Ph.D. DISSERTATION

Hierarchical Context Encoder for Natural Language Processing via Leveraging Contextual Information and Memory Attention

자연어 처리를 위한 문맥 정보 및 메모리 어텐션을 활용하는 계층적 문맥 인코더

BY

Hyeongu Yun

AUGUST 2022

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

Hierarchical Context Encoder for Natural Language Processing via Leveraging Contextual Information and Memory Attention

자연어 처리를 위한 문맥 정보 및 메모리 어텐션을 활용하는 계층적 문맥 인코더

BY

Hyeongu Yun
AUGUST 2022

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Hierarchical Context Encoder for Natural Language Processing via Leveraging Contextual Information and Memory Attention

자연어 처리를 위한 문맥 정보 및 메모리 어텐션을 활용하는 계층적 문맥 인코더

지도교수 정 교 민

이 논문을 공학박사 학위논문으로 제출함

2022년 8월

서울대학교 대학원

전기 컴퓨터 공학부

윤 현 구

윤현구의 공학박사 학위 논문을 인준함

2022년 8월

위 원 장:	최 진 영
부위원장:	정 교 민
위 원:	문 태 섭
위 원:	양 인 순
위 원:	권 동 현

Abstract

Recently, the standard architecture for Natural Language Processing (NLP) has evolved from Recurrent Neural Network to Transformer architecture. Transformer architecture consists of attention layers which show its strength at finding the correlation between tokens and incorporate the correlation information to generate proper output. While many researches leveraging Transformer architecture report the new state-of-the-arts performances on various NLP tasks, These recent improvements propose a new challenge to deep learning society: exploiting additional context information. Because human intelligence perceives signals in everyday life with much rich contextual information (*e.g.* additional memory, visual information, and common sense), exploiting the context information is a step forward to the ultimate goal for Artificial Intelligence.

In this dissertation, I propose novel methodologies and analyses to improve context-awareness of Transformer architecture focusing on the attention mechanism for various natural language processing tasks. The proposed methods utilize the additionally given context information, which is not limited to the modality of natural language, aside the given input information. First, I propose Hierarchical Memory Context Encoder (HMCE) which efficiently embeds the contextual information over preceding sentences via a hierarchical architecture of Transformer and fuses the embedded context representation into the input representation via memory attention mechanism. The proposed HMCE outperforms the original Transformer which does not leverage the additional context information on various context-aware machine translation tasks. It also shows the best performance evaluated in BLEU among the baselines using the additional context. Then, to improve the attention mechanism between context representation and input representation, I deeply analyze the representational similarity between the context representation and the input representation. Based on my analy-

ses on representational similarity inside Transformer architecture, I propose a method for optimizing Centered Kernel Alignment (CKA) between internal representations of Transformer. The proposed CKA optimization method increases the performance of Transformer in various machine translation tasks and language modelling tasks. Lastly, I extend the CKA optimization method to Modality Alignment method for multi-modal scenarios where the context information takes the modality of visual information. My Modality Alignment method enhances the cross-modality attention mechanism by maximizing the representational similarity between visual representation and natural language representation, resulting in performance improvements larger than 3.5% accuracy on video question answering tasks.

keywords: deep learning, natural language processing, Transformer, context representation, representation similarity, multi-modal learning, cross-modal attention
student number: 2015-20956

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Transformer, the attention based deep neural architecture to encode and generate natural language texts, have shown a great improvements in various tasks including machine translation, text classification, dialogue generation, and other natural language processing tasks [3]. With the advance of hardware and the large number of data, pre-trained large language morels which adopt the Transformer architecture recently exceeded the human level performance and opened the new horizon in natural language processing [4, 5, 6, 7]. These studies inspire the other researchers to investigate the way how Transformer architecture effectively exploits the information inside the input data and incorporates the information in order to generate the outputs [8, 9, 10, 11, 12, 13, 14, 15].

These aforementioned success of attention based network have also opened a new direction for NLP; utilizing the additional information aside from the given input data. Several researches have reported that the model using the additional information yields better performance [16, 17, 18, 19]. For examples, as described in Figure 1.1, a machine translation model that uses preceding sentences which include useful information generate more accurate and consistent translations, extracting the related information in additional context takes a critical role in open domain question answering system, and amalgamating the visual context information with the textual information

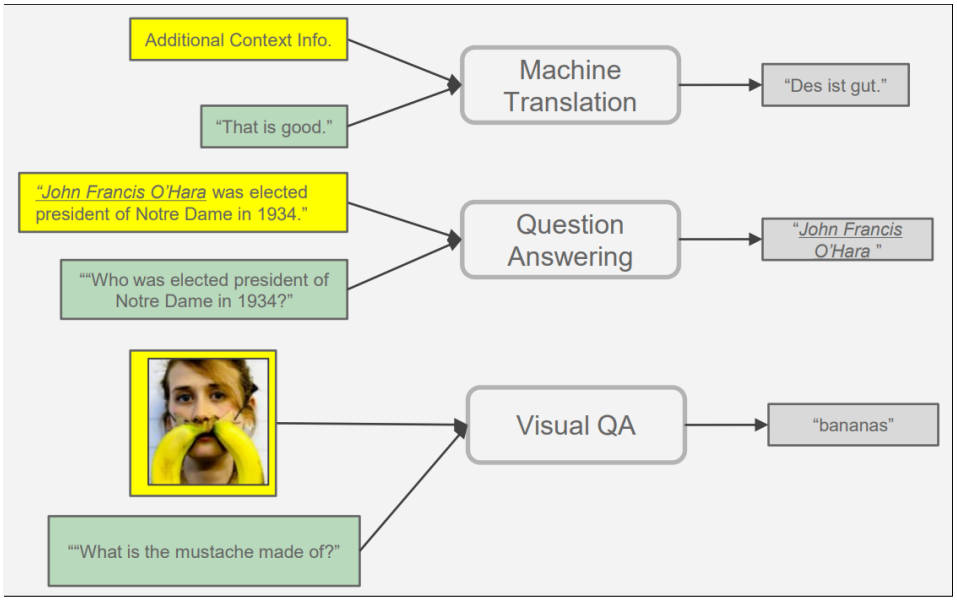


Figure 1.1: The main concept of utilizing context representation. Utilizing additional context representation yields more accurate output by leveraging rich information outside of the given input text.

is required to solve multi-modality tasks.

Nevertheless, there still remains two questions, "How can we properly extract the context information in various form?" and "How can we effectively fuse the extracted context representation into the current model architecture?". In this dissertation, I seek to answer the two questions in natural language processing tasks with the context information given in either text modality or visual modality. The main area of interests in this dissertation is described as Figure 1.2.

To answer the first question, I propose a novel hierarchical Transformer architecture which is designed to extract the contextual information from given data which allows my context-aware Transformer to achieve higher performance. First, I focus on situational awareness, where additional data is explicitly provided as natural language. Transformers have shown excellent results in many natural language processing tasks

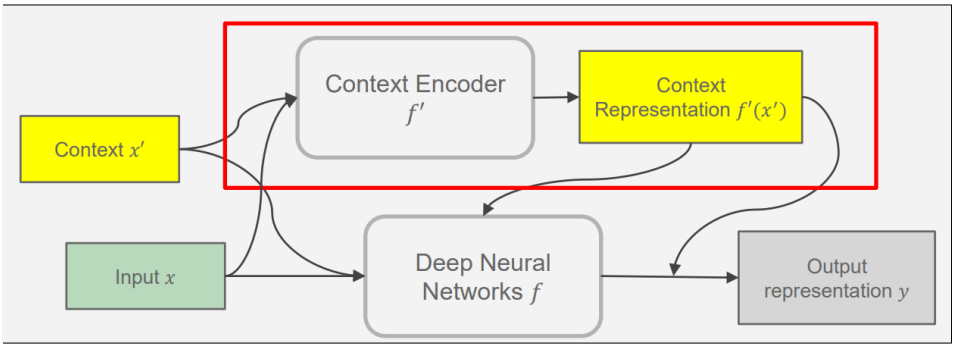


Figure 1.2: The main area of interests in this dissertation. I first focus on designing additional context encoder f' to extract the contextual representation from the explicitly given additional data x'. Then, I propose novel methods to fuse the context representation into the original deep neural networks in order to improve the model performance based on the analyses of representational similarity in Transformer architecture.

and have become solid baselines, but there has been a growing demand for context-aware models that can efficiently utilize contextual information scattered across multiple sentences. In order to meet those demands, I introduce a novel architecture that efficiently encodes context information and fuses extracted representations into current model architectures. Applying my method to machine translation tasks, I propose a new neural machine translation model which outperforms all the baseline models through the following research:

- **Hyeongu Yun***, Yongkeun Hwang*, and Kyomin Jung. "Improving context-aware neural machine translation using self-attentive sentence embedding." Proceedings of the AAAI conference on artificial intelligence. Vol. 34. No. 05. 2020.

This study proposes Hierarchical Context Encoder (HCE) that uses hierarchical transformer structures to leverage multiple context sentences individually. My proposed HCE first abstracts sentence-level information with a self-attentive method from pre-

vious sentences and then hierarchically encodes the context-level information. The hierarchical structure yields not only faster training and inference time compared to the naive concatenation strategy but also higher performance in widely used criterion. I also propose a memory-based encoder architecture upon HCE architecture. Based on the End-to-End Memory Network [20] structure, I make HCE possible to leverage multiple context sentences as vectors. Then my final architecture, Hierarchical Memory Context Encoder (HMCE), utilizes correlated tokens by using attention mechanism in encoder and decoder. Through extensive experiments, I observe that the proposed HCE and HMCE record the best performance measured in BLEU score on English-German, English-Turkish, and English-Korean corpus. I also extend my experiments to document-level machine translation tasks and multi-modal machine translation tasks where HMCE outperforms other baselines. In addition, I observe that HCE records the best performance in a crowd-sourced test set which is designed to evaluate how well an encoder can exploit contextual information. Finally, evaluation on English-Korean pronoun resolution test suite also shows that HCE and HMCE can properly exploit contextual information.

Another main topic of this dissertation is to measure similarity between the contextual representation and the input representation. For deeper understanding of the representation inside Transformer, I analyze the properties of the contextual representation with multiple similarity measures and also propose a novel method to exploit the similarity. I enhance the multi-head attention by optimizing the inter-head diversity through the following research:

- **Hyeongu Yun**, Taegwan Kang, and Kyomin Jung. "Analyzing and Controlling Inter-Head Diversity in Multi-Head Attention." Applied Sciences 11.4 (2021): 1548.

Multi-head attention, a powerful strategy for Transformer, is assumed to utilize information from diverse representation subspaces. However, measuring diversity between heads' representations or exploiting the diversity has been rarely studied. We quantita-

tively analyze inter-head diversity of multi-head attention by applying recently developed similarity measures between two deep representations: Singular Vector Canonical Correlation Analysis (SVCCA) and Centered Kernel Alignment (CKA). By doing so, I empirically show that multi-head attention does diversify representation subspaces of each head as the number of heads increases. Based on my analyses, I hypothesize that there exists an optimal inter-head diversity with which a model can achieve better performance. To examine my hypothesis, I deeply inspect three techniques to control the inter-head diversity; (1) CKA optimization among representation subspaces, (2) Orthogonality regularizer, and (3) Drophead as zero-outing each head randomly in every training step. In the experiments on various machine translation and language modeling tasks, I show that controlling inter-head diversity leads to the best performance among baselines.

Lastly, corresponding to the second question, I apply CKA optimization method to align the context representation and the input representation for effectively amalgamating two different representations. I extend the proposed CKA optimization method to the multi-modal task which includes visual information given by the form of image or video. Multi-modality tasks require the ability of multi-modal reasoning which is to handle both visual information and text information simultaneously across time. In this point of view, a cross-modality attention module that fuses video representation and text representation takes a critical role in most recent approaches. However, existing Video-and-Language models merely compute the attention weights without considering the different characteristics of video modality and text modality. Such naïve attention module hinders the current models to fully enjoy the strength of cross-modality. I enhance the cross attention by optimizing the similarity between visual context representation and textual context representation through the following research:

- **Hyeongu Yun**, Yongil Kim, and Kyomin Jung. "Modality Alignment between Deep Representation for Effective Video-and-Language Learning." Proceedings of The 13th International conference on Language ResIces and Evaluation (LREC),

Marseille, France, June 2022.

I propose a novel Modality Alignment method that benefits the cross-modality attention module by guiding it to easily amalgamate multiple modalities. Specifically, I exploit CKA which was originally proposed to measure the similarity between two deep representations. My method directly optimizes CKA to make an alignment between video and text embedding representations, hence it aids the cross-modality attention module to combine information over different modalities. Experiments on real-world Video QA tasks demonstrate that the method outperforms conventional multi-modal methods significantly with +3.57% accuracy increment compared to the baseline in a popular benchmark dataset. Additionally, in a synthetic data environment, I demonstrate that learning the alignment with Modality Alignment method boosts the performance of the cross-modality attention.

Overall, in this dissertation, I examine transformers in depth from a structural perspective to leverage contextual information with input data. The main contributions of this dissertation can be listed as follows:

- I propose a hierarchical memory context encoder based on the architecture of Transformer encoder as my proposed architecture is able to efficiently exploit preceding sentences or documents as context information in various machine translation tasks.

- I investigate the nature of inter-head diversity among head in multi-head attention. I empirically demonstrate that the inter-head diversity increases as the number of heads increases, which is a widely known but unproven feature of Transformer. Furthermore, I introduce three methods to control and optimize the inter-head diversity in order to find the optimal inter-head diversity.

- I propose a Modality Alignment method that optimize the representational similarity between multi-modal embeddings that has different characteristics. The proposed Modality Alignment method can align video and text representations

to have similar inter-example structure, enhancing the cross-attention module of Transformer.

Through step-by-step structural improvements and extensive experiments, I improve the Transformer encoder architecture in order to efficiently leverage the contextual information in various modalities.

The remainder of this dissertation is organized as follows. In Chapter 3, I propose a novel structure to encode the contextual information scattered across multiple sentences. I verify the proposed context encoder, Hierarchical Memory Context Encoder, with extensive experiments on machine translation and offer various analyses. In Chapter 4, I deeply investigate the similarity measures between deep neural representations. I observe that those similarity measures can be directly applied to my model in various scopes; *e.g.* between the context representation and the input representation, between the multi-head representations, or between the visual context representation and the text representation. Further, I show that Centered Kernel Alignment method [21] can be optimized with the gradient ascent framework to train the optimal similarity between deep representations. In Chapter 5, I extend the CKA optimization method toward multi-modality tasks by introducing the novel Modality Align method. Modality Align method optimizes CKA between visual context representation and text representation in order to improve the cross-modality attention mechanism. I validate Modality Align method on the synthetic environments as well as real-world standard benchmark tasks on image captioning tasks and video question answering tasks. I conclude the dissertation in Chapter 6.

Chapter 2

Backgrounds

In this chapter, we briefly overview several terminologies and methodologies widely used through this dissertation.

Tokenization is a basic methodology to process natural language text into a sequence of "tokens". Various types of tokenization methods have been proposed depending on what is used as the basic unit of tokens; *e.g.* a character-level tokenization sets the unit token as ASCII characters or byte-level characters and a word-level tokenization sets the unit token as words in the pre-defined dictionary. Each tokenization method has its own strengths and limitations. A character-level tokenization method can process the given texts with very small number of vocabulary length (*i.e.* the size of the pre-defined dictionary), but the length of the generated sequence tends to be large. On the other hand, a word-level tokenization method leverages a very large vocabulary set in order to generate shorter sequence. However, a word-level tokenization is often at risk of being exposed to unknown words that are not n the vocabulary; this risk is called Out-of-Vocabulary problem. To take the advantages of both character-level tokenization and word-level tokenization, subword tokenization methods are widely used, such as Byte-pair Encoding (BPE) [22] or WordPiece algorithm [23]. In this dissertation, we mostly use BPE as the tokenization method.

Enc-Dec Model is a general structure to generate output sequence with the given input sequence, hence often called "Seq2Seq Model". For a given sequence $\mathbf{x} = (x_1, ..., x_N)$ with n many tokens, the goal of the Enc-Dec model is to generate the output sequence $\mathbf{y} = (y_1, ..., y_M)$. It is trained to generate the most probable output $\hat{\mathbf{y}}$ with the given $\mathbf{x}$;

$$\hat{\mathbf{y}} = \arg\max_{y} P_\theta(\mathbf{y}|\mathbf{x}), \tag{2.1}$$

where θ is a set of model parameters. In general, the "encoder" part of the Enc-Dec model compute a high-dimensional vector (or a sequence of vectors) containing the information of $\mathbf{x}$ and the "decoder" part generates (or decodes) the output of the encoder. Although there are two main streams (auto-regressive and non-auto-regressive) depending on how the model generates $\mathbf{y}$, we only use the auto-regressive decoding in this dissertation. Auto-regressive decoding generates each output token at a decoding step with the given input sequence $\mathbf{x}$ and the previously generated output sequence $\mathbf{y}_{<m}$. Therefore, the conditional probability of the output sequence $\mathbf{y}$ is modelled as following;

$$P_\theta(\mathbf{y}|\mathbf{x}) = \prod_{m=1}^{M} P_\theta(y|\mathbf{y}_{<m}, \mathbf{x}), \tag{2.2}$$

where M is the maximum length of $\mathbf{y}$.

Transformer is a family of deep neural encoder-decoder architecture proposed by [3]. A unit module of Transformer encoder is composed of a self-attention layer followed by a feed-forward layer with the residual connection. For Transformer decoder, a enc-dec attention layer is inserted in between. All attention layer use QKV attention, which leverage the attention score with given a query vector q, a key tensor K, and a value tensor V. The output matrix X' of the QKV attention is computed as follows;

$$X' = softmax(\frac{qK^T}{\sqrt{d}})V, \tag{2.3}$$

where d is the size of hidden dimension. The authors also introduced the multi-head attention, which is known to diversify the hidden representation and enrich information

to improve the performance of a model. In this dissertation, we use the Transformer architecture for all of our experiments. In particular, in chapter 4, we deeply investigate the properties of the multi-head attention.

Neural Machine Translation (NMT) is a real-world task that takes a part of the natural language processing field. From a given text written in a source language $\mathbf{A}$, the neural machine translation model leverages a deep neural network, usually a Enc-Dec structure, to generate the translated text in a target language $\mathbf{B}$ [24, 25, 26]. Recent studies have discovered that the power of deep neural networks can also be applied to the machine translation tasks. In particular, Recurrent Neural Networks (RNN) based models and Transformer based models have shown remarkable improvements, reported high performance comparable to human experts. In chapter 3 and 4, we conduct extensive experiments on machine translation tasks in various language pairs.

BLEU metric is an automatic evaluation metric for machine-translated texts compared to the the reference text [27]. BLEU measures N-gram overlaps between the generated translation (or hypothesis) and the reference. With computed N-gram precision, usually up to 4-gram, the final score is given as a form of the weighted geometric average.

Multi-modality is a terminology for the type of tasks where input data consists of various form (modality); *i.e.* the input data includes both image and text or both video and text. Multi-modality has became more and more important as it pursues the way how human intelligence perceives everyday life [28, 29, 30]. These tasks require the ability of multi-modal reasoning which is to handle both visual information and text information simultaneously. In chapter 5, we extend our thesis to the multi-modality domain, including the image captioning task and the video question answering tasks.

