.

[55] J. Lei, L. Yu, M. Bansal, and T. L. Berg. Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696*, 2018.

[56] J. Lei, L. Yu, T. L. Berg, and M. Bansal. Tvqa+: Spatio-temporal grounding for video question answering. *arXiv preprint arXiv:1904.11574*, 2019.

[57] J. Li, H. Su, J. Zhu, S. Wang, and B. Zhang. Textbook question answering under instructor guidance with memory networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3655–3663, 2018.

[58] M. Liu, X. Wang, L. Nie, X. He, B. Chen, and T.-S. Chua. Attentive moment retrieval in videos. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 15–24, 2018.

[59] P. Liu, X. Qiu, J. Chen, and X. Huang. Deep fusion lstms for text semantic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1034–1043, 2016.

[60] X. Liu, P. He, W. Chen, and J. Gao. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*, 2019.

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

[62] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[63] B. McCann, J. Bradbury, C. Xiong, and R. Socher. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6297–6308, 2017.

[64] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[65] S. Na, S. Lee, J. Kim, and G. Kim. A read-write memory network for movie story understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 677–685, 2017.

[66] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

[67] Y. Nie and M. Bansal. Shortcut-stacked sentence encoders for multi-domain inference. *arXiv preprint arXiv:1708.02312*, 2017.

[68] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[69] E. Pavlick, J. Bos, M. Nissim, C. Beller, B. Van Durme, and C. Callison-Burch. Adding semantics to data-driven paraphrasing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1512–1522, 2015.

[70] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[71] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[72] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[73] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. 2018.

[74] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

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

[76] J. Rao, H. He, and J. Lin. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1913–1916. ACM, 2016.

[77] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[78] M. Richardson, C. J. Burges, and E. Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, 2013.

[79] L. Romano, M. Kouylekov, I. Szpektor, I. Dagan, and A. Lavelli. Investigating a generic paraphrase-based approach for relation extraction. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006.

[80] C. N. d. Santos, K. Wadhawan, and B. Zhou. Learning loss functions for semi-supervised learning via discriminative adversarial networks. *arXiv preprint arXiv:1707.02198*, 2017.

[81] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[82] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

[83] D. Shen, M. R. Min, Y. Li, and L. Carin. Adaptive convolutional filter generation for natural language understanding. *arXiv preprint arXiv:1709.08294*, 2017.

[84] G. Shen, Y. Yang, and Z.-H. Deng. Inter-weighted alignment network for sentence pair modeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1179–1189, 2017.

[85] T. Shen, T. Zhou, G. Long, J. Jiang, S. Wang, and C. Zhang. Reinforced self-attention network: a hybrid of hard and soft attention for sequence modeling. *arXiv preprint arXiv:1801.10296*, 2018.

[86] Y. Shen, W. Rong, Z. Sun, Y. Ouyang, and Z. Xiong. Question/answer matching for cqa system via combining lexical and sequential information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[87] Y. Tay, A. T. Luu, and S. C. Hui. Enabling efficient question answer retrieval via hyperbolic neural networks. *CoRR abs/1707.07847*, 2017.

[88] Y. Tay, L. A. Tuan, and S. C. Hui. A compare-propagate architecture with alignment factorization for natural language inference. *arXiv preprint arXiv:1801.00102*, 2017.

[89] W. L. Taylor. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433, 1953.

[90] G. S. Tomar, T. Duque, O. Täckström, J. Uszkoreit, and D. Das. Neural paraphrase identification of questions with noisy pretraining. *arXiv preprint arXiv:1704.04565*, 2017.

[91] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017.

[92] S. Wan, M. Dras, R. Dale, and C. Paris. Using dependency-based features to take the'para-farce'out of paraphrase. In *Proceedings of the Australasian language technology workshop 2006*, pages 131–138, 2006.

[93] K. Wang, Z. Ming, and T.-S. Chua. A syntactic tree matching approach to finding similar questions in community-based qa services. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 187–194, 2009.

[94] M. Wang, N. A. Smith, and T. Mitamura. What is the jeopardy model? a quasi-synchronous grammar for qa. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

[95] Z. Wang, W. Hamza, and R. Florian. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*, 2017.

[96] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009.

[97] A. Williams, N. Nangia, and S. R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

[98] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[99] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*, pages 451–466. Springer, 2016.

