Attention Mechanism Optimization for Sub-Symbolic-Based and Neural-Symbolic-Based Natural Language Processing

Jinjie Ni

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2023

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

30/Dec/2022

Date

Jinjie Ni

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accordance with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

30/Dec/2022

Date

Erik Cambria

Authorship Attribution Statement

This thesis contains material from 4 paper(s) accepted by the following peer-reviewed journal(s) / conferences in which I am listed as an author.

Section 2.7 is published with material from: Jinjie Ni, Tom Young, Vlad Pandelea, Fuzhao Xue, Erik Cambria. Recent Advances in Deep Learning Based Dialogue Systems: A Systematic Survey. Artificial Intelligence Review, DOI: 10.1007 /s10462-022-10248-8.

The contributions of the co-authors are as follows:

- I came up with the key ideas, researched the related works, and drafted the whole manuscript.
- Tom Young, Vlad Pandelea, and Fuzhao Xue made up the missing works and revised the article.
- A/P Erik Cambria provided guidance on the main ideas and revised the article.

Chapter 4 is published with material from: Jinjie Ni, Rui Mao, Zonglin Yang, Han Lei, Erik Cambria. Finding the Pillars of Strength for Multi-Head Attention. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2023).

The contributions of the co-authors are as follows:

- I came up with the key ideas, designed and implemented all source code and experiments, and drafted the whole manuscript.
- Dr. Rui Mao, Zonglin Yang, and Han Lei provided valuable advice on the experiment design and paper organization and revised the manuscript.
- A/P Erik Cambria provided guidance on the main ideas and revised the article.

Chapter 5 is published with material from: Jinjie Ni, Yukun Ma, Wen Wang, Qian Chen, Dianwen Ng, Han Lei, Trung Hieu Nguyen, Chong Zhang, Bin Ma, Erik Cambria. Adaptive Knowledge Distillation between Text and Speech Pre-trained Models. In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023).

The contributions of the co-authors are as follows:

- I came up with the key ideas, designed and implemented all source code and experiments, and drafted the whole manuscript.

- Dr. Yukun Ma, Dr. Wen Wang, and Dr. Qian Chen provided valuable suggestions on the proposed ideas, helped prepare the data, and helped revise the article.
- Dianwen Ng and Han Lei made constructive advice on the details of the idea and helped with the article organization.
- Dr. Trung Hieu Nguyen and Dr. Chong Zhang helped build the GPU deployment platform.
- Dr. Bin Ma and A/P Erik