Cross-modal Attention is a type of QKV attention between embedding sequences from different modalities. Because Transformer architectures also have shown the best performances in several multi-modality tasks, the cross-modal attention can be regarded as a key component to successful multi-modal models. In chapter 5, we suggest to align two different embedding representations from different modalities before computing the attention score.

Chapter 3

Context-aware Hierarchical Transformer Architecture

Context-awareness problem becomes more important for machine translation because the additional context data often include critical information to generate proper translations. For instance, duplicated aforementioned words in preceding sentences tend to be dropped out in colloquial style compared to the formal written style. That omitted information often cause inaccurate, incomplete or ambiguous translations of colloquial style languages which take the most part of spoken languages. Nevertheless, most of commercialized machine translation services are based on Transformer architecture [3] translating a single input sentence to a corresponding sentence in a target language, not taking account of additional context information. Hence, they show lower quality of translation in colloquial styled input such as subtitles of movie or TV series compared to the translation performance in formal styled documents such as news articles. This issue is known as the context-awareness machine translation.

Previous researches have tackled the context-awareness problem with another Transformer encoder that embeds the additional sentences into the context representation [17, 31]. They have introduced the context encoder where the each additional sentence is encoded into the word-level vectors and the remaining decoder part of Transformer use the vectors to generate the translation. If multiple sentences are given as the additional contexts, their context encoder project the additional sentences into a long

sequence of word-level vectors, by concatenating all word-level vectors from the multiple sentences.

However, these methodologies has a critical disadvantage in processing a wider range of context information. The computational efficiency of Transformer increases quadratically with the length of tokens N in each context sentence and the number of context sentences M. Furthermore, Transformer is also empirically known to have limitations in capturing long-distance dependencies in translation tasks. [32, 33]. Therefore, naively concatenating multiple sentences and treating as one long sentence is not only computationally inefficient, but also weakens the context-awareness when the number of context sentences M is large.

To strengthen the context-awareness of Transformer encoder, I propose Hierarchical Context Encoder (HCE) which first encodes each sentence into a sentence-level vector and then hierarchically encodes the sentence-level vectors into a context-level tensor. HCE extract a sentence-level vector from each sentence by the attentive weighted sum module which is a pooling layer with the self-attention. Since each sentence embedding vector contains the contextual information of each contextual sentence, it enables building a context representation tensor by concatenating sentence-level embedding vectors. Then, HCE passes the context representation into another Transformer encoder in order to compute correlative information between contexts. The final output tensor is obtained as a combination of the source tensor and the final encoder output. HCE processes each context sentence separately instead of a long concatenated sentence, hence it shows efficiency in computational complexity. The computational complexity of HCE increases linearly as the number of context sentences increase and HCE shows the fastest running time among standard baseline models in the experiments.

To generate more precise translation, I also propose the fusion of HCE and the memory networks, Hierarchical Memory Context Encoder (HMCE). With HMCE, each context sentence is vectorized using self-attention in the lower stage, which is

the same as HCE. Then, in the upper stage of the HMCE, the source embedding information is utilized by referring to the context-input attention of the memory context encoder as well as self-attention. In the final stage of HMCE, the result of the encoder is emitted through source-context attention similar to HCE.

I conduct a series of extensive experiments on NMT with various language pairs to empirically show that HCE properly yields better translation with multiple context sentences. The experiments include public OpenSubtitles corpus in English-German, English-Turkish and the web-crawled movie subtitles corpus in English-Korean. On all language pairs, I observed that the translation qualities of HCE and HMCE outperform all the other models measured in BLEU score. Also, experiments on document embedding shows that our HMCE outperforms the baseline machine translation models.

Furthermore, I have constructed an English→Korean evaluation set by crowdsourcing in order to analyze how well HCE and HMCE exploits contextual information. The evaluation set consists of two parts, a part where contextual information is helpful for translation and another part where contextual information is unhelpful. I measure translation performances in each part and analyze the effects of contextual encoders by evaluating the performance gap of the two parts. The results from this evaluation set also show that HCE and HMCE outperforms the baseline models.

Overall, the main contributions on this chapter can be summarized as follows:

- I propose a novel architecture for embedding multiple sentences into a tensor in order to exploit contextual information in machine translation tasks.
- I empirically show the effectiveness the proposed HCE by BLEU score, crowdsourced helpful/unhelpful evaluation set and a pronoun resolution test suite.
- I extend HCE with memory attention which shows higher performance in document-level machine translation tasks.

3.1 Related Works

3.1.1 Using Multiple Sentences for Context-awareness in Machine Translation

When designing context-aware machine translation models, it is important to focus on additional contexts. In Statistical Machine Translation (SMT), context-awareness is explicitly modeled designed for the specific discourse phenomena [34]. For example, in translation, anaphora resolution typically involves identifying previously mentioned nouns, numbers, and genders in the source document and manipulating the restoration of the target sentence accordingly.

With the advance of deep learning, neural machine translation models overpowered the previous SMT models. To address context-awareness in NMT models, one can consider either context of the source or the target language. Using the source side of the context requires an encoder for efficient representation of multiple context sentences [35, 17]. On the other hand, the use of target-side contexts often involves multi-pass decoding, which initially translates parts of a document or discourse, and then subdivides translations that use previous translations as target contexts [36, 37]. In this chapter, the proposed models, HCE and HMCE, focus on exploiting the source side of context-awareness only.

The simplest approach to incorporating context into a source document is to concatenate all contextual sentences together and deliver them to a sentence-level model [38]. Additional encoders for contexts have recently been introduced. Additional encoder modules for contextual sentences are natural extensions because the original sentence and context sentence do not have the same importance in translation. In those studies, context sentences are encoded separately and then incorporated into source sentence representations using context-source attention and gating network on encoder [17], decoder [39] or both [31].

3.1.2 Structured Neural Machine Translation Models for Context-awareness

Furthermore, structured modeling of context sentences is also suggested to capture complex dependencies between a source sentence and context sentences. For example, [40] uses Recurrent Neural Networks (RNN) encoders that operate at the sentence and document levels. [35] introduces a hierarchical attention network that encodes context sentences and then summarize the context using hierarchical structures. [41] introduces a memory network augmented model that summarizes and stores context sentences. Because the proposed encoder incorporates hierarchically structured abstractions of encoded context sentences, the proposed method in this chapter is closely related to such approaches. [42] suggests a context attention module that participates in both word and sentence-level contexts. It uses mean word embeddings as sentence-level representations, while ours generates sentence-level tensors with transformer encoders, providing richer sentence representations.

3.1.3 Evaluating Context-awareness with Generated Translation

On the other hand, how to obtain the quality of translation with contextual information is another major research question [39, 43]. Such research focuses primarily on the design of evaluation tasks that evaluate the performance of translation models dealing with discourse phenomenon problems such as pronoun resolution [17, 44]. [37] also suggests that because standard metrics such as BLEU are insensitive to measuring context-to-translation consistency, a carefully designed test suite is important to evaluate context-aware translation models.

3.2 Proposed Approach: Context-aware Hierarchical Text Encoder with Memory Networks

In this section, I briefly describe common parts of encoders in the context-aware NMT framework. I also review other context-aware encoder-decoder structures as our base-

line models. Then I describe a detailed explanation of Hierarchical Context Encoder (HCE) and Hierarchical Memory Context Encoder (HMCE). In addition, I borrow the computational complexity analyses of proposed encoder and other baseline structures from my previous paper.

3.2.1 Context-aware NMT Encoders

NMT models without contexts take an input sentence x in a source language and return an output sentence y' in a target language. I denote a target sentence as y which is used as a golden truth sentence in supervised learning. Each of x, y, and y' is a tensor that is composed of word vectors, also learnable weights during training.

I especially focus on attention based dense models like Transformer [3] which has recently been a standard model for machine translation thanks to its overwhelming performance. Transformer consists of an encoder module and a decoder module, an encoder extract features in x using self-attention and a decoder generate an output y' from the extracted features using both self-attention with itself and attention with the encoder.

Through a single layer in Transformer encoder, an input tensor passes a self-attention layer using multi-head dot product attention and a position-wise feed-forward layer [3]:

$$TransformerEncoder(x) = FFN(MultiHead(x, x)). \tag{3.1}$$

The position-wise feed forward layer, denoted as $FFN(x)$, is composed double linear transformation layer with a ReLU activation in between. The multi-head dot product attention $MultiHead$ and the dot product attention $DotProduct$ are given as follows;

$$\begin{aligned} &MultiHead(q, v) = \\ &\quad [DotProduct(q, v)_1, ..., DotProduct(q, v)_H]W, \end{aligned} \tag{3.2}$$

$$DotProduct(q, v) = softmax(\frac{qW^q W^k v^T}{\sqrt{D}})vW^v, \tag{3.3}$$

where all W denote learnable weights, D is a dimension of hidden space, and H is a number of heads. Both the self-attention layer and position-wise feed-forward layer are followed by skip connection and layer normalization. In addition, a stack with multiple $TransformerEncoder$ is generally used in order to capture more abundant representations.

With N many additional context sentences $[c_0, ..., c_{N-1}]$ are given, an encoder has to capture contextual information among them then combine the contextual information with source sentence representations. I list four previously suggested models as follows, which are also baseline models in the experiments;

- **Transformer without contexts (TwoC)**: As a baseline, I have experimented with Transformer without contexts (TwoC) model which has the same structure as [3]. TwoC completely ignores given additional context sentences and only incorporates with the input x and the target y. The computational complexity is $\mathcal{O}((L_s)^2)$, where L_s is a length of input x.

- **Transformer with contexts (TwC)**: The simplest approach is concatenating all context sentences and an input sentence and consider the concatenated sentence as a single input sentence;

$$x' = Concat([x, c_0, ..., c_{N-1}]). \tag{3.4}$$

  Then, the output of TwoC encoder is the output of a stacked transformer encoder with x'. The computational complexity is $\mathcal{O}((L_s + NL_c)^2)$, where L_c is a fixed length of context sentences. The complexity becomes quadratically expensive as N grows.

- **Discourse Aware Transformer (DAT) [17]**: DAT handles context sentences with an extra context encoder which is also a stacked transformer encoder. I slightly modified DAT to make it available at handling multiple context sentences since [17] is originally designed for handling a single context sentence.

The context encoder has the same structure and even shares its weights with the source encoder through $N_{Layer} - 1$ layers. In the last layer, the context encoder has another transformer encoder module without sharing its weights. The last layer of the source encoder takes an intermediate output tensor h' which is resulted from $N_{Layer} - 1$ stacked transformer encoder, processes both self-attention and context-source attention with t using $MultiHead$;

$$\begin{aligned} t =& Concat([StackedTransformerEncoder(c_0), ..., \\ & StackedTransformerEncoder(c_N)]), \end{aligned} \tag{3.5}$$

$$h_{context} = MultiHead(h', t), \tag{3.6}$$

and

$$h_{source} = MultiHead(h', h'). \tag{3.7}$$

the final output tensor of encoder h is given with the gated sum as follows;

$$h = \sigma(W^h[h_{source}, h_{context}] + b^h), \tag{3.8}$$

where W^h is a learnable weights and b^h is a learnable bias term.

The computational complexity of DAT is $\mathcal{O}(L_s^2 + NL_c^2)$, which is comparable to HCE. However, in order to process context-source attention with multiple context sentences, it concatenates all tensors from each context encoders to a long tensor where long-range dependencies of transformers may be limited.

- **Document-level Context Transformer (DCT) [31]**: The encoder of DCT is similar to the DAT, except for the integration of the context and source encoder. Instead of context-source attention and gated sum at the output of both encoders, each layer of the source encoder takes encoded contextual information t and compute context-source attention followed by point-wise feed-forward layer;

$$h_{context} = MultiHead(h', t), \tag{3.9}$$

and

$$h = FFN(h_{context}). \tag{3.10}$$

Since the extensive use of the context-source attention in the encoder, the computational complexity of DCT is $\mathcal{O}(NL_cL_s + L_s^2 + NL_c^2)$. This can grow prohibitively, especially on handling long context sentences or when the number of context sentences is large.

- **Hierarchical Attention Networks (HAN) [35]**: HAN has a hierarchical structure with two stage at every HAN layer. At the first level of the hierarchy, a single HAN layer encodes each context sentence c_i to an intermediate tensor $e_i \in \mathbb{R}^{L_c \times D}$ with context-source attention;

$$e_i = MultiHead(h', c_i), \tag{3.11}$$

where h' denotes an output from a previous layer or an input x. Each e_i is a tensor with a length of L_c and let e_i^j be the j-th vector of e_i.

At the second level of hierarchy, e_i^j in all context sentences are concatenated through i dimension, resulting tensors $s^j \in \mathbb{R}^{N \times D}$;

$$s^j = Concat([e_0^j, ..., e_N^j]), \tag{3.12}$$

where N is a number of context sentences. Then, an intermediate output tensor t which contains contextual information queried by each word from the input sentence can be given as follows;

$$t = MultiHead(h', s^j). \tag{3.13}$$

All $MultiHead$ layers are followed by position-wise feed forward layers and normalization layers. Finally, the output tensor h of HAN encoder is computed with a gated-sum module introduced by [45]. The aforementioned structure of a single layer in HAN is stacked N_{Layer} times.

The computational complexity of HAN encoder is $\mathcal{O}(NL_cL_s + L_s^2 + NL_c^2)$ which is also comparable to my proposed model. Nonetheless, HAN encoder requires context-source attention two times at every layers. Also, since the second context-source attention is performed on $s_i = Concat([e_0^j, ..., e_N^j])$, HAN does not take account of internal correlations among $[e_i^0, ..., e_i^{L_c}]$.

3.2.2 Hierarchical Memory Context Encoder

I propose a novel context encoder that hierarchically encodes multiple sentences into a tensor. The proposed context encoder, Hierarchical Context Encoder (HCE), is designed to capture correlations between sentences in contexts as well as correlations between words in each sentence.

Each context sentence c_i after word embedding layer is given as a tensor of order 2; $c_i \in \mathbb{R}^{L_c \times D'}$ where L_c is a maximum length of each context sentence and D' is a dimension of word embeddeing vectors. In the lower part of hierarchy, HCE encodes each of c_i to sentence-level tensor e_i using the stacked transformer encoder as [3];

$$e_i = StackedTransformerEncoder(c_i). \tag{3.14}$$

Each encoded sentence-level tensor e_i is also a tensor of order 2, $e_i \in \mathbb{R}^{L_c \times D}$ where D is a hidden dimension.

I then compress each encoded sentence-level tensor into a sentence-level vector by a self-attentive weighted sum module which is similar to that of [46]. The self-attentive weighted sum module takes e_i as an input tensor and computes a vector s_i as follows;

$$s_i = \sum_j \alpha_j e_{ij}, \tag{3.15}$$

$$\alpha = FFN(MultiHead(e_i, e_i)). \tag{3.16}$$

The output of the attentive weighted sum module s_i is a vector representing the information of each i-th context sentence. Then I concatenate $[s_0, ..., s_N]$ to a context embedding tensor s. The context embedding tensor $s \in \mathbb{R}^{N \times D}$ is fed into another

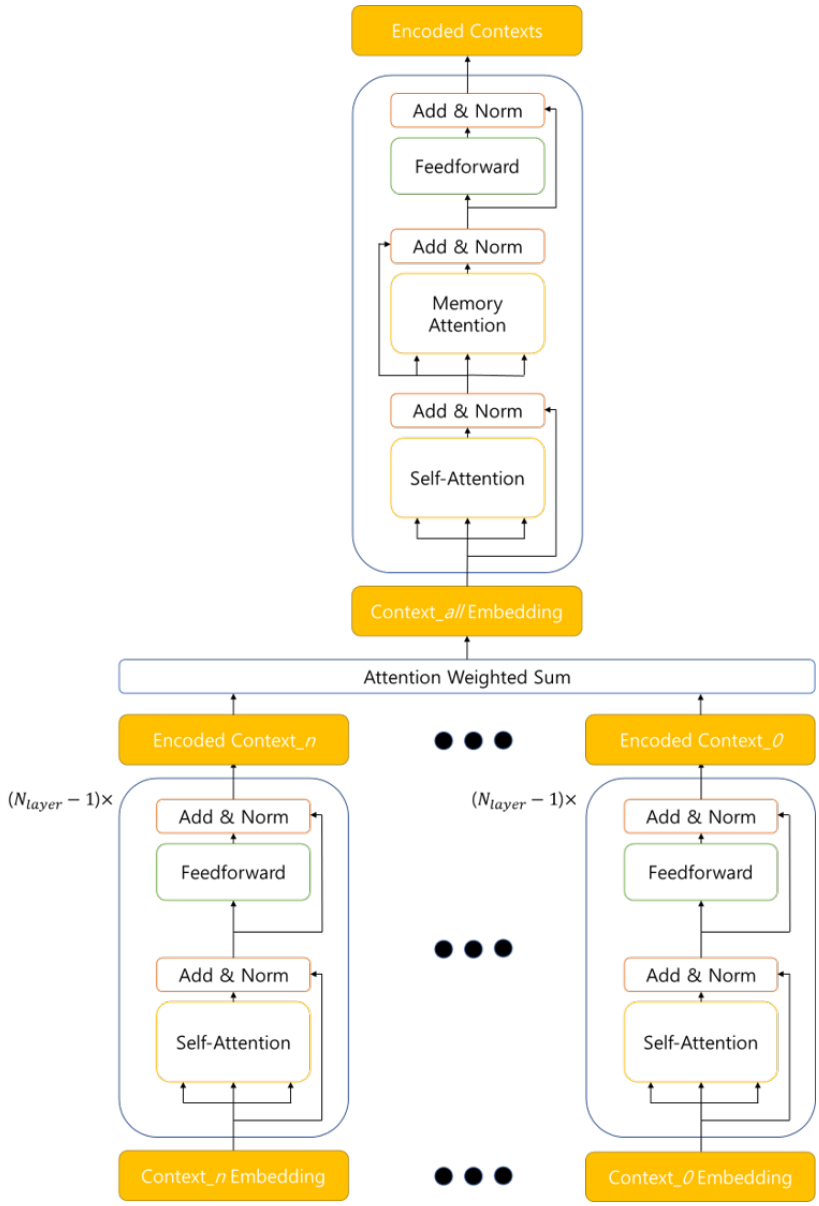


Figure 3.1: Hierarchical Context Encoder. Each Transformers encoder followed by attention weighted sum layer in lower hierarchy encodes each context sentence into a sentence-level vector. Transformer encoder in upper hierarchy takes the sequence of sentence-level vectors as an input tensor and encodes into the context-level tensor.

transformer encoder layer which is the upper part of the hierarchy to encode the whole contextual information into a single tensor t;

$$t = TransformerEncoder(s). \tag{3.17}$$

Finally, the contextual information tensor t is combined to source encoder by gated sum as Equation 3.6, 3.7, and 3.8, which is the same process introduced by [17]. Full structure of HCE is depicted in Figure 3.1.

The main difference between HCE and other baseline models especially HAN is that HCE encodes each context sentence as the way of sentence embedding with self-attention independent to the source word, while HAN uses context-source attention. To explain more in detail, two main differences between the hierarchical transformer structures of HAN and HCE are as follows: 1) at the bottom part of the hierarchy, HCE encodes each context sentence to a tensor with self-attention while HAN encodes each context sentence with context-source attention using query words from input sentences; and 2) at the upper part of the hierarchy, HCE first uses the self-attentive weighted sum to encode a tensor into a vector which contains the whole information from each context sentence, then encodes the whole contexts with self-attention again. On the other hand, HAN uses context-source attention again. To summarize, HCE only models the context-source relations at the upper part of the hierarchy resulting in a simpler and clearer model structure.