[100] M. Yan, H. Zhang, D. Jin, and J. T. Zhou. Multi-source meta transfer for low resource multiple-choice question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7331–7341, 2020.

[101] L. Yang, Q. Ai, J. Guo, and W. B. Croft. anmm: Ranking short answer texts with attention-based neural matching model. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 287–296. ACM, 2016.

[102] L. Yang, K. Tang, J. Yang, and L.-J. Li. Dense captioning with joint inference and visual context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2193–2202, 2017.

[103] Z. Yang, N. Garcia, C. Chu, M. Otani, Y. Nakashima, and H. Takemura. Bert representations for video question answering. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[104] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[105] L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman. Deep learning for answer sentence selection. *arXiv preprint arXiv:1412.1632*, 2014.

[106] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748, 2018.

초 록

질의 응답 시스템은 딥 뉴럴 네트워크의 발전에 힘입어 자연어 처리 분야에 있어서 중요한 어플리케이션 중 하나가 되고 있다. 본 논문에서는 다양한 질의 응답 모델에서 활용되고 있는 텍스트 매칭 연구를 진행하였다.

먼저 질의 응답 시스템을 구성하는 질문 의미 판단 (Question paraphrase identification), 자연 언어 추론 (Natural language inference), 그리고 정답 문장 선택 (Answer sentence selection) 등에 활용이 될 수 있는 문장 쌍의 의미적 매칭에 대한 연구를 진행하였다. 우리는 상호 연결된 깊은 회기 신경망 구조를 제안하였는데, 이 네트워크는 가장 낮은 레이어인 워드 임베딩부터 가장 높은 레이어까지, 모든 출력 표상 (representation) 들이 변형 없이 이용될 수 있도록 하였다. 다만, 이러한 구조의 문제점으로는 레이어가 깊어질 수록 벡터의 차원이 커진다는 문제가 있는데, 이를 Autoencoder 를 활용하여 큰 차원의 벡터를 압축 함으로써 이러한 문제를 완화하였다.

두 번째로는, 텍스트 컨텍스트로부터 질의 응답을 하기 위해 집중해서 봐야할 부분을 잘 매칭하기 위한 기법들을 제안한다. 먼저, 컨텍스트에 전문 용어들이 많이 등장하는 경우, 키워드를 잘 파악하는것이 중요한데 우리는 이를 위해 컨텍스트 문서의 각 문장에 대해 Dependency Parser 로 컨텍스트 그래프를 구축하였다. 그리고 구축된 그래프에서 질문과 답변에 등장하는 용어가 있는 노드를 앵커 노드로 지정을 하였다. 이 앵커 노드로부터 멀리 있는 노드

들을 삭제함으로써 더 정확한 답을 할 수 있도록, 확인해야하는 컨텍스트의 범위를 좁혔다. 또한, 긴 자막을 가지고 있는 질의 응답 태스크에서, 답변을 하기 위해 필요한 자막 문장을 매칭 점수로 분류하는 분류기를 부가적으로 학습에 활용함으로써 질의 응답의 성능을 높일 수 있도록 하였다.

마지막으로는 객관식 유형의 질의 응답 시스템의 성능을 높이기 위한 학습 방식을 제안하였다. 제안한 학습 방식으로는, 실제 해결하고자 하는 태스크를 진행하기 전 자가지도 사전 학습을 통해 좀 더 좋은 초기 파라미터를 가질 수 있도록 하였고, 실제 학습 단계에서는 대조적 손실 함수를 활용하였다. 먼저, 사전 학습을 위해, 답을 맞춰야 하는 문제의 유형을 변형하여 주어진 컨텍스트에 더 어울리는 질문을 맞추도록 만들어 학습하였고, 본 태스크를 진행하기 전에 좀 더 좋은 모델 파라미터를 가질 수 있도록 하였다. 본 단계에서는 정답과 오답간의 임베딩 영역이 잘 분리 될 수 있도록 대조적 학습 손실 함수를 추가하였고, 이는 최종적인 질의 응답 성능에 도움을 주었다. 이를 통해 TVQA, TVQA+, 또는 DramaQA 와 같은 객관식 유형의 비디오 기반 질의 응답 태스크에서 기존 제안된 다른 모델들보다 더 좋은 성능을 달성할 수 있었다.

주요어: 질의 응답, 딥 뉴럴 네트워크, 텍스트 매칭, 문장 매칭, 자기주도 학습, 대조적 학습

학번: 2017-30004