HCE encodes each context sentence into a sentence-unit vector through a model composed of two hierarchies, and then allows the vector to encode the whole context-unit tensor through self-attention. In order to improve HCE, I leverage the memory attention as source-context attention that refers source embedding in encoding these vectors into a tensor containing the entire context information. Finally, I propose Hierarchical Memory Context Encoder (HMCE) where each context statement is elaborately encoded through self-attention to affect the encoded context vector through memory attention when referencing it in a subsequent input sentences.

Figure 3.2 describes the structure of HMCE. The encoding process of the context

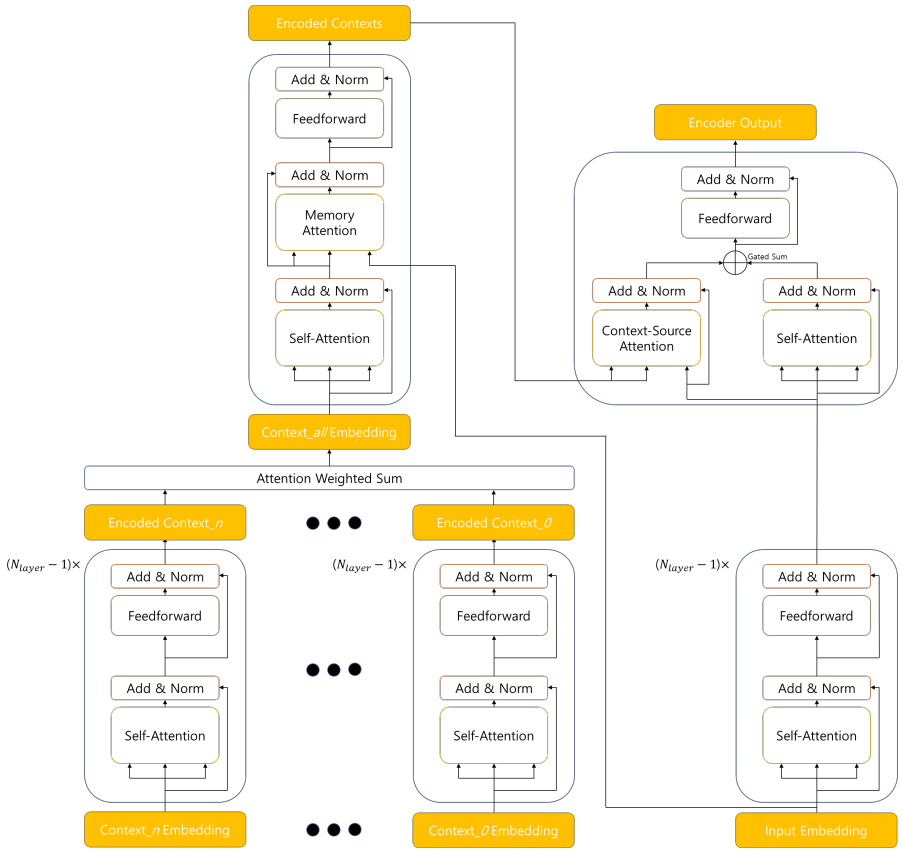


Figure 3.2: The overview of Hierarchical Memory Context Encoder. Upon HCE, Memory attention layer is added after the self-attention in upper hierarchy. The memory attention layer takes its query value from the input representation as it computes the correlation between the input representation and the context-level tensor. To generate the translation, the output of HMCE is fused to the decoder with a gated sum module after a enc-dec attention layer (Context-Source Attention).

and source sentences in the lower hierarchy where the encoding of the sentence unit is performed is the same as that of the HCE. Each $context_n$ embedding is converted into a tensor called Encoded $context_n$ through a total of N self-attention layers. After that, each encoded context is converted into a vector through the attention weighted sum (average over time after self-attention) introduced by the HCE. An out tensor of the lower hierarchy that collects all encoded contexts becomes the input of the hierarchy above. In the upper part, the tensor containing all context information first passes through the self-attention layer and then through the memory attention layer that receives the source embedding as value. Since the memory attention is also multi-head dot product attention, the effect of each context sentence from the word in the source is additionally calculated through this process. Finally, context encoding is completed through feedforward layer, and encoding is completed through source-context attention layer and gated sum.

The computational complexity of HCE and HMCE are both $\mathcal{O}(L_s^2 + NL_c^2)$. HCE and HMCE extract more compact context-level representation from each sentence-level representation by self-attentive weighted sum over each e_i, hence it complements DAT [17] and DCT [31] whereas they take the whole contexts as a single sentence by concatenation. Besides, the encoding procedure of context sentences is not dependent on the input sentence x unlike HAN. This allows HCE to cache context-level representations t of frequently appeared context sentences, which is important in implementing a real-time application.

3.3 Experiments

I evaluate HCE and HMCE by BLEU score, model complexity, and BLEU scores on helpful/unhelpful set. All experimental results show the effectiveness of the proposed structures compared to baseline models.

Table 3.1: Bilingual subtitle samples from the web-crawled English-Korean test files

Start Time	End Time	English	Korean
. . .			
337733	339967	Daniel likes hanging out with his cousins.	다니엘은 사촌들과 노는걸 좋아했거든요
340035	341168	He's been going back and forth until Leith and I	양육권을 제대로 가질 수 있을때까지
341236	342303	can settle custody.	왔다 갔다 했어요
344373	345940	Listen, don't worry.	너무 걱정 마세요
. . .			

3.3.1 Data

I experimented with HCE, HMCE and baseline models on English-German TED corpus, English-German OpenSubtitles corpus, English-Turkish OpenSubtitles corpus, English-German WMT19 corpus, and the web-crawled English-Korean subtitle corpus.

English-German IWSLT 2017 Corpus

I use the English-German corpus from the IWSLT 2017 evaluation campaign [47], which is publicly available on WIT^3 website[1]. The corpus consist of transcriptions and their translations of TED talks. I combine `dev2010` and `tst2010` into a development(*dev*) set and `tst2015` as a *test* set. I extract context-aware dataset where each set consists of a *source*, a *target* sentence and multiple *context* sentences. Since the corpus is aligned as sentence level, I assume that every 2 preceding sentences are *context* sentences. I also include context sentences only within the same talk of the source sentence, as the data is separated as talks. The resulting dataset consists of $211k$, $2.4k$, $1.1k$ examples of *train*, *dev*, *test* sets respectively. Also, I put a special *beginning of context* token at the beginning of each context sentences to differentiate from source sentences. Finally, I have used a byte-pair encoded vocabulary with about 16,000 tokens.

[1] https://wit3.fbk.eu/mt.php?release=2017-01-trnted

OpenSubtitles Corpus

I also choose the OpenSubtitles corpus for English-German and English-Turkish tasks. I use the 2018 version [48] of the data, each consist of $24.4M$, $47.4M$ parallel sentences respectively. Following the approach in [37], I first cleaned the data by picking only pairs with a time overlap of subtitle frames at least 0.9. After cleaning, I take $7.5M$ and $20.2M$ sentences for English-German and English-Turkish corpus.

I then take the *context* sentences by using the timestamp of each subtitle. The timestamps contain start time and end time in ms for each subtitle. I focus on the start times to compile a set of data including a source sentence and preceding contextual sentences. I assume that if the start time of a preceding sentence is within 3000 ms from the start time of a sentence then that preceding sentence contains the contextual information. I set the maximum number of preceding contextual sentences up to 2.

English-Korean Subtitle Corpus

Finally, for English-Korean experiments, I construct a web-crawled subtitle corpus with 5,917 files. These files are English-Korean bilingual subtitle files of movies, TV series, and documentary films from various online sources. I set randomly selected $5.3k$ files for *train*, 500 files for *dev*, and 50 files for *test* set. The *train* set includes $3.0M$ sentences, the *dev* set includes $28.8k$ sentences, and the *test* set includes $31.1k$ sentences. The web-crawled English-Korean bilingual subtitle files include time stamps for each subtitles. Thus I pre-process those files as similar as processing in Section 3.3.1. The resulting data have $1.6M$ sets of serial sentences in *train set*, $155.6k$ sets in *dev set*, and $18.1k$ sets in *test set*. I also have used a byte-pair encoded vocabulary with about 16,500 tokens for English-Korean experiments. I display some raw samples from the test set in Table 3.1.

3.3.2 Hyperparameters and Training Details

Through my experiments, I use 512 hidden dimensions for all layers including words embedding layers, transformer layers, and the encoded context layer. I set $N_{Layer} = 6$ for all models and share the weights of the source encoder to context encoder for the DAT, HAN, and HCE models. For all attention mechanisms, I set the number of heads as 8. The dropout rate of each layers is set to 0.1.

For each language pair, I tokenize each text by the wordpiece model [49, 23] with a vocabulary of about 16,000 tokens. Also, I put a special *beginning of context* token `<BOS>` at the beginning of each context sentences to differentiate from source sentences.

I implement all the evaluated models using the `tensor2tensor` framework [50]. I train all models with ADAM [51] optimizer with learning rate 1e-3 and adopt early stopping with *dev* loss. Unlike [35, 31, 42], I do not use the iterative training which trains the model on a sentence-level task first, then fine-tunes the model with contextual information. All the models I have evaluated are trained from scratch with random initialization.

For scoring BLEU, I use the `t2t-bleu` script[2] which outputs the identical results as Moses script [52].

3.3.3 Overall BLEU Evaluation

I measure performances of HCE and other five baseline models in English-German (IWSLT'17 and OpenSubtitles), English-Turkish (OpenSubtitles), and English-Korean(The Web-crawled corpus). Overall BLEU scores on all eight datasets are displayed in Table 3.2. HCE yields the best performances on all eight datasets. Especially on the Web-crawled English-Korean, HCE shows superior performance compared to other models. These results indicate that HCE and HMCE exploit given contextual sentences effectively and translate better than all five baseline models in English-German, English-

[2] `https://github.com/tensorflow/tensor2tensor`

Table 3.2: Overall Translation Quality Evaluated with BLEU score upon 2 context sentences. The proposed Hierarchical Context Encoder have shown the best results in all language pairs.

Corpus	IWSLT'17		OpenSubtitles		OpenSubtitles		Web-crawled	
Language pair	En→De	De→En	En→De	De→En	En→Tr	Tr→En	En→Ko	Ko→En
Transformer without contexts	28.25	32.18	27.95	33.93	24.89	36.27	8.58	23.67
Transformer with contexts	28.65	32.68	28.07	34.04	23.96	35.81	9.46	24.23
DCT [31]	26.76	30.33	26.3	32.05	21.91	34.3	6.5	20.72
DAT [17]	28.82	32.59	28.09	33.99	24.30	35.23	8.56	23.91
HAN [35]	28.85	32.72	28.00	34.42	24.86	36.55	8.76	24.41
HCE	**28.89**	**33.01**	**28.40**	**34.59**	**25.11**	**36.84**	**11.30**	**26.70**
HMCE	28.73	28.94	28.31	34.48	24.24	35.93	9.80	24.65

Table 3.3: BLEU score on multiple context sentences.

Model	En→De	En→De
TwC	23.20	28.55
HCE with 2 context sentences	23.32	28.69
HCE with 2 context sentences + Document vector	23.59	29.27
HCE with 10 context sentences + Document vector	23.71	29.30
HMCE with 2 context sentences	24.56	28.85
HMCE with 2 context sentences + Document vector	24.82	**29.63**
HMCE with 10 context sentences + Document vector	**24.96**	29.61

Turkish and English-Korean translation tasks. HMCE also records comparable BLEU score to HCE but slightly lower.

In document-level NMT using WMT 2019 En-De corpus, HMCE shows its advantages. Unlike the previous experiments where I use two preceding sentences as contexts, I expand the context range up to 10 sentences. As a result of the experiment, both HCE and HMCE show better performance than other baselines of the same number of learning times when fine-tuned by adding a document level vector. Although it is not as dramatic as in En-Ko corpus, there is an improvement in BLEU score by about 5% compared to the Transformer in sentences based on En-De. On the other hand, in comparison between HCE and HMCE, the addition of the memory attention module and the document level vector shows a performance improvement of about 2-3%. Therefore, it can be considered that the document level information helped improve translation quality.

3.3.4 Model Complexity Analysis

I also observe that HCE is the most efficient in training speed and inference time among all baselines. In Table 3.4, HCE records the fastest training speed and inference time indicating that HCE has the most computationally efficient structure. As well

Table 3.4: Training speed, inference time and number of parameters.

Model	training speed (steps/sec)	inference time (tokens/sec)	# of Params
TwC	4.07	62.10	61.0M
DCT	2.42	45.32	98.7M
DAT	4.59	65.07	69.9M
HAN	4.47	64.05	66.2M
HCE	**4.67**	**65.12**	66.7M
HMCE	4.65	65.07	68.8M

as HCE, HMCE also has more time-efficient structure than other baselines. HMCE records comparable speed in both training and inference. These results also show that the performance gain of HCE and HMCE is not only from the complexity of the model but the structural strength because the number of parameters is comparable to others.

3.3.5 BLEU Evaluation on Helpful/Unhelpful Context

In order to verify that HCE and HMCE effectively utilize the contextual information to improve translation quality, I conduct an additional experiment with a part of data where contextual sentences are helpful for translating and the other part of data where they are not. I randomly choose 10,000 sets of serial sentences from the *test set* of En→Ko data and split them up into two parts by crowd-sourcing with Amazon Mechanical Turk [53]. The first part consists of 4,331 sets of which context sentences are helpful for translating (*e.g.* context sentences include critical information, exact referred object by pronouns, or residual parts of an incomplete source sentence). The remaining part consists of 5,669 sets of which context sentences are unrelated to translate the source sentences.

I examine BLEU scores of two parts separately to observe how well each model uses helpful contexts. The results are displayed in Table 3.5. I observe a large gap

Table 3.5: BLEU score evaluations with helpful contexts set and unhelpful contexts set from En→Ko test data. All four baseline models have shown large gap between BLEU score on *helpful* contexts set and BLEU score on *unhelpful* contexts set. On the other hand, the proposed Hierarchical Context Encoder has almost closed the gap between BLEU scores on two sets.

Model	Total set	helpful set	unhelpful set	BLEU gap
Transformer without contexts	7.46	6.69	8.04	+1.35
Transformer with contexts	8.29	7.45	8.92	+1.47
DAT [17]	8.22	7.48	8.77	+1.29
HAN [35]	8.34	7.44	9.01	+1.57
HCE	10.27	10.08	10.40	**+0.32**
HMCE	9.56	8.78	10.16	+1.38

between BLEU score on *helpful set* and that on *unhelpful set* with all four baseline models, showing that *helpful set* is harder to translate because abstracting and exploiting contextual information is likely to be mandatory to translate *helpful set*. On the other hand, HCE closes the gap between BLEU scores on each set, indicating that HCE understands the contextual information and is able to perform on *helpful set* as well as on *unhelpful set*.

3.3.6 Qualitative Analysis

Table 3.6 shows context-aware translation examples from the test set. As in the first example, HMCE is able to capture the context of "*didn't know*" from Context 1 and yields the corresponding Korean translation "*할 줄 알았거든요.*" whereas Google Translator does not concern such context. The second example also shows the strength of HMCE that is able to complete the translation although the input sentence is a phrase rather than a complete sentence. While Google Translator yields a Korean phrase which is correspond to the word-by-word translation, HMCE captures the context of "*did find*

Table 3.6: English→Korean Context-aware translation examples.

Example #1	
Context 1	Yeah.
Context 0	See, I didn't know that I was coming here
Input	and we were gonna have a serious conversation.
HMCE	진지한 대화를 할 줄 알았거든요.
Google	그리고 우리는 진지한 대화를 나누기로 했습니다.
Example #2	
Context 1	Unfortunately, no.
Context 0	But I did find something
Input	on the other recording that the killer left us.
HMCE	범인이 남긴 다른 녹음에서 뭔가 찾았어요.
Google	살인자가 우리에게 남긴 다른 녹음에서.
Example #3	
Context 1	I am just stating the facts, ma'am.
Context 0	Next time you talk to the AUSA,
Input	I'd share that little fact.
HMCE	나라면 그 사실을 공유하겠네.
Google	그 작은 사실을 공유합니다.
Example #4	
Context 1	What'd it say? I don't know. I haven't listened to it.
Context 0	I've been too afraid, you know, to hear his voice.
Input	Well, do you want to listen to it right now?
HMCE	지금 들어볼래?
Google	자, 지금 바로 듣고 싶으세요?

something" from the preceding context sentence and generate the complete translation sentence. In the third example, HMCE generates the translation in recommendation style corresponding to "*I'd*" in the context, yet Google Translator translates into the meaning of "I will share ..." which is less proper than HMCE's translation. Nevertheless, in few examples such as the fourth, HMCE often omits too much information (it omits the meaning of "*Well,*" and "*right*"), which I suspect the reason is such omissions are quite often in the web-crawled spoken-style dataset.

3.3.7 Limitations and Future Directions

Although I show that HCE and HMCE outperform other baseline architectures using preceding sentences for machine translation, there still remain a few limitations of this study. I compare these limitations with the latest studies one by one, and suggest the direction to move forward.

- **Decoder-friendly Architecture**: HCE and HMCE are mainly focused on the encoder side of Transformer. Hence, HCE and HMCE are limited to use encoder-side contextual information, such as the preceding sentences of input languages. Additionally, this limitation may keep these architectures from leveraging the power of big language models (*e.g.* GPT-3[7]), because most of big models adopt decoder-only architectures. To overcome such limitation, I suggest a future direction of this study as applying hierarchical structure and memory attention into the decoder-only architecture in future studies.

- **Exploiting Following Sentences**: In the experiments, I only utilize one to three preceding sentences before the input sentence which is to be translated. There may be benefits to use following sentences which come after the input sentence, since there are often important information at the end of a paragraph or a document particularly in spoken languages. However, when applying these methods in real-life scenarios, one must take care of the inference phase where there are

not given information of future unlike to the training phase. In order to utilize various form of context information, deeper understanding of internal characteristics of Transformer is mandatory. Therefore, I suggest to study further about the characteristics of representation inside Transformer to achieve better understanding of attention layer. This directly leads to the next part of this dissertation, the analyses on representational diversity of Transformer.

3.4 Conclusion

In this chapter, I have proposed Hierarchical Context Encoder (HCE) structure and Hierarchical Memory Context Encoder (HMCE) which are able to efficiently exploit multiple contextual sentences. Based on Transformer architecture, HCE and HMCE outperform all baselines in various machine translation tasks including English-German, English-Turkish and English-Korean translation tasks. Also the proposed models are the most efficient in terms of computational complexity. I also have shown that HCE closes the gap of translation quality between the sentences with helpful contexts and the sentences with unrelated contexts, implying that HCE is better at exploiting the helpful contextual information for translating than baseline models. In document-level machine translation, HMCE also have recorded the highest performance among the baseline models.

Chapter 4

Optimizing Representational Diversity of Transformer Architecture

Since multi-head attention has been introduced by Vaswani et al. [3], it has become a standard setting across various Natural Language Processing (NLP) tasks. Vaswani et al. have stated that multi-head strategy can collocate information from different representation subspaces and thus improves the performance of attention mechanism, whereas single-head attention averages the information. Most of the state-of-the-arts models report that multi-head attention is helpful to increase their performances, including BERT [4] and XLNet [6] for language understanding, Transformer [3] for machine translation, and HIBERT [54] for document summarizing.

Despite its huge empirical success and dominant usage, few studies have explored the roles of the multi-head strategy to give us a better understanding on how it enhances a model's performance. Clark et al. [8] have analyzed attention maps of multi-head attention and showed that certain heads are relevant to specific linguistic phenomena. Similarly, Voita et al. [9] has analyzed that certain heads are respectively sensitive to various linguistic features by using layer-wise relevant propagation. Although these studies imply that there exists diversity of representation subspaces among multiple heads, their analyses are mainly focused on linguistic diversity.

In order to inspect essential effects of multi-head attention in representational subspaces, Li et al. [55] have proposed the disagreement score which measures cosine similarity between two heads' representation and maximized the disagreement score to diversify inter-head representations. Li et al. have shown that maximizing the disagreement score increases performance, which implies that inter-head statistics in multi-head attention are closely related to the model's performance. However, disagreement score has its limitation since cosine similarity of two random vectors in high dimension are close to 1, as known as the curse of dimensionality.

To overcome the limitations of previous studies, I seek answers to following three fundamental questions: (1) Does multi-head strategy diversify the subspace representations of each head? (2) Can we finely optimize the degree of inter-head diversity without changing model's architecture? and finally (3) Does controlling inter-head diversity improve a model's performance?

I measure the inter-head similarity of multi-head attention with Singular Vector Canonical Correlation Analysis (SVCCA) [10] and Centered Kernel Alignment (CKA) [21], as they are recently developed tools to measure similarities of two deep representations. Applying these similarity measures, I empirically show that the diversity of multi-head representations does increase as the number of heads increases which is solid evidence supporting the statement of Vaswani et al. [3] that the multi-head strategy utilizes diverse representational subspaces. Furthermore, I suggest three techniques to optimize the degree of diversity among heads without architectural change of a model.

I first focus on trainability of CKA because CKA is differentiable and its gradients can be easily computed with popular frameworks such as Tensorflow [56]. I adopt Hilbert-Schmidt Independence Criterion (HSIC) inspired by CKA as an augmented loss in order to directly diversify the inter-head diversity of a model.

Then, I revisit the orthogonality regularizer that adds disagreement loss [55] between representations of heads. Surprisingly, opposed to the expectation of Li et al. [55]

expected, I empirically show that the orthogonality regularizer does not force a model's inter-head diversity to increase measured in SVCCA and CKA. Instead, I find that it helps a model by encouraging top-few SVCCA directions to be closer which can be interpreted as core representations [57].

Lastly, I inspect Drophead method [58] by which a model randomly drops outputs of each head at training to show that I also can decrease the inter-head diversity without architectural change. Drophead reduces an effective number of heads at each training step and hence increases the inter-head similarity, while a model also benefits from the advantages of Dropout [59].

I test the methods on various tasks including De-En IWSLT17 corpus [60], Zh-En in UN parallel corpus [61] on machine translation, and also PTB corpus on language modeling. The experimental results show that the suggested three methods complement each other and find the optimal inter-head diversity. The models with the proposed methods achieve higher performances compared to their baselines in all experiments.

4.1 Related Works

4.1.1 Analyses of Diversity in Multi-Head Attention

As the multi-head strategy has shown its strength in many NLP tasks, there have been several attempts to analyze it with various approaches. By evaluating attention weights of ambiguous nouns in machine translation, Tang et al. [62] have shown that multi-head attention tends to focus on ambiguous tokens more than general tokens. Clark et al. [8] and Raganato et al. [11] also have analyzed attention weights and concluded that each head plays different roles to understand syntactic features. Voita et al. [9] and Michel et al. [13] have claimed that most of the heads can be pruned once the model trained as they have analyzed the multi-head mechanism via layer-wise relevant propagation and ablating heads respectively.

On the other hand, several works have tried to analyze the similarity between representation spaces of neural networks in favor of achieving interpretability. Li et al. [12] have proposed alignment methods with a correlation of neurons' responses and claimed that core representations are shared between different networks while some rare representations are learned only in one network. More recently, Raghu et al. [10] have first applied CCA as a similarity measure and proposed SVCCA in order to pick out perturbing directions from deep representations, and Morcos et al. [63] have suggested Projection Weighted CCA (PWCCA) as a method to make SVCCA more reflective to the subspaces of representations via projection. Kornblith et al. [21] have proposed CKA as a more robust similarity measure to small numbers of samples using a normalized index of HSIC with kernel methods.

4.1.2 Similarities between Deep Neural Representations

Towards the interpretability of the deep representation, some studies have utilized similarity measures of deep representations. Maheswaranathan et al. [64] have applied CCA, SVCCA, and CKA to Recurrent Neural Networks (RNN) and discovered that the geometry of RNN varies by tasks, but the underlying scaffold is universal. Kudugunta et al. [65] have applied SVCCA across languages on multilingual machine translation to show there are shared representations among language representations. Bau et al. [57] also have applied SVCCA to identify meaningful directions in machine translation and showed that top-few directions in SVCCA similarity are core representations since they are critical to a model's performance when erased.

Closely related to the orthogonal loss, decorrelation methods have been proposed in node level [66, 14, 67] and in group of nodes level [55, 68]. Rodriguez et al. [66], Xie et al. [14], and Bansal et al. [67] have shown that decorrelating each node through orthogonal constraint can achieve higher performances. Li et al. [55] have applied the decorrelating term to multi-head attention, which inspires us to use orthogonal constraints in order to control inter-head diversity. Gu et al. [68] have showed that

cosine similarity based constraint in group of nodes can achieve higher performances, as it improves generalization capacity of the model.

4.2 Similarity Measures for Multi-Head Attention

4.2.1 Multi-Head Attention

Multi-head attention is first suggested by Vaswani et al. [3] as a strategy that diversifies representation subspaces in order to fully utilize a model's capability. I briefly review how single-head and multi-head attention operates.

For single-head attention, an output matrix $X' \in \mathbb{R}^{L \times d}$ with its inputs (a query vector $q' \in \mathbb{R}^d$, a key matrix $K' \in \mathbb{R}^{L \times d}$, and a value matrix $V' \in \mathbb{R}^{L \times d}$) is computed as follows:

$$\text{X'} = \text{softmax}(\frac{q'K'^T}{\sqrt{d}})V', \tag{4.1}$$

where L is a length of key and value matrix and d is a hidden dimension size. The single-head attention first computes attention weights by taking softmax function onto similarity scores between a query vector and key matrix, then finally operates multiplication with value matrix which can be considered as a pooling operation from a value matrix with the attention weights.

On the other hand, multi-head attention operates H-many single-head attentions in parallel with $q_i \in \mathbb{R}^{d_h}$, $K_i \in \mathbb{R}^{L \times d_h}$, $V_i \in \mathbb{R}^{L \times d_h}$, where q_i, K_i, V_i are projections of q, K, V onto smaller dimension d_h with weight matrices W_i^q, W_i^K, $W_i^V \in \mathbb{R}^{d \times d_h}$ respectively for each i-th head. The output of multi-head attention is calculated by concatenating all outputs of H-many heads followed by final linear projection:

$$X = [X_1, ..., X_H]W^O, \tag{4.2}$$

where X_i indicates an output of the i-th head and $W^O \in \mathbb{R}^{d \times d}$ is a weight matrix.

Figure 4.1 shows examples of self-attention weights in Transformer model. Given the query word "*representation*", the single-head attention module outputs attention

weights for other words (a). On the other hand, each head in multi-head attention assigns different weights for other words as each head has its own weight matrix (b).

Although it has been believed that multi-head attention diversifies representation subspaces, measuring the similarity among deep representations of each head has been rarely studied. Measuring the inter-head similarity requires taking account of heads' response over the entire dataset. To do so, I adopt the following advanced tools for measuring similarity of representations in neural networks.

4.2.2 Singular Vector Canonical Correlation Analysis (SVCCA)

To measure the similarity between two deep representations, Raghu et al. [10] have amalgamated Canonical Correlation Analysis (CCA) with Singular Value Decomposition (SVD) into a novel method, Singular Vector Canonical Correlation Analysis (SVCCA). Raghu et al. [10] has claimed that SVCCA is invariant to affine transform, hence it can measure the similarity between unaligned deep representations.

SVCCA proceeds in two steps to seek correlation coefficients between two deep representations with N samples X_i and $X_j \in \mathbb{R}^{N \times d}$: (1) SVCCA performs SVD of each representation to pick out core representations, then (2) computes CCA of the core representations. Resulting SVCCA coefficients ρ_{ij} are computed as follows:

$$\rho_{ij} = \max_{a,b} corr(a^T U_i X_i, b^T U_j X_j), \tag{4.3}$$

where U_i and U_j are left orthogonal matrices computed from SVD of X_i and X_j respectively. SVCCA similarity between two deep representations using SVCCA is defined as a mean value over top SVCCA coefficients with a threshold such that covers all meaningful subspaces. In this chapter, I measure inter-head similarity by averaging SVCCA similarity between two heads over all possible pairs of heads.

(**a**) Single-head attention

(**b**) Multi-head attention

Figure 4.1: Visualization of attention weights in single-head attention and multi-head attention. Each head in multi-head attention assigns different weights to each word.

4.2.3 Centered Kernel Alignment (CKA)

Kornblith et al. [21] have introduced Centered Kernel Alignment (CKA) as a similarity measure between deep representations. The authors have pointed out a limitation of SVCCA that it is invariant to invertible linear transformation when dimension size exceeds the number of data, whereas CKA shows robustness regardless of a small number of data N.

CKA is calculated by normalizing an index of Hilbert-Schmidt Independence Criterion (HSIC) [69] in order to keep invariance to isotropic scaling. For a pair of heads $X_i = (x_{i1}, x_{i2}..., x_{iN})^T$ and $X_j = (x_{j1}, x_{j2}..., x_{jN})^T$, I can define two matrices $K_{ikl} = \kappa(x_{ik}, x_{il})$ and $K_{jkl} = \kappa(x_{jk}, x_{jl})$ where κ is kernel function. Then HSIC between two heads is computed as follows:

$$\text{HSIC}(K_i, K_j) = \frac{1}{(N-1)^2} tr(K_i C K_j C), \tag{4.4}$$

where C is a centering matrix $C_N = I_N - \frac{1}{N}\mathbf{1}\mathbf{1}^T$, where $\mathbf{1}$ is a vector of ones. CKA of a pair of heads is computed by normalizing HSIC [70, 71]:

$$\text{CKA}(K_i, K_j) = \frac{\text{HSIC}(K_i, K_j)}{\sqrt{\text{HSIC}(K_i, K_i)\text{HSIC}(K_j, K_j)}}. \tag{4.5}$$

Finally, I define inter-head similarity using CKA as an average value over CKA of every possible pair of heads:

$$\text{CKA}_{\text{multi}} = \frac{1}{\#\text{ of } pairs} \sum_{i<j} \text{CKA}(K_i, K_j). \tag{4.6}$$

In this chapter, CKA similarity is used as not only a tool for analyzing inter-head diversity as well as SVCCA statistics but also an augmented loss to control inter-head diversity.

4.3 Proposed Approach: Controlling Inter-Head Diversity

In this section, I inspect three methods for multi-head attention to finely control inter-head diversity in training. The three methods are architecture-agnostic, task-agnostic,

and able to fine-tune so that they can be easily applied to any existing models with multi-head attention.

4.3.1 HSIC Regularizer

Because Kornblith et al. [21] have demonstrated that CKA robustly performs even with a small number of samples, I exploit it directly as an augmented loss term to enforce inter-head representations to be diverse. While SVCCA similarity is inappropriate for a regularizer term to be used in training because it requires many samples, CKA can properly operate within samples randomly drawn from a mini-batch. Since CKA is fully differentiable function and its gradient can be properly back-propagated through neural networks, I can directly use CKA as an additional loss term in training. As directly optimizing CKA loss, I expect representational subspaces of multi-head attention to be diverse.

To prevent high computational cost in training, I only compute HSIC term (Equation (4.4)) as an augmented loss. HSIC regularizer term L_{hsic} is computed by average of HSIC values with every possible pair of heads as follows:

$$L_{\text{hsic}} = \lambda_{hsic} \cdot \frac{1}{\#\text{ of } pairs} \sum_{i<j} \text{HSIC}(X_i, X_j), \tag{4.7}$$

where X_i is a representation of the i-th head. HSIC is zero when two variables are independent, hence I expect that HSIC regularizer increases inter-head diversity by minimizing L_{hsic} in training.

4.3.2 Orthogonality Regularizer

I also revisit the orthogonality loss [55] which adds disagreement term on between heads' representations. The disagreement term can be interpreted as a weak orthogonal constraint term since it is computed by cosine similarity between V_{li} and V_{lj}, where V_{li} is the l-th vector in the i-th head. Therefore, the disagreement term orthogonalizes

an orientation through minimizing the cosine similarity. I apply the disagreement term to q, K, and V in my model, assuming that it can give variation to inter-head diversity with SVCCA and CKA.

In line with orthogonality regularization, Bansal et al. [67] have suggested Spectral Restricted Isometry Property (SRIP) regularization as a stricter orthogonal constraint. SRIP regularizer minimizes a spectral norm of orthogonality to its target matrix more strictly because the spectral norm requires all singular values of its target matrix to be close to 1. Thus, by utilizing both SRIP regularizer and the disagreement regularizer, I suggest an orthogonality regularizer for multi-head attention as a tool for controlling inter-head diversity. The orthogonality term L_{ortho} is computed as follows. I first build V_{all} by collecting every l-th vector of *value* matrix V in every i-th head, $V_{all} = [V_0, ..., V_i, ..., V_H]$. Then, I take SRIP of V_{all}:

$$L_{\text{ortho}} = \lambda_{ortho} \cdot \sigma(V_{all}^T V_{all} - I), \tag{4.8}$$

where $\sigma(W)$ is the spectral norm of W.

Surprisingly, although Li et al. [55] has claimed that the disagreement regularizer encourages inter-head diversity, I find it slightly decreases inter-head diversity measured with SVCCA and CKA. However, instead of encouraging inter-head diversity, we observe that the orthogonality regularizer increases top-few SVCCA coefficients that can be regarded as core representations. I report detailed results and discussion in Section 4.5.

4.3.3 Drophead

I also inspect Drophead [58] as the very naive but effective method to control the diversity. Zhou et al. [58] have introduced Drophead as a regularizing method in order to reduce overfitting similar to Dropout [59]. Zhou et al. have introduced Drophead as a method that drops an entire attention head during training and shown that the Drophead improves the model's robustness and performance with carefully scheduled

dropout rate. Unlike Zhou et al., I mainly focus on how Drophead controls and diversifies the inter-head similarity. I use more naive Drophead method that randomly zero-out each head in training with a dropout rate γ, a real value ranged from 0.0 to 1.0. Drophead only requires a scalar hyperparameter γ while a model can keep its architecture identical. Also, Drophead can be applied to training without additional computational cost.

Drophead reduces the *effective* number of heads by randomly dropping it out in training, hence it operates similarly to *decreasing number of heads* in training and decreases inter-head diversity. Simultaneously, applying Drophead can benefit the advantages of Dropout as well as Zhou et al. have shown. In the experiments, Drophead is applied independently to Dropout.

4.4 Inter-Head Similarity Analyses

In this section, I investigate how SVCCA and CKA values change with respect to the number of heads. By analyzing the diversity of representation subspaces, I show that how SVCCA and CKA reflect the dynamics of inter-head similarity in terms of the numbers of heads.

4.4.1 Experimental Details for Similarity Analysis

- **Data and Setups:** I choose De→En IWSLT17 machine translation task [60] for my analysis in this section. Training set consists of 223,162 sentences, development set consists of 8130 sentences, and test set consists of 1116 sentences. To tokenize the corpus, I use Byte Pair Encoding [22] with a vocabulary size of 16,384. I use Transformer [3] architectures with various numbers of heads and hidden dimension sizes for comparison. For all models, I use 6 layers for encoder's self-attention, decoder's self-attention, and encoder-decoder attention modules.

- **Performances of trained models:** Table 4.1 shows BLEU scores of models with various hidden dimension sizes and numbers of heads. As represented in Table 4.1, increasing hidden size d results in higher BLEU performances with a fixed number of heads, although increasing the number of heads does not always assure higher performance with fixed hidden size.

4.4.2 Applying SVCCA and CKA

In order to verify whether the multi-head strategy affects models' representation subspaces, I examine SVCCA statistics between representations of heads in each model. To utilize SVCCA and CKA, I collect responses $X = [X^1, ..., X^N]$ of each head at the last layers of three modules (encoder's self-attention, decoder's self-attention, and encoder-decoder attention) from development dataset consisting of *num_sentence* sentences, so that I have $N =$ *num_sentence* $\times$ *token_per_sentence* many d-dimensional vectors. We compare nine models with a number of heads $h = \{2, 4, 8, 16\}$ and hidden size $d = \{64, 128, 256, 512\}$ in order to examine how those architectural parameters change inter-head diversity. We report the results of the last layer of the encoder-decoder attention module only, yet I find the same tendency through every layer of every module.

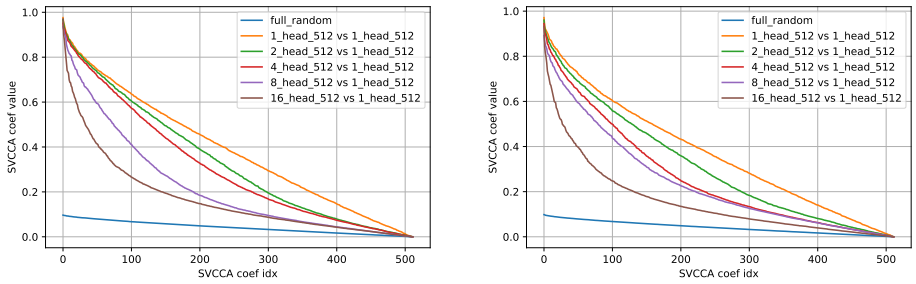
4.4.3 Analyses on Inter-Model Similarity

I first examine SVCCA statistics of representations of five models versus representation of a single-headed model. I compare five models with varying numbers of heads ($H = 1, 2, 4, 8$, and 16) and fixed hidden size d as 512.

As shown in Table 4.2, SVCCA similarities between multi-headed models and a single-headed model, I can see that the response of a model is getting more dissimilar to a single-headed model as the number of heads increases. SVCCA coefficient curves also show similar results in Figure 4.2. SVCCA coefficients drop more rapidly with

Table 4.1: BLEU scores comparison with various hidden size d and *number of head* H on IWSLT17 De→En corpus.

Hidden Size d	***Number of Heads H***				
	1	**2**	**4**	**8**	**16**
64	26.33	27.47			
128	31.30	32.75	31.71		
256	32.63	33.23	33.42	33.62	
512	33.33	33.42	33.90	33.67	32.69



(**a**) Encoder's self-attention (**b**) Encoder-Decoder attention

Figure 4.2: Singular Vector Canonical Correlation Analysis (SVCCA) coefficient curves versus a single headed model.

large number of heads in every layer. These results indicate that multi-head strategy can induce a model to find some representations uncorrelated to a single-headed model while its core representations remain, as shown as top few SVCCA coefficients are high.

4.4.4 Does Multi-Head Strategy Diversify a Model's Representation Subspaces?

Table 4.3 shows inter-head similarity using SVCCA and CKA of each model. Both inter-head similarity measures using SVCCA and CKA show a persistent tendency that the inter-head similarity of each model decreases as the number of heads increases. On the other hand, I observe that increasing hidden dimension size d does not meaningfully affect the inter-head similarity with a fixed number of heads.

In addition to Table 4.3, I plot SVCCA coefficient curves of inter-head similarity in Figure 4.3. With various number of heads $H = \{2, 4, 8, 16\}$ and fixed $dim/head = \{32, 64\}$ ((a) and (b) in Figure 4.3), I observe that increasing number of heads make SVCCA coefficients smaller, indicating that inter-head diversity also increases. I also observe the same tendency with fixed dim ((c) in Figure 4.3), while I cannot find any consistency of inter-head similarity with fixed number of heads ((d) in Figure 4.3).

Table 4.2: SVCCA similarities versus a single headed model.

Modules	***Number of Heads H***				
	1	**2**	**4**	**8**	**16**
encoder's self-attention	0.448	0.407	0.374	0.272	0.202
decoder's self-attention	0.474	0.4	0.353	0.328	0.237
enc-dec attention	0.429	0.378	0.319	0.289	0.189

Table 4.3: Inter-head similarity with various numbers of heads and hidden dimension.

Models	***Dim/Head***	**SVCCA**	**CKA**
2_H_64_d	32	0.793	0.553
2_H_128_d	64	0.712	0.488
2_H_512_d	256	0.559	0.344
4_H_128_d	32	0.504	0.277
4_H_256_d	64	0.541	0.309
4_H_512_d	128	0.560	0.277
8_H_256_d	32	0.346	0.143
8_H_512_d	64	0.419	0.197
16_H_512_d	32	0.252	0.117

Besides, I observed an interesting feature of SVCCA similarity curves that well-trained models have steep slopes on top-few SVCCA coefficients. I later discuss the steepness of top-few SVCCA coefficients in Section 4.5. My analysis of inter-head similarity measured by SVCCA and CKA statistically support the hypothesis that multi-head attention diversifies deep representations.

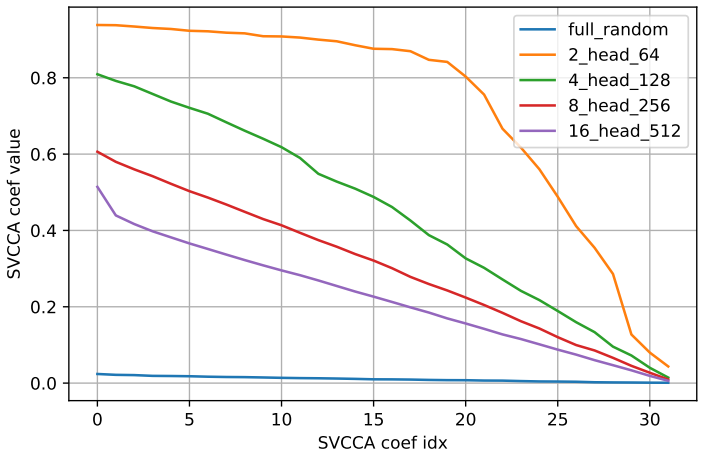
4.5 Experiments on Controlling Inter-Head Similarity Methods

To examine how the three methods affect multi-head attention, I analyze inter-head similarity statistics on De→En machine translation task with IWSLT17 corpus. I also report the experimental results through extensive experiments on machine translation and language modeling tasks to empirically verify that the three methods can make a model achieve higher performance than its baseline model.

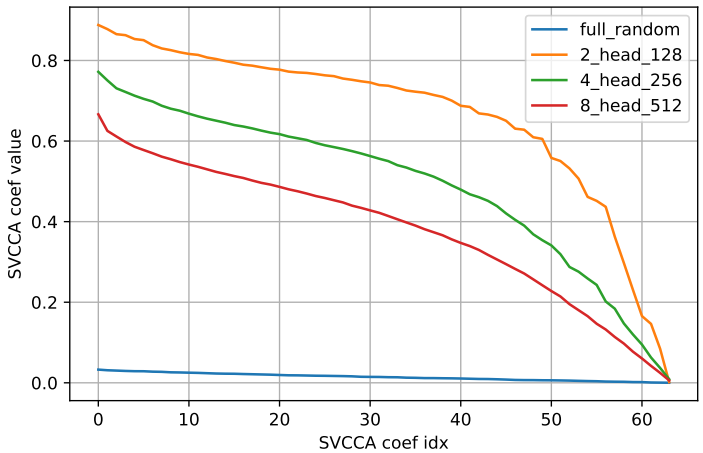
4.5.1 Experimental Details

- **Data and Setups:** I test the proposed methods on machine translation tasks with De-En WMT17 corpus [72], Ru-En UN corpus, and Zh-En UN corpus [61]. For WMT17 and UN corpus, I sample 2.5 M sentences randomly from each training set for training and use the whole development/test sets, similar to the setup of Voita et al. [9]. Each corpus has development set consisting of 16,573 and 4000 sentences respectively and test set consisting of 3004 and 4000 sentences respectively. I also test the methods on a language modeling task with the Penn Treebank corpus [73]. I follow the rest of details as mentioned in Section 4.4

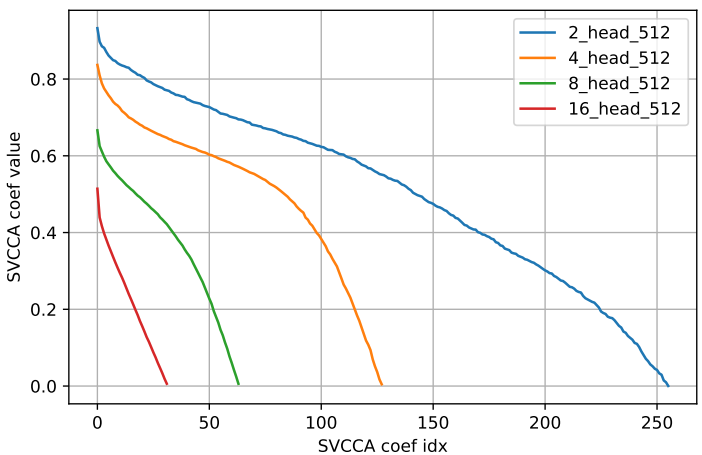
- **Model architectures:** I set a baseline model as an encoder-decoder Transformer with 6 layers, 512 hidden size, and 8 heads for every machine translation task. For language modeling, I use only the decoder part of Transformer only with



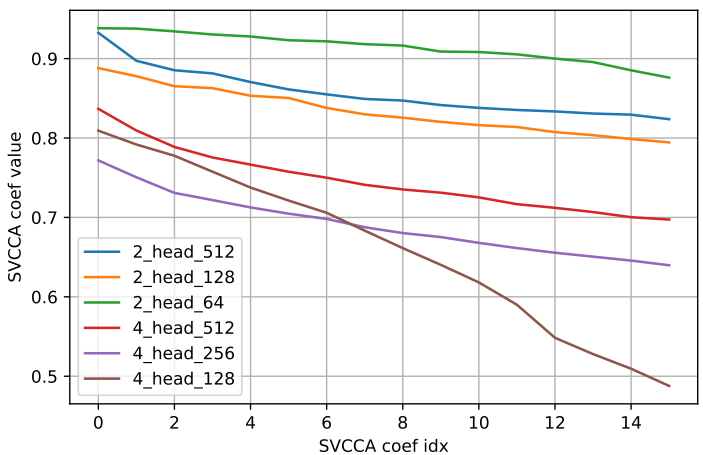
(**a**) Models with fixed *dim / head* 32



(**b**) Models with fixed *dim / head* 64



(**c**) Models with fixed *dim* 512



(**d**) Top 16 SVCCA coefficients with $H = \{2, 4\}$

Figure 4.3: SVCCA coefficient curves of inter-head similarity.

2 layers, 256 hidden sizes, and 4 heads. For each model named with ORTHO and HSIC, I add each regularization term L_{ortho} and L_{hsic} to Transformer's default loss term. I choose the value of hyperparameters *Drophead rate*, λ_{ortho} and $lambda_{hsic}$ by grid search on De→En IWSLT17 task; *Drophead rate*$= 0.1$, $\lambda_{ortho} = 1.0$, and $\lambda_{hsic} = 10^{-7}$. I apply the same values for other models.

4.5.2 Analysis on Controlling Inter-Head Diversity

I report the performances of the suggested methods in Table 4.4 and the controlled inter-head similarity with the suggested methods in Table 4.5. I also plot SVCCA coefficient curves in Figure 4.4.

With Drophead, all models show increased inter-head similarity compared to the baseline. As γ increases to 0.0 to 0.5, inter-head similarity indeed increases to 0.397 to 0.709, indicating that Drophead affects inter-head similarity by reducing the number of *effective* heads as desired. I observe this clear tendency by comparing SVCCA coefficient curves (a) in Figure 4.4 to (b) in Figure 4.3. The curve of $8\_H\_512\_d$ with $\gamma = 0.3$ is very similar to that of $4\_H\_256\_d$, and as the rate increases $\gamma = 0.5$, the curve becomes similar to that of the model with fewer heads $2\_H\_128\_d$.

In addition, as opposed to the expectation of Li et al. [55] have expected, I find that the orthogonality loss does not diversify inter-head similarity. For +ORTHO and +HSIC, every model shows average disagreement score [55] as 0.999, which implies that two vectors from different heads are orthogonal. However, instead of diversifying, the orthogonality loss slightly increases inter-head similarity measured in both SVCCA (from 0.397 to 0.420) and CKA (from 0.199 to 0.366). Nevertheless, the model only with the orthogonality loss performs better than a baseline as it records 34.03 BLEU score (+ORTHO ONLY in Table 4.4). I suspect that the performance improvements are caused by steep rises of top-few SVCCA coefficients. The affects of the orthogonality loss on top-few SVCCA coefficients are depicted in (b) and (d) in Figure 4.4 (as comparing curves of *baseline, ortho 0.1, ortho 1.0, and ortho 10.0*).

Table 4.4: BLEU evaluation with controlled inter-head similarity on En-De IWSLT17 corpus.

Models	**Language Pairs**	
	De→En	**En→De**
Baseline Transformer	33.67	29.76
+ drophead only	34.26	30.13
+ ortho only	34.03	30.27
+ HSIC only	34.43	30.32
+ALL	**34.53**	**30.38**

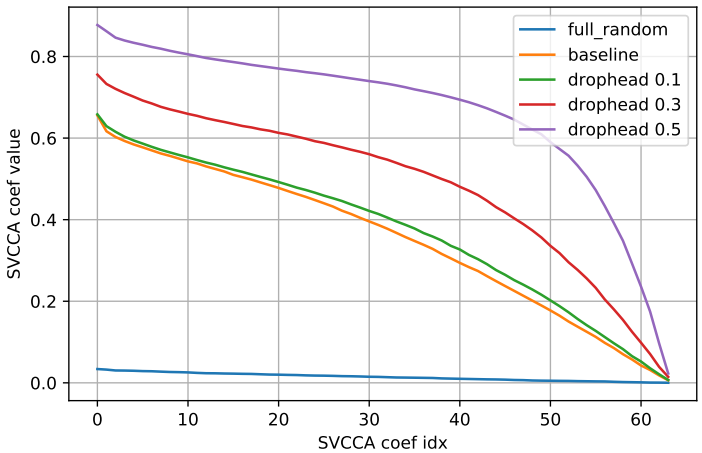
The orthogonality regularizer makes the heads similar to each other in a prime direction while sustaining other directions diverse, hence it makes the model robust to both general features and rare features.

Lastly, I observe that HSIC regularizer directly enforces each head to be diverse as shown in both Table 4.5 and (c) in Figure 4.4. While the other two methods increase inter-head similarity, HSIC regularizer is the only method to diversify inter-head similarity without modifying a model's architecture. Although *increasing number of heads* H also diversify inter-head similarity, it has a critical downside that architectural modification must be accompanied.

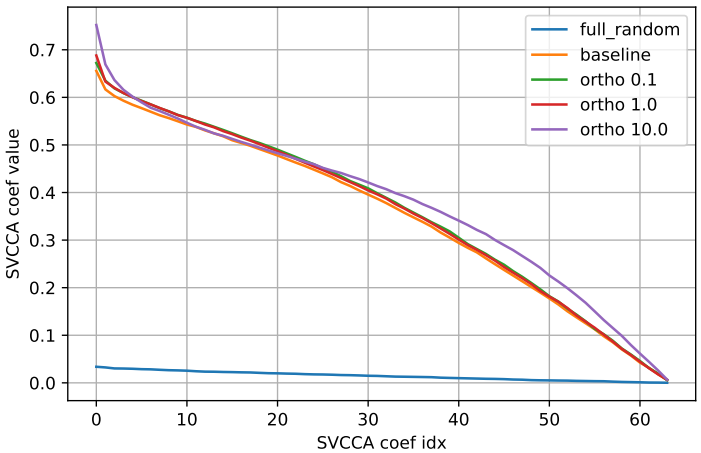
4.5.3 Quantitative Evaluation

I report BLEU scores on every language pairs in Tables 4.4 and 4.6. These results support my hypothesis that a multi-head attention model can extend its own capability by controlling inter-head diversity with the suggested methods. Models with all three suggested methods applied (+ALL) show the best performances on every language pair.

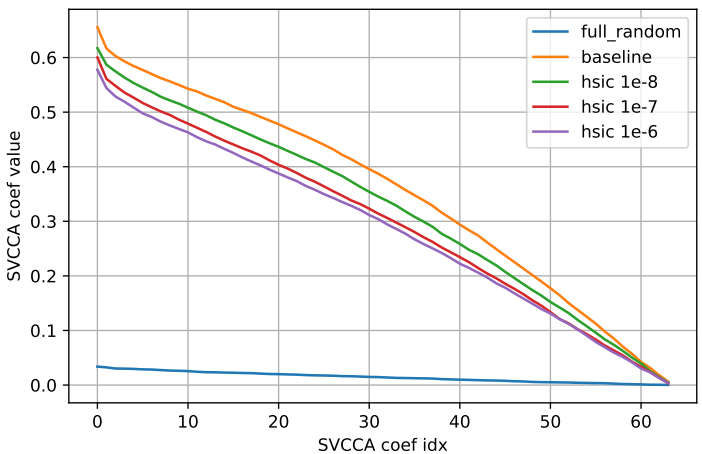
I also verify the effect of the suggested methods on language modeling task in order to show that the methods can be applied to tasks other than machine translation.



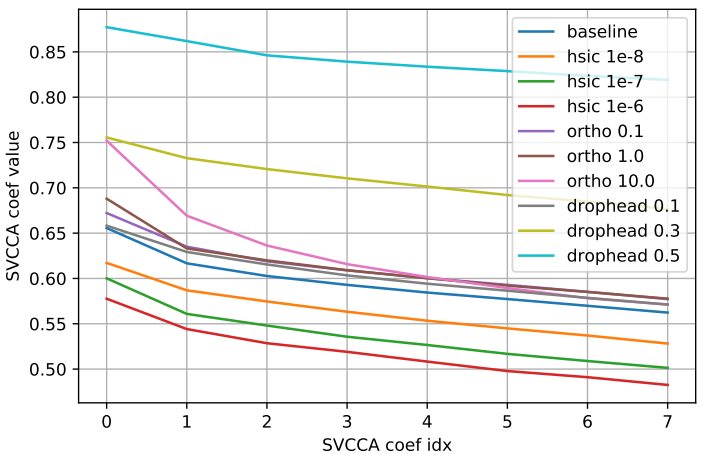
(**a**) Drophead with various rate



(**b**) Orthogonality loss with various λ_{ortho}



(**c**) HSIC loss with various λ_{hsic}



(**d**) Top 8 SVCCA coefficients

Figure 4.4: SVCCA coefficient curves of inter-head similarity with controlling methods.

Table 4.5: Controlled inter-head similarity with suggested methods.

Models		**SVCCA**	**CKA**
Baseline Transformer		0.397	0.199
+ drophead	*0.1*	0.415	0.207
+ drophead	*0.3*	0.534	0.317
+ drophead	*0.5*	0.709	0.527
+ ortho	*0.1*	0.408	0.208
+ ortho	*1.0*	0.407	0.223
+ ortho	*10.0*	0.420	0.366
+ HSIC	10^{-8}	0.364	0.182
+ HSIC	10^{-7}	0.338	0.158
+ HSIC	10^{-6}	0.325	0.125

Table 4.7 shows perplexity score on language modeling task with PTB corpus. As well as on the encoder-decoder Transformer, the methods applied to the decoder-only Transformer also increases its performance on the language modeling task. Applying + HSIC only shows the best performance, even better than applying all methods. Nevertheless, all of the methods clearly improve the perplexity of the decoder-only Transformer. The experimental results show that the methods can easily be applied to various model architectures that use multi-head attention. Note that the suggested methods and my analyses in Section 4.4 do not relate to the size of the model (i.e., the hidden size or the number of layers). I strongly believe that the methods can be applied to larger language models such as BERT [4] or XLM-R [15], because they also exploit the multi-head attention as the same way as the Transformer model in the experiments.

4.5.4 Limitations and Future Directions

In this chapter, I show that the multi-head attention increases the inter-head diversity and the proposed methods can control the inter-head diversity in order to yield bet-

Table 4.6: BLEU evaluation on various language pairs with controlled inter-head similarity on WMT17 corpus and UN corpus.

Models	**Language Pairs**					
	De→En	**En→De**	**Ru→En**	**En→Ru**	**Zh→En**	**En→Zh**
Baseline Transformer	31.47	25.69	52.68	44.62	52.21	47.08
+ Drophead only	31.69	25.73	52.90	44.82	52.60	47.09
+ Ortho only	31.63	25.86	52.92	44.86	52.55	47.28
+ HSIC only	31.57	25.89	52.89	44.82	52.54	47.16
+ALL	**31.76**	**25.91**	**53.02**	**45.23**	**52.69**	**47.33**

Table 4.7: Perplexity with controlled inter-head similarity on PTB language modeling.

Models	Perplexity
Baseline Transformer	120.38
+ drophead only	102.72
+ ortho only	102.62
+ HSIC only	**101.89**
+ALL	102.07

ter performances. Nevertheless, the improvements of BLEU are slightly larger than marginal increment which undermine the benefits of optimizing the inter-head diversity. Therefore, I recommend the future directions of this chapter as follows.

- **CKA optimization for Multi-modality**: Optimizing the inter-head diversity of the head representing the same language may have smaller impact to the final performance. However, in chapter 5 of this dissertation, I observe that the optimization technique for CKA show significant improvements on multi-modal tasks. Because there is a large gap between representation spaces among various modalities, I believe that maximizing CKA between two representation spaces of different modalities can align two representation spaces and consequently aids the cross-modality attention. In the next chapter of this dissertation, I extend the CKA optimization method and apply it on the multi-modal video-question answering task.

- **CKA optimization for Few-shot Learning**: Because CKA compute the robust value despite of the lack of data, I suggest to optimize the inter-head statistics for few-shot learning tasks. With only a few samples, a multilingual language model or a machine translation model suffers a huge drop of performance. To address such limitations, in my future studies, I plan to adopt the optimizing methods on low-resource settings.

4.6 Conclusions

In this chapter, I analyze the inter-head similarity of multi-head attention using SVCCA and CKA to unveil representation of each heads' subspaces. I show an empirical proof that multi-head attention diversifies its representations as the number of heads increases. Based on my observation, I hypothesize that there is an optimal degree of inter-head diversity that fully utilizes a model's capability. Then, I introduce three methods to control the degree of inter-head diversity; (1) HSIC regularizer, (2) the orthogonality regularizer revisited, and (3) Drophead method. The three methods are all able to fine-tune the inter-head diversity without architectural change. I show that HSIC regularizer diversifies the inter-head diversity and Drophead works the other way, whereas the orthogonality regularizer gathers the core representations of multi-head attention. Finally, we empirically show that controlling inter-head diversity can make the model utilize its own capability better resulting in higher performances on various machine translation and language modeling tasks. The three methods to control inter-head diversity can be easily applied to every model that uses multi-head attention including Transformer, BERT, and XLNet.

Chapter 5

Modality Alignment for Cross-modal Attention

For deep learning researchers, multi-modality recently became an important keyword as multi-modal models have shown the ability to collate plentiful information scattered over various modalities [74, 75, 76]. In particular, Video-and-Language learning which includes both video modality and text modality is attracting a huge attention [2, 77, 78, 79, 80, 81]. Specifically, Video-and-Language learning such as video captioning or video question answering require the ability of reasoning over both time and multiple modality. For example, a video question answering model should be able to find appropriate visual information in a video frame sequence with a given question. That is to say, capturing the relationship between video information and text information is important for a multi-modal model for Video-and-Language learning.

Cross modality attention module which combines correlation over different modalities becomes a critical component for Video-and-Language learning [82, 83, 84].Generally, the attention mechanism induces a model to learn the most important representation among the whole sequence with a given query. For single modality models, the attention module finds crucial parts to concentrate, greatly improves the performance of the model [3]. However, the cross-modal attention mechanism in multi-modal models is less effective than in single modality models because of the noticeable differences characteristics between multiple modalities. Existing Video-and-Language models do

not take this into account and merely utilize the attention mechanism as the same way as in single modality models, which hinders the models to fully enjoy the strength of the attention mechanism.

In this chapter, I propose a novel Modality Alignment method that optimizes the alignment between representation structures of the video modality and the text modality. My method leverages Centered Kernel Alignment (CKA) as an auxiliary objective to be maximized. As training the auxiliary loss via gradient descent frameworks, the embedding representation structures of both modalities are also trained to be similar. Therefore, Modality Alignment method enhances the cross modality attention module inside a multi-modal model to be more aware of correlated information, eventually improving the final performance.

CKA was originally designed to measure similarity between neural networks representations [71]. Recently, [21] discovered the robustness of CKA, which comes from the invariance to orthogonal transformations and isotropic scaling. In this work, I reveal another desirable property of CKA that can be directly optimized through gradient descent frameworks. With the robustness and trainability of CKA, I utilize CKA in order to align multi-modal representations. As far as I know, this is the first attempt to exploit CKA as a training objective in handling multi-modality. Also, Modality Alignment method can be easily applied to various multi-modal tasks.

I validate the proposed method through various experiments with both synthetic dataset and real-world dataset. Firstly, I show that aligning the embedding representations through maximizng CKA can effectively boost the performance on cosine similarity learning, which is a basis of the attention mechanism. Then, in the real world Video QA task, I empirically demonstrate that out method makes a multi-modal model to effectively learn the cross-modal attention. For TVQA [1] and TVQA+ [2], which are challenging benchmarks in Video QA, the models applied with the method outperforms the baseline models.

Namely, my contributions are listed as followings:

- I show that Centered Kernel Alignment, a similarity measurement between neural network representations, can be exploited to align two embedding representations from different modalities.
- I demonstrate that Modality Alignment method which optimizes the similarity between embedding representations is helpful for the cross-modal attention.
- I examine that Modality Alignment method, which can be easily applied to existing models, improves the performance in various multi-modal tasks through extensive experiments.

5.1 Related Works

5.1.1 Representation Similarity between Modalities

Several works have attempted to analyze the similarity between representation in neural networks to achieve interpretability. The most fundamental measurements that can be used with this neural network similarity are correlation and Canonical Correlation Analysis (CCA) [85].

An alignment method using the correlation of neuron responses has been proposed to share core representations between different networks [12]. Similarly, Singular Vector Canonical Correlation Analysis (SVCCA) [10] has been introduced in order to pick out perturbing directions from representations with applying CCA as a similarity measure. [63] subsequently have proposed Projection Weighted CCA (PWCCA) which is more reflective to subspaces of representations via projection. More recently, [21] have shown that Centered Kernel Alignment (CKA) is a appropriate measure for representation similarity because CKA is robust to the lack of data.

Also, there have been studies that use these similarity measures between neural networks directly or indirectly in deep representation learning. By applying CCA, SVCCA, and CKA, [64] have discovered that the geometry of Recurrent Neural Net-

work (RNN) architecture varies by task while the underlying scaffold is universal. On multilingual machine translation task, [65] have leveraged SVCCA across languages to show that there are shared representations among language representations. [57] have applied SVCCA to identify meaningful directions in machine translation and concluded that the top-few directions of SVCCA similarity indicates a key representation.

Unlikely, I propose a method that directly optimizes CKA between multi-modal representation structures to be maximized. The robustness in CKA enables the method in the way that CKA is reliable even in a mini-batch where the number of data is small.

5.1.2 Video Question Answering

Video-and-Language learning requires fine-grained interaction with information from multiple modalities. To study the fusion of visual modality and text modality, Image QA task which takes a single image input with a question in natural language has attracted the attention of many researchers [75, 86, 87]. However, unlike single image processing, video information is made up of a large number of image frames in a sequence, which is much larger and includes additional temporal information.

To date, the *de facto* way to solve the Video QA task is to fuse and learn both modality information using cross-modal attention after processing the video input and text input respectively. The video processing part has been developed based on existing video analysis schemes, such as recurrent networks of frame functions [88] or 3D convolution operators for action recognition [89]. Video representation is then fused via a co-attention module with textual input as query [90, 91], a hierarchical attention [92, 93], or a memory networks module [94, 95]. These methods have applied their novel methods on how to fuse two modality information well, but they all merely combine multi-modality information without considering the differences in modality characteristics.

I observe that there is a significant difference in characteristics between the two

modalities which may aggravate the cross modality attention. Thus, Modality Alignment method increases the similarity between multi-modality representation structures to enhance the fusion more effective. I validate the proposed method for synthetic dataset first, and then apply it to a real-world VideoQA dataset which has significant differences in characteristics between the two modalities.

5.2 Proposed Approach: Modality Align between Multi-modal Representations

With a new use of CKA as a learning objective, I propose a novel Modality Alignment method that directly maximizes CKA to align representation structures between various modalities.

5.2.1 Centered Kernel Alignment Review

As a tool to measure similarity between two deep representations, Centerend Kernel Alignment (CKA) has been proposed [71, 70]. Recently, [21] bring CKA back to the surface, addressing that CKA can aid in gaining a deep understanding of internal neural network architectures.

CKA is obtained by normalizing Hilbert-Schmidt Independence Criterion (HSIC) [69]. For a pair of neural network representations $X_i = (x_{i1}, x_{i2}..., x_{iN})^T$ and $X_j = (x_{j1}, x_{j2}..., x_{jN})^T$, I define two matrices $K_{ikl} = \kappa(x_{ik}, x_{il})$ and $K_{jkl} = \kappa(x_{jk}, x_{jl})$ where κ is kernel function and N is a number of sampled data from each representation. Then, HSIC between two representations is computed as follows:

$$\text{HSIC}(K_i, K_j) = \frac{1}{(N-1)^2} tr(K_i C K_j C), \tag{5.1}$$

where C is a centering matrix $C = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$ ($\mathbf{1}$ is a vector of ones and N is the number of sampled data). For linear kernels (e.g. $\kappa(x, y) = x^T y$), HSIC computes the

squared Frobenius norm of the cross-covariance:

$$\left\|\text{cov}(X_i^T, X_j^T)\right\|_F^2 = \frac{1}{(n-1)^2} tr(X_i X_i^T X_j X_j^T). \tag{5.2}$$

Thus, HSIC can be interpreted as the similarity between the inter-example similarity structures. Normalizing HSIC results CKA as follows:

$$\text{CKA}(K_i, K_j) = \frac{\text{HSIC}(K_i, K_j)}{\sqrt{\text{HSIC}(K_i, K_i)\text{HSIC}(K_j, K_j)}}. \tag{5.3}$$

The normalizing process makes the output value of CKA between 0 and 1 where $\text{CKA}(X, Y) = 0$ implies independence. Also, this process makes CKA invariant to isotropic scaling.

5.2.2 Why CKA is Proper to Modality Alignment

CKA exhibits desirable properties for not only measuring similarity between two deep representations but also training the alignment of inter-example similarity structures with gradient descent. I list three properties that enable the methodology for the modality alignment.

- **Invariance to orthogonal transformations:** [21] especially pointed out that CKA is invariant to orthogonal transformations of deep representation, i.e. $\text{CKA}(X, Y) = \text{CKA}(XU, YV)$ for any orthonormal matrices U and V. Because neural networks are randomly initialized and trained by gradient descent with random mini-batches, there is a high probability that neurons are permuted even in the same networks. Therefore, invariance to orthogonal transformations, which includes permutations, is one of the essential characteristics required for the similarity indexes.

  Although other similarities such as CCA [85] or SVCCA [10] are invariant to affine transformations, [21] spotted the limitation of invariance to affine transformations that it requires more examples than the size of dimension to robustly measure the similarity between representations. This limitation makes CCA and

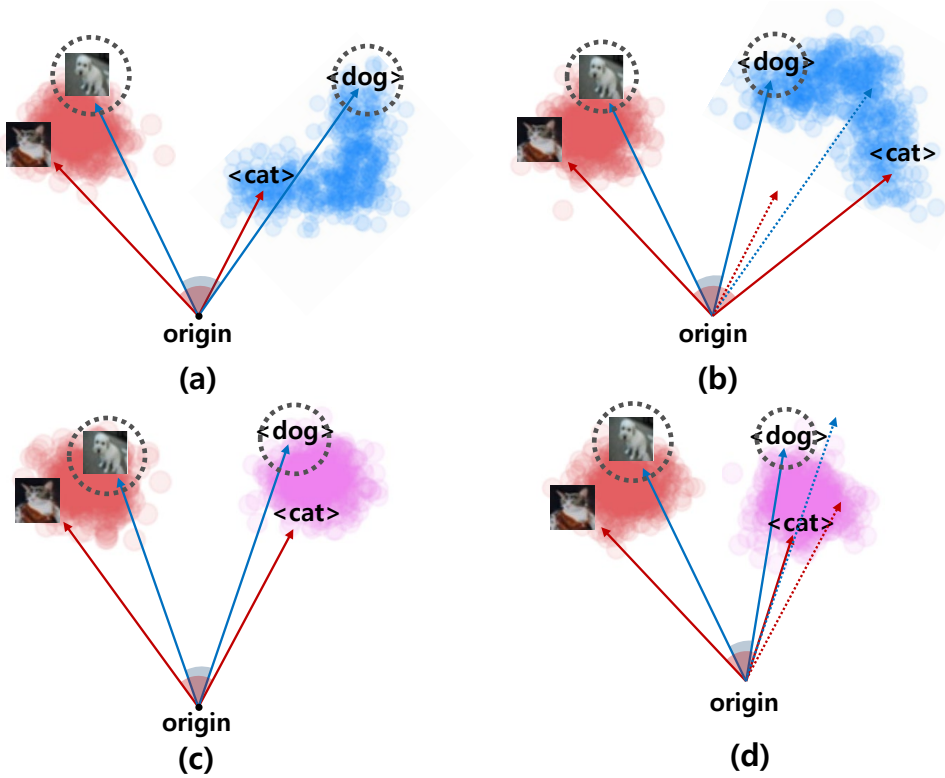


Figure 5.1: Main concept of Modality Alignment. *(a)*: During training cross modal attention module with a given mini-batch (inside the dotted circle), the model is trained to increase the attention score based on cosine similarity between the vector of "dog" and the correlated video frame vector. *(b)*: After a training step, the model is updated to narrow the gap. However, because the inter-example similarity structures are differently formed, there is potential harm to examples outside of the mini-batch; the cosine similarity between the "cat" vector and the correlated video vector decreases. *(c)* and *(d)*: Modality Alignment method keep the inter-example similarity structures to be close to each other, significantly reducing such adverse effects.

SVCCA unsuitable for training objective where the number of examples in a mini-batch is usually smaller than the size of dimension. However, unlike CCA or SVCCA, CKA shows robustness even with a small number of data (e.g. in a mini-batch).

- **Invariance to isotropic scaling:** CKA is also invariant to isotropic scaling, i.e. $\text{CKA}(X, Y) = \text{CKA}(\alpha X, \beta Y)$ for any $\alpha, \beta \in \mathbb{R}^+$. Invariance to isotropic scaling implies that CKA value remains the same even if each representation is scaled respectively, which often happens in neural networks training. [21] also mentioned that invariance to non-isotropic scaling is not a desired property because a similarity index that is invariant to both orthogonal transformations and non-isotropic scaling is invariant to any invertible linear transformation, which lacks the robustness.

- **Trainabilty via gradient descent methods:** As CKA is calculated with fully differentiable operations such as dot-product, CKA itself is also differentiable with respect to the parameters of neural networks. That said, CKA itself can be used as a training objective for gradient descent algorithms. [96] reported that CKA between representations of different layers in a model can be minimized or maximized via stochastic gradient descent. I exploit the trainability of CKA in order to align each representation in different modalities.

Using above three properties of CKA, I set CKA between video representation and text representation as an auxiliary training objective to be maximized. Note that maximizing CKA does not assure two representations to be overlapped. Nevertheless, it urges the inter-example structures of the two representations to be similar. For example, maximizing CKA between video representation and text representation makes the cosine similarity between a word “dog” and a word “cat” close to the similarity between a video frame with a dog and a video frame with a cat.

Meanwhile, the cross modality attention module computes its attention score based

on cosine similarity between two vectors. In other words, the cosine similarity between a frame-level video embedding vector and a word-level text embedding vector becomes higher as the cross modality attention module is optimized, if there is semantic correlation between the video frame and the word. Maximizing CKA can enhance the training of the attention module since the inter-example structures of two different modalities are kept aligned.

Figure 5.1 further depicts the main concept of Modality Alignment method. The objective of a cross attention module is to narrow the gap between two vectors with semantic correlation. Suppose fitting a cross attention module with a mini-batch which includes a word "dog" and a video frame with the dog (*(a)* of figure 5.1). After one training step, the parameters of networks are updated to narrow the angle between "dog" and the frame with the dog (*(b)* of figure 5.1). However, the cosine similarity between a word "cat" and a video frame with the cat decreases after the training step due to the difference of inter-example similarity structures. On the other side, with Modality Alignment, such adverse effects are significantly reduced because the inter-example structures are also trained to be similar (*(c)* and *(d)* of figure 5.1). Hence, maximizing CKA between multi-modal representations can boost the training of cross modality attention, resulting fast convergence and higher performance.

5.2.3 Proposed Method

The proposed Modality Alignment method computes CKA between the output representation of the video embedding module and that of the text embedding module in each mini-batch and directly maximizes it as an auxiliary objective.

In Video-and-Language learning, a model usually consists of a video embedding module, a text embedding module, and a video-text fusion module. Let $V = [v_1, ..., v_L]$ be a sequence of video frames v_i and $T = [t_1, ..., t_M]$ be a sequence of word tokens t_j. A video embedding module f_{vid} encodes the sequence of video frames into the video embedding representation: $f_{vid}(V) = X$, where $X \in \mathbb{R}^{L \times d}$ is a sequence of embed-

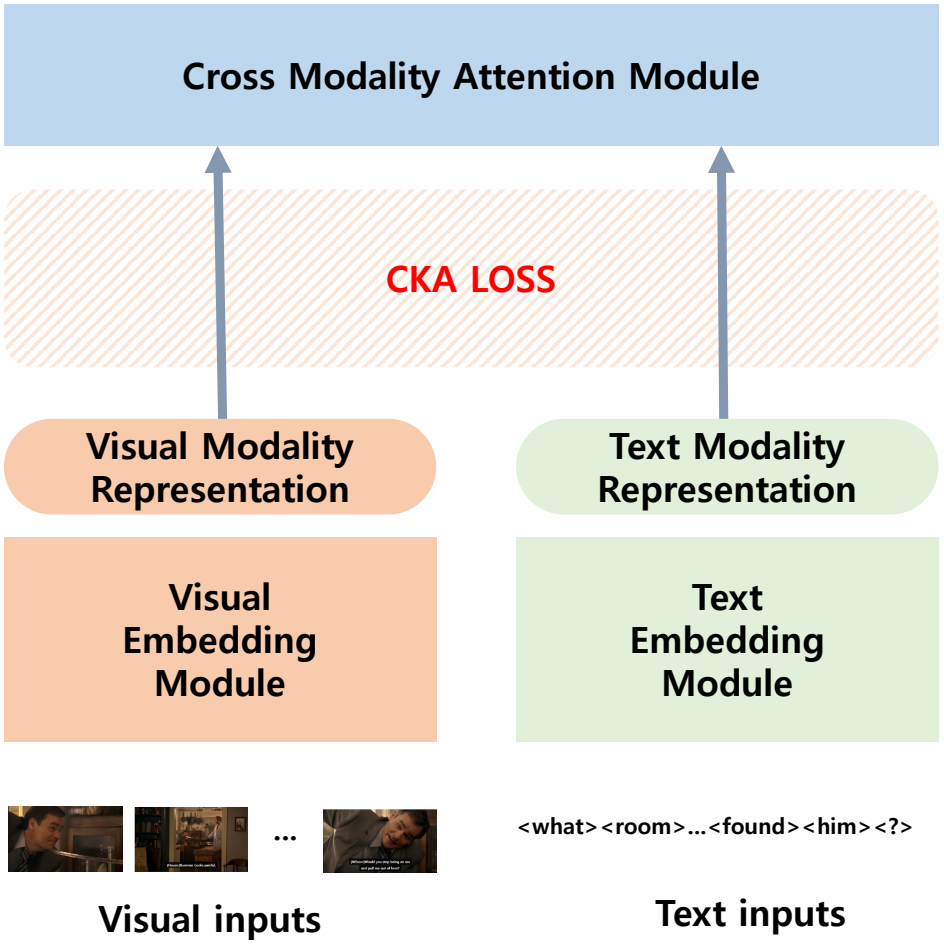


Figure 5.2: My proposed method. The input of each modality is embedded into the representation vector through each encoder module. CKA between representation vectors with different modalities is directly maximized to align the inter-example structure of each representation.

ded video representation vectors with dimension size of d. Similarly, a text embedding module f_{text} encodes the sequence of tokens into the text embedding representation: $f_{text}(T) = Y$, where $Y \in \mathbb{R}^{M \times d}$ is a sequence of text representation vectors. I randomly sample N-many representation vectors from both video representation tensor X and text representation tensor Y in order to match the number of examples. Finally, the modality alignment s between two representations is measured with equation (5.3).

I directly maximize the CKA as an auxiliary objective of the original loss to align two representations. Specifically, with a scaling hyperparameter λ_{cka}, I construct the final loss objective for minimizing by subtracting CKA loss term $\mathcal{L}_{cka}$ to the original loss term $\mathcal{L}_{orig}$ as follows:

$$\mathcal{L}_{final} = \mathcal{L}_{orig} - \lambda_{cka} * \mathcal{L}_{cka}. \tag{5.4}$$

Thus, the method can be applied to any model that handles multi-modality with cross attention module based on cosine similarity. I search the appropriate value of λ_{cka} by grid-searching in each experiment.

One can interpret Modality Alignment method as a new variant of contrastive learning since the method takes account of relationships between data examples within a mini-batch. The method has a novel strength in the respect that it can optimize the whole representation structures at every training step because of the robustness of CKA, while most of contrastive learning methods maximize the gap between irrelevant examples only in a mini-batch [97, 98].

5.3 Experiments

I conduct two experiments to verify Modality Alignment method. I show that the auxiliary CKA loss term boosts cosine similarity learning with a synthetic dataset. Then, I manage the real-world experiment on Video QA task in which the method outperforms conventional baselines. All experiments demonstrate that Modality Alignment

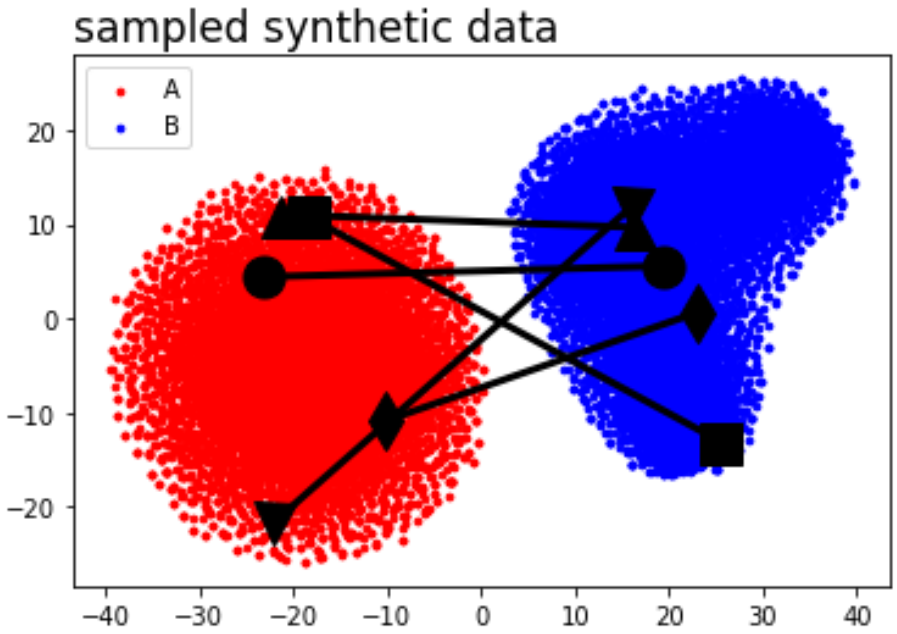


Figure 5.3: t-SNE visualization of my synthetic data distribution. I sampled two groups from different distributions and set one-to-one alignments to emulate hard attention.

method enhances the cross modal attention module, consequently resulting higher performance of the multi-modal model.

5.3.1 Cosine Similarity Learning with CKA

I empirically verify that optimizing CKA is helpful for cosine similarity learning. The attention mechanism learns cosine similarity between two corresponding source representation and target representation to be increased during training. However, because there are no ground truth attention weights in most real-world datasets, directly evaluating the performance of cross attention module is difficult. In order to verify that the method improves the performance of cross attention module, I conduct an experiment with a synthetic dataset in which a model is trained to maximize cosine similarity with one-to-one correspondence.

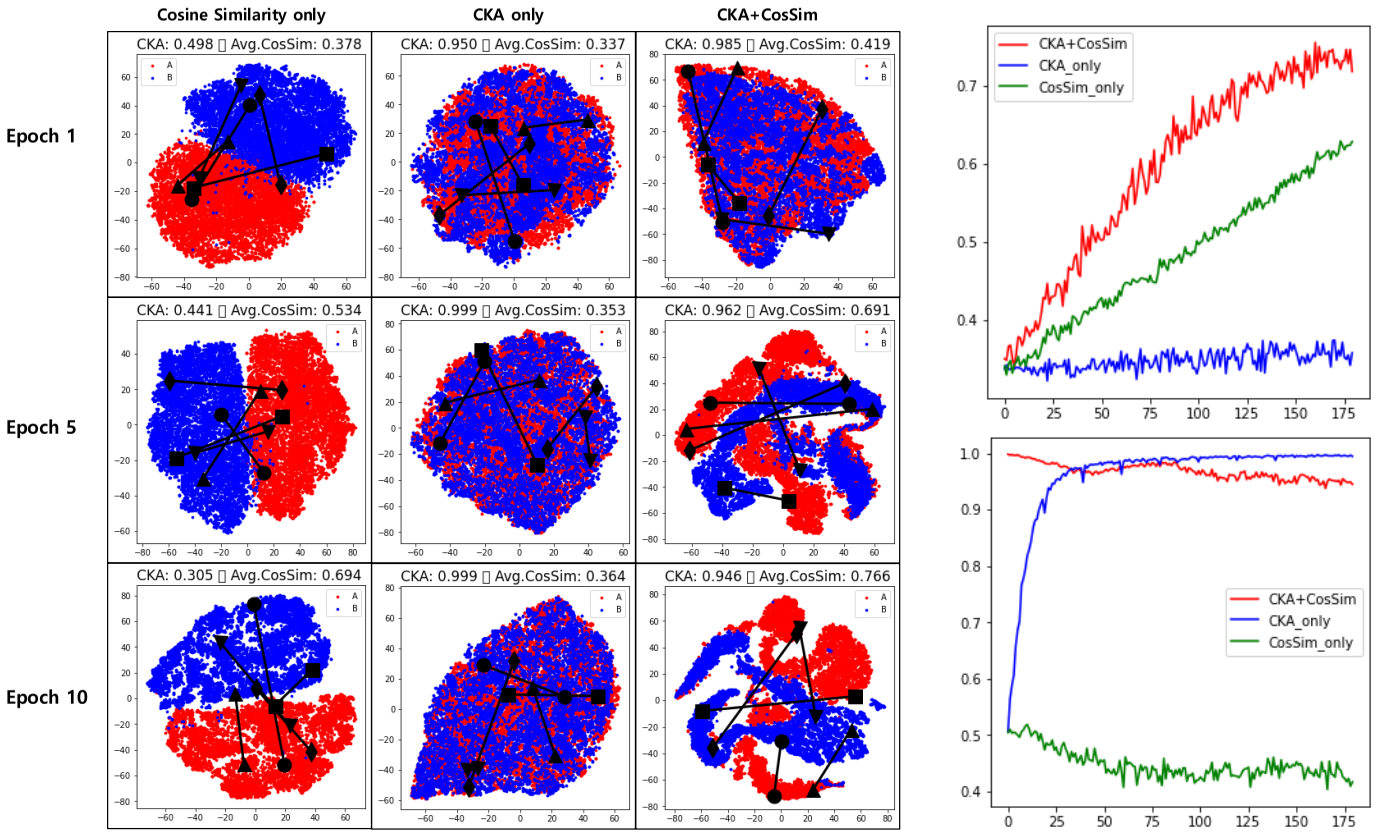


Figure 5.4: Cosine similarity learning results on the synthetic dataset. *(Left)* t-SNE visualizations of both encoded representations after 1, 5, and 10 epochs. *(Right top)* Training curve of the averaged cosine similarity over training steps. *(Right bottom)* Training curve of CKA between two representations over training steps.

Experiment settings

I make a synthetic dataset which simulates two different modalities with completely different characteristics as following three steps.

- I sample 10,000 class 'A' examples from a multivariate normal distribution with dimension size of 64.
- I also sample 10,000 class 'B' examples from a intricately designed mixture of multivariate normal distribution with the same dimension size.
- To simulate ground truth hard attention, I randomly make one-to-one correspondences between each example of 'A' and 'B'.
- The goal is to train two encoders for both 'A' and 'B' in the way that maximizes cosine similarity between two corresponding embedded vectors.

The main criterion for evaluation is the averaged cosine similarity between all corresponding examples of class 'A' and class 'B'. Figure 5.3 describes the t-SNE visualization of my synthetic dataset.

Then, I build two neural networks models to substitute for embedding modules. Each neural networks takes samples of each class as input respectively and encodes them into output vectors. I regard the output of each networks as two different representations of different modalities. Both encoders have the same architecture but do not share the weights. Each encoder has three fully-connected layers with the hidden size of 32, each layer followed by the ReLU activation and Batch Normalization. The mini-batch size is set to 512 and λ_{CKA} value is 0.1. I train the model with ADAM optimizer with initial learning late of 0.001, $\beta_1 = 0.9$, and $\beta_2 = 0.999$.

I test three methods for comparison; (a) directly maximize only the averaged cosine similarity, (b) directly maximize CKA only, and (c) Modality Alignment method that optimize both the criterion and CKA loss $\mathcal{L}_{cka}$. In the experiment with the method, I observe that pre-training CKA alone for few steps before optimizing the final loss

$\mathcal{L}_{final}$ as a warm-up increases the performance. All experiments with the method in this paper are also performed this warm-up.

Experiments Results

I summarize the results in Figure 5.4. The t-SNE visualizations of encoded representations over epochs reveals an interesting effect of the method (Left of Figure 5.4). Comparing the first column (trained with only cosine similarity loss) and the second column (trained with only CKA loss), only maximizing cosine similarity like existing multi-modal models does not make two representations similar as shown as CKA value drops from 0.498 to 0.305. In contrast, training with only CKA loss makes two encoders learn the inter-example structures extremely well. Also, it even increases the average cosine similarity slightly implying that there is indeed a correlation between inter-example structure similarity and cosine similarity. Finally, Modality Alignment method which optimizes both CKA loss and cosine similarity outperforms the conventional methods (*CosSim_only*), showing that the method can boost the training of cross attention module (Right top of Figure 5.4).

5.3.2 Modality Align on Video Question Answering Task

Lastly, I verify Modality Alignment method in Video Question Answering tasks as real-world scenarios. Video QA is one of the most challenging among multi-modal tasks because there exhibits a great deal of differences between the video and text modalities, causing severe text bias problem. With two standard benchmarks, following experiments demonstrate that Modality Alignment method also improves conventional models even in video QA tasks as the method closes the gap between two modalities.

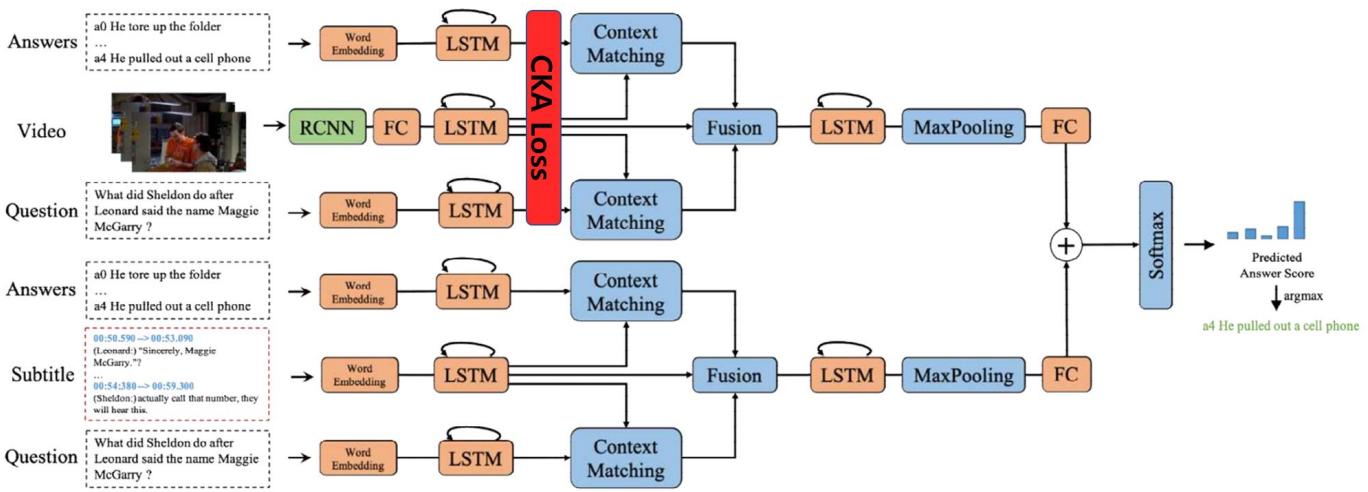


Figure 5.5: TVQA$_{abc}$ model. I utilize auxiliary CKA Loss between both modalities before the context matching module. The base structure is from [1].

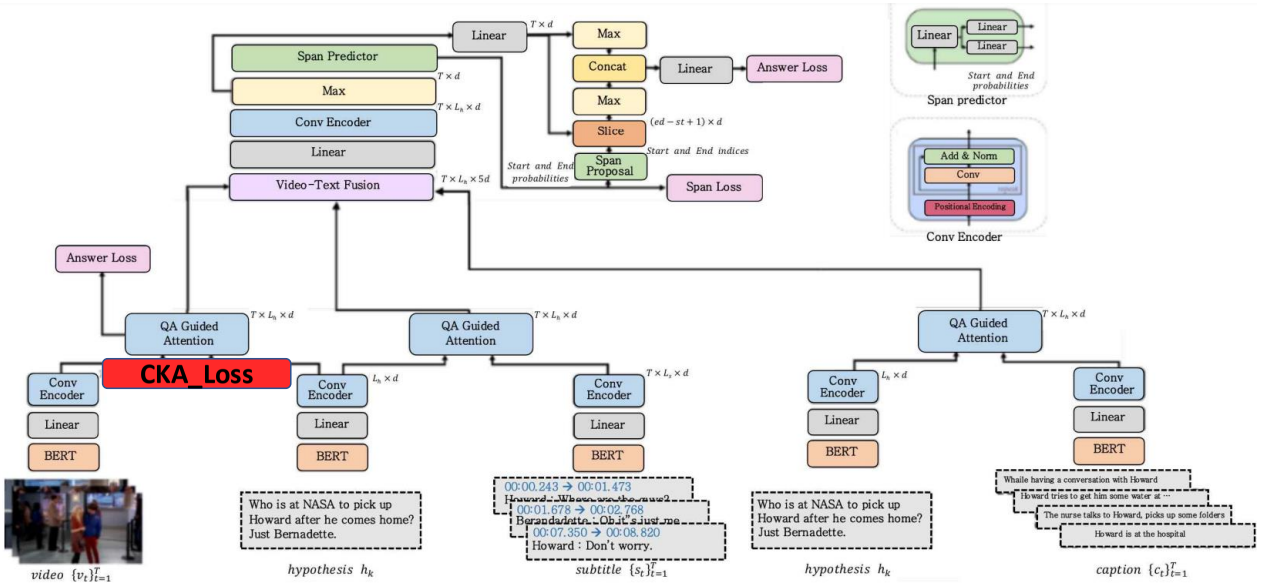


Figure 5.6: STAGE model with dense video captions. I utilize auxiliary CKA Loss between both modalities before the QA Guided Attention. I used a dense video captioning model MMT to solve text bias of the baseline model, to create captions from video information and use them as additional information. The base structure is from [2]

Datasets and Training details

I evaluate my approach on two benchmarks: TVQA [1] and TVQA+ [2]. TVQA is a large-scale video question answering dataset based on six popular TV shows: *The Big Bang Theory, How I Met Your Mother, Friends, Grey's Anatomy, House, Castle*. As a baseline model, I utilize TVQA$_{abc}$ which is proposed together with the TVQA benchmark. I apply CKA loss between the video embedding representation and the QA embedding representation of TVQA$_{abc}$ to apply Modality Alignment. The structure of TVQA$_{abc}$ I borrowed

For TVQA$_{abc}$, the pretrained Faster R-CNN and LSTM are used as visual embedding modules, and word embedding with LSTM are used as text embedding modules. And then CKA loss is configured before context matching information where cross modal attention takes place as in figure 5.5. The details of implementations are the same as the baseline model, λ_{CKA} value is 0.2, and the batch size is 32.

TVQA+ is a subset of TVQA that only uses The Big Bang Theory clips yet contains additional bounding box annotation for visual region feature. The training, validation, and test-public set consist of 23,545, 3,017, and 2,821 questions, respectively. I utilize STAGE as a baseline, a model proposed in TVQA+ benchmark paper. In STAGE model, the input images are encoded with pretrained Faster R-CNN as a visual embedding module and the input texts are encoded with pretrained BERT encoder as a text embedding module. Consequently, CKA loss is computed before the two presentations cross-modal attention is performed. I compute and maximize CKA between video representation and text representation before the cross-modal attention layer. The details of implementations are the same as the baseline model, λ_{CKA} value is 0.1, and the batch size is 4. I apply the grid search method to find the best value of λ_{CKA}.

Additionally, I used a dense video captioning model MMT [99] to reduce the text bias of the baseline model. I create dense captions from video information and use them as additional information. I briefly describe in the right part of the figure 5.6 that captured information is added as input stream.

In both cases, linear kernel is used for computing CKA, based on the findings of the previous study by [21] that there is no significant difference from other kernels such as RBF kernel.

Experiments Results

I report the experimental results evaluated with QA accuracy in Table 5.2 and corresponding CKA values in Table 5.1.

In experiments on TVQA dataset, QA accuracy of the baseline (TVQA$_{abc}$) is 67.70%, while Modality Alignment method increases the accuracy up to 69.38%. CKA value between video embedding representation and QA representation is also increased significantly from 0.3907 to 0.7815. I suppose that the trained alignment between two different modalities leads to the final performance improvement.

Similarly in experiments on TVQA+ dataset, comparing STAGE and STAGE+CKA in Table 5.2 shows a significant accuracy improvement from 70.31 to 72.89 with Modality Alignment method. CKA value also shows a large increase from 0.2694 to 0.6708 in Table 5.1, indicating the inter-example similarity structures of image representation and text representation are well trained to be similar. Through these results, I conclude that training the representational alignment between multiple modalities improves a Video QA model by enhancing the cross attention module.

In addition to Modality Alignment method, I exploit the generated caption in order to reduce the text bias by a video captioning model [99]. In Table 5.2, STAGE (video only) indicates the result of the model using only video features without subtitle information, and STAGE (sub) is the result of vice versa. The significant accuracy drop in STAGE (video only) implies that STAGE model is biased toward text modality as known as the text bias problem [74]. I generate additional captions with Multi Modal Transformer (MMT) model [99]. The generated captions are passed to the model as additional text inputs. With the aligning method plus the generated captions, I achieve the best result as shown at the bottom of Table 5.2 showing an additional 0.99 accuracy

Table 5.1: CKA between various modalities. In the case of uni-modality, the CKA value is initailly high, which means that the similarity between the representations is high, but the case of multi-modality is not. However, after CKA learning through my method, multi-modality also shows a high CKA value, increasing the similarity between the representations.

Model	CKA(Vid$_{emb}$,QA$_{emb}$)	CKA(Sub$_{emb}$,QA$_{emb}$)	CKA(Cpt$_{emb}$,QA$_{emb}$)
	Multi-modality	Uni-modality(Text)	Uni-modality(Text)
TVQA$_{abc}$	0.3907	0.8798	-
TVQA$_{abc}$ + CKA	**0.7815**	0.8528	-
STAGE	0.2694	0.8999	-
STAGE + Caption	0.3998	0.8625	0.8741
STAGE + Caption + CKA	**0.6708**	0.8878	0.9215

Table 5.2: VideoQA results evaluated with QA accuracy.

Model	QA Accuracy (%)
TVQA$_{abc}$	67.70
TVQA$_{abc}$ + CKA	**69.38**
STAGE (video only)	52.75
STAGE (sub only)	67.99
STAGE	70.31
STAGE + CKA	72.89
STAGE + CKA + Caption	**73.88**

improvement finally resulting 73.88 accuracy.

I also examine the impact of Modality Alignment method on embedding representation similarity of multiple modalities. As shown in Table 5.1, both CKA(Cpt$_{emb}$,QA$_{emb}$) and CKA(Sub$_{emb}$,QA$_{emb}$) are high because the subtitles, the generated captions and the QA pairs have the same text modality. However, CKA(Vid$_{emb}$,QA$_{emb}$) values are low without the method, indicating the different characteristics between two modalities. Applying Modality Alignment, CKA(Vid$_{emb}$,QA$_{emb}$) values become high as CKA of a single modality. Thus, the aligning method closes the gap between the video modality and the text modality.

In a nutshell, all three experiments verify that learning the representational alignment with CKA fits two different representations to have similar structures, enhances the cross attention module, and eventually leads to the performance improvement in Video-and-Language learning. In addition, my method can be easily applied to not only Video QA models but also any models for multi-modal tasks.

5.4 Conclusion

In this chapter, I propose Modality Alignment method which is based on CKA optimization in the previous chapter of this dissertation. In multi-modal tasks such as Video QA, there is a difference in characteristics between the two modalities, which reduces the effectiveness of cross-modal attention. To address this, I propose Modality Alignment method that optimizes the similarity between two embedding representation structures of two different modalities. Specifically, I maximize the similarity between representations by directly exploiting CKA as a training objective. In experiments, I verify that Modality Alignment method boosts cosine similarity learning in a synthetic environment, which is the basis of the attention method, and further improves the performance of multi-modal models for real word tasks. In the future, I will test the proposed method on various multi-modal learning tasks including Video-and-Language learning in order to confirm that it improves state-of-the-art modality alignment strategies.

Chapter 6

Conclusion

The Transformer architecture consists of attention layers that show strengths in extracting correlations between tokens and integrate the extracted information to produce appropriate outputs. Many studies have reported that utilizing Transformer architectures can yield new state-of-the-arts performance in various natural language processing problems. These advances have recently presented a new challenge to exploit additional contextual information outside of input data given to deep learning societies. In this dissertation, I propose a novel method to effectively utilize additional contextual information in addition to the input given in various tasks. Since humans recognize signals with much richer contextual information in their daily lives, using contextual information is a step toward human intelligence. I first propose an encoder for utilizing contextual representations that include contextual information from previous sentences for machine translation. I then propose a novel optimization objective function by in-depth analysis of representation similarities between contextual representations and input representations. Finally, I extend the context representation utilization methodology presented above to the multi-modal problems, presenting a methodology that significantly improves performance by simultaneously utilizing the context of visual information and the context of text information.

Bibliography

[1] J. Lei, L. Yu, M. Bansal, and T. Berg, "Tvqa: Localized, compositional video question answering," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1369–1379, 2018.

[2] J. Lei, L. Yu, T. Berg, and M. Bansal, "Tvqa+: Spatio-temporal grounding for video question answering," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8211–8225, 2020.

[3] 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*, pp. 5998–6008, 2017.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

[5] 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," *Journal of Machine Learning Research*, vol. 21, pp. 1–67, 2020.

[6] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in neural information processing systems*, pp. 5754–5764, 2019.

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[8] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does bert look at? an analysis of bert's attention," in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 276–286, 2019.

[9] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov, "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808, 2019.

[10] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein, "Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability," in *Advances in Neural Information Processing Systems*, pp. 6076–6085, 2017.

[11] A. Raganato and J. Tiedemann, "An analysis of encoder representations in transformer-based machine translation," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 287–297, 2018.

[12] Y. Li, J. Yosinski, J. Clune, H. Lipson, and J. E. Hopcroft, "Convergent learning: Do different neural networks learn the same representations?," in *FE@ NIPS*, pp. 196–212, 2015.

[13] P. Michel, O. Levy, and G. Neubig, "Are sixteen heads really better than one?," in *Advances in Neural Information Processing Systems*, pp. 14014–14024, 2019.

[14] D. Xie, J. Xiong, and S. Pu, "All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6176–6185, 2017.

[15] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," *arXiv preprint arXiv:1911.02116*, 2019.

[16] S. Maruf, A. F. Martins, and G. Haffari, "Selective attention for context-aware neural machine translation," in *Proceedings of NAACL-HLT*, pp. 3092–3102, 2019.

[17] E. Voita, P. Serdyukov, R. Sennrich, and I. Titov, "Context-aware neural machine translation learns anaphora resolution," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1264–1274, 2018.

[18] Y. Levine, I. Dalmedigos, O. Ram, Y. Zeldes, D. Jannai, D. Muhlgay, Y. Osin, O. Lieber, B. Lenz, S. Shalev-Shwartz, *et al.*, "Standing on the shoulders of giant frozen language models," *arXiv preprint arXiv:2204.10019*, 2022.

[19] B. Yang, J. Li, D. F. Wong, L. S. Chao, X. Wang, and Z. Tu, "Context-aware self-attention networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 387–394, 2019.

[20] S. Sukhbaatar, J. Weston, R. Fergus, *et al.*, "End-to-end memory networks," *Advances in neural information processing systems*, vol. 28, 2015.

[21] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *International Conference on Machine Learning*, pp. 3519–3529, 2019.

[22] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, 2016.

[23] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," tech. rep., 2016.

[24] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, pp. 3104–3112, 2014.

[25] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[26] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (Lisbon, Portugal), pp. 1412–1421, Association for Computational Linguistics, Sept. 2015.

[27] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

[28] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022.

[29] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, *et al.*, "Flamingo: a visual language model for few-shot learning," *arXiv preprint arXiv:2204.14198*, 2022.

[30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, pp. 8748–8763, PMLR, 2021.

[31] J. Zhang, H. Luan, M. Sun, F. Zhai, J. Xu, M. Zhang, and Y. Liu, "Improving the transformer translation model with document-level context," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 533–542, 2018.

[32] G. Tang, M. Müller, A. Rios, and R. Sennrich, "Why Self-Attention? A Targeted Evaluation of Neural Machine Translation Architectures," in *EMNLP*, 2018.

[33] K. Tran, A. Bisazza, and C. Monz, "The importance of being recurrent for modeling hierarchical structure," *arXiv preprint arXiv:1803.03585*, 2018.

[34] K. Sim Smith, "On Integrating Discourse in Machine Translation," in *Proceedings of the Third Workshop on Discourse in Machine Translation*, no. Section 2, pp. 110–121, 2017.

[35] L. Miculicich, D. Ram, N. Pappas, and J. Henderson, "Document-Level Neural Machine Translation with Hierarchical Attention Networks," in *EMNLP*, no. i, pp. 2947–2954, 2018.

[36] H. Xiong, Z. He, H. Wu, and H. Wang, "Modeling Coherence for Discourse Neural Machine Translation," in *AAAI*, no. 10, 2019.

[37] E. Voita, R. Sennrich, and I. Titov, "When a Good Translation is Wrong in Context: Context-Aware Machine Translation Improves on Deixis, Ellipsis, and Lexical Cohesion," in *ACL*, 2019.

[38] J. Tiedemann and Y. Scherrer, "Neural Machine Translation with Extended Context," in *Proceedings of the Third Workshop on Discourse in Machine Translation*, pp. 82–92, 2017.

[39] S. Jean, S. Lauly, O. Firat, and K. Cho, "Does neural machine translation benefit from larger context?," *arXiv preprint arXiv:1704.05135*, 2017.

[40] L. Wang, Z. Tu, A. Way, and Q. Liu, "Exploiting Cross-Sentence Context for Neural Machine Translation," in *EMNLP*, 2017.

[41] S. Maruf and G. Haffari, "Document Context Neural Machine Translation with Memory Networks," in *ACL*, pp. 1275–1284, 2018.

[42] S. Maruf, A. F. T. Martins, and G. Haffari, "Selective Attention for Context-aware Neural Machine Translation," in *NAACL*, mar 2019.

[43] R. Bawden, R. Sennrich, A. Birch, and B. Haddow, "Evaluating Discourse Phenomena in Neural Machine Translation," in *NAACL*, pp. 1304–1313, 2018.

[44] M. Müller, A. Rios, E. Voita, and R. Sennrich, "A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation," in *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 61–72, 2018.

[45] Z. Tu, Y. Liu, Z. Lu, X. Liu, and H. Li, "Context gates for neural machine translation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 87–99, 2017.

[46] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130*, 2017.

[47] M. Cettolo, M. Federico, L. Bentivogli, J. Niehues, S. Stüker, K. Sudoh, K. Yoshino, and C. Federmann, "Overview of the IWSLT 2017 Evaluation Campaign," in *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT 2017)*, pp. 1–14, 2017.

[48] P. Lison, J. Tiedemann, and M. Kouylekov, "Opensubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018.

[49] M. Schuster and K. Nakajima, "Japanese and Korean voice search," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 1, pp. 5149–5152, 2012.

[50] A. Vaswani, S. Bengio, E. Brevdo, F. Chollet, A. N. Gomez, S. Gouws, L. Jones, L. Kaiser, N. Kalchbrenner, N. Parmar, R. Sepassi, N. Shazeer, and J. Uszkoreit, "Tensor2tensor for neural machine translation," *arXiv preprint*, vol. arXiv:1803.07416, 2018.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[52] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pp. 177–180, 2007.

[53] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data?," *Perspectives on psychological science*, vol. 6, no. 1, pp. 3–5, 2011.

[54] X. Zhang, F. Wei, and M. Zhou, "Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5059–5069, 2019.

[55] J. Li, Z. Tu, B. Yang, M. R. Lyu, and T. Zhang, "Multi-head attention with disagreement regularization," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2897–2903, 2018.

[56] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[57] A. Bau, Y. Belinkov, H. Sajjad, N. Durrani, F. Dalvi, and J. Glass, "Identifying and controlling important neurons in neural machine translation," *arXiv preprint arXiv:1811.01157*, 2018.

[58] W. Zhou, T. Ge, K. Xu, F. Wei, and M. Zhou, "Scheduled drophead: A regularization method for transformer models," *arXiv preprint arXiv:2004.13342*, 2020.

[59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[60] M. Cettolo, M. Federico, L. Bentivogli, N. Jan, S. Sebastian, S. Katsuitho, Y. Koichiro, and F. Christian, "Overview of the iwslt 2017 evaluation campaign," in *International Workshop on Spoken Language Translation*, pp. 2–14, 2017.

[61] M. Ziemski, M. Junczys-Dowmunt, and B. Pouliquen, "The united nations parallel corpus v1. 0," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 3530–3534, 2016.

[62] G. Tang, R. Sennrich, and J. Nivre, "An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation," *WMT 2018*, p. 26, 2018.

[63] A. Morcos, M. Raghu, and S. Bengio, "Insights on representational similarity in neural networks with canonical correlation," in *Advances in Neural Information Processing Systems*, pp. 5727–5736, 2018.

[64] N. Maheswaranathan, A. Williams, M. Golub, S. Ganguli, and D. Sussillo, "Universality and individuality in neural dynamics across large populations of recurrent networks," in *Advances in neural information processing systems*, pp. 15603–15615, 2019.

[65] S. R. Kudugunta, A. Bapna, I. Caswell, N. Arivazhagan, and O. Firat, "Investigating multilingual nmt representations at scale," *arXiv preprint arXiv:1909.02197*, 2019.

[66] P. Rodríguez, J. Gonzalez, G. Cucurull, J. M. Gonfaus, and X. Roca, "Regularizing cnns with locally constrained decorrelations," *arXiv preprint arXiv:1611.01967*, 2016.

[67] N. Bansal, X. Chen, and Z. Wang, "Can we gain more from orthogonality regularizations in training deep cnns?," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 4266–4276, Curran Associates Inc., 2018.

[68] S. Gu, Y. Hou, L. Zhang, and Y. Zhang, "Regularizing deep neural networks with an ensemble-based decorrelation method.," in *IJCAI*, pp. 2177–2183, 2018.

[69] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with hilbert-schmidt norms," in *International conference on algorithmic learning theory*, pp. 63–77, Springer, 2005.

[70] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 795–828, 2012.

[71] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. S. Kandola, "On kernel-target alignment," in *Advances in neural information processing systems*, pp. 367–373, 2002.

[72] B. Ondrej, R. Chatterjee, F. Christian, G. Yvette, H. Barry, H. Matthias, K. Philipp, L. Qun, L. Varvara, M. Christof, *et al.*, "Findings of the 2017 conference on machine translation (wmt17)," in *Second Conference onMachine Translation*, pp. 169–214, The Association for Computational Linguistics, 2017.

[73] M. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a large annotated corpus of english: The penn treebank," 1993.

[74] R. Cadene, C. Dancette, M. Cord, D. Parikh, *et al.*, "Rubi: Reducing unimodal biases for visual question answering," *Advances in neural information processing systems*, vol. 32, pp. 841–852, 2019.

[75] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, *et al.*, "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *European Conference on Computer Vision*, pp. 121–137, Springer, 2020.

[76] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4779–4783, IEEE, 2018.

[77] R. Zellers, X. Lu, J. Hessel, Y. Yu, J. S. Park, J. Cao, A. Farhadi, and Y. Choi, "Merlot: Multimodal neural script knowledge models," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[78] L. Li, Y.-C. Chen, Y. Cheng, Z. Gan, L. Yu, and J. Liu, "Hero: Hierarchical encoder for video+ language omni-representation pre-training," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2046–2065, 2020.

[79] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2630–2640, 2019.

[80] Z. Yu, D. Xu, J. Yu, T. Yu, Z. Zhao, Y. Zhuang, and D. Tao, "Activitynet-qa: A dataset for understanding complex web videos via question answering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9127–9134, 2019.

[81] A. Yang, A. Miech, J. Sivic, I. Laptev, and C. Schmid, "Just ask: Learning to answer questions from millions of narrated videos," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1686–1697, 2021.

[82] L. Ye, M. Rochan, Z. Liu, and Y. Wang, "Cross-modal self-attention network for referring image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10502–10511, 2019.

[83] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *Advances in neural information processing systems*, vol. 32, 2019.

[84] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "Uniter: Universal image-text representation learning," in *European conference on computer vision*, pp. 104–120, Springer, 2020.

[85] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[86] P. Zhang, X. Li, X. Hu, J. Yang, L. Zhang, L. Wang, Y. Choi, and J. Gao, "Vinvl: Revisiting visual representations in vision-language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5579–5588, 2021.

[87] L. Chen, X. Yan, J. Xiao, H. Zhang, S. Pu, and Y. Zhuang, "Counterfactual samples synthesizing for robust visual question answering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10800–10809, 2020.

[88] K.-M. Kim, M.-O. Heo, S.-H. Choi, and B.-T. Zhang, "Deepstory: video story qa by deep embedded memory networks," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 2016–2022, 2017.

[89] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 6450–6459, 2018.

[90] 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*, pp. 2758–2766, 2017.

[91] Y. Ye, Z. Zhao, Y. Li, L. Chen, J. Xiao, and Y. Zhuang, "Video question answering via attribute-augmented attention network learning," in *Proceedings of*

the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 829–832, 2017.

[92] J. Liang, L. Jiang, L. Cao, Y. Kalantidis, L.-J. Li, and A. G. Hauptmann, "Focal visual-text attention for memex question answering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1893–1908, 2019.

[93] Z. Zhao, X. Jiang, D. Cai, J. Xiao, X. He, and S. Pu, "Multi-turn video question answering via multi-stream hierarchical attention context network.," in *IJCAI*, vol. 2018, p. 27th, 2018.

[94] A. Wang, A. T. Luu, C.-S. Foo, H. Zhu, Y. Tay, and V. Chandrasekhar, "Holistic multi-modal memory network for movie question answering," *IEEE Transactions on Image Processing*, vol. 29, pp. 489–499, 2019.

[95] J. Kim, M. Ma, K. Kim, S. Kim, and C. D. Yoo, "Progressive attention memory network for movie story question answering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8337–8346, 2019.

[96] H. Yun, T. Kang, and K. Jung, "Analyzing and controlling inter-head diversity in multi-head attention," *Applied Sciences*, vol. 11, no. 4, p. 1548, 2021.

[97] X. Pan, M. Wang, L. Wu, and L. Li, "Contrastive learning for many-to-many multilingual neural machine translation," *arXiv preprint arXiv:2105.09501*, 2021.

[98] 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*, pp. 815–823, 2015.

[99] J. Lei, L. Yu, T. L. Berg, and M. Bansal, "Tvr: A large-scale dataset for video-subtitle moment retrieval," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pp. 447–463, Springer, 2020.

초 록

최근 자연어 처리(NLP)를 위한 표준 아키텍처가 순환 신경망에서 트랜스포머 아키텍처로 발전했다. 트랜스포머 아키텍처는 토큰 간의 상관 관계를 추출하는 데 강점을 보여주고 추출한 정보를 통합하여 적절한 출력을 생성하는 attention layer들로 구성된다. 이러한 발전은 최근 딥 러닝 사회에 주어진 입력 데이터 밖의 추가 컨텍스트 정보를 활용하는 새로운 도전을 제시했다. 본 학위 논문에서는 다양한 자연어 처리 작업에서 주어진 입력 외에 추가적인 컨텍스트 정보를 효과적으로 활용하는 새로운 방법과 분석을 attention layer에 초점을 맞추어 제안한다. 먼저, 이전 문장에 대한 컨텍스트 정보를 효율적으로 내장하고, 메모리 어텐션 메커니즘을 통해 내장된 문맥 표현을 입력 표현에 융합하는 계층적 메모리 컨텍스트 인코더(HMCE)를 제안한다. 제안된 HMCE는 다양한 문맥 인지 기계 번역 작업에서 추가 문맥 정보를 활용하지 않는 트랜스포머와 비교하였을 때 더 뛰어난 성능을 보인다. 그런 다음 문맥 표현과 입력 표현 사이의 어텐션 메커니즘을 개선하기 위해 문맥 표현과 입력 표현 사이의 표현 유사성을 Centered Kernel Alignment(CKA)를 이용하여 심층 분석하며, CKA를 최적화하는 방법을 제안한다. 마지막으로, 문맥 정보가 시각 양식으로 주어지는 다중 모달 시나리오에 대해 CKA 최적화 방법을 모달리티 정렬 방법으로 확장한다. 이 Modality Alignment 방법은 멀티 모달간 표현 유사성을 극대화하여 비디오 질문 응답 작업에서 큰 성능 향상을 가져온다.

주요어: 딥 러닝, 자연어 처리, 문맥 표현, 멀티 모달 학습, 크로스 모달 어텐션
학번: 2015-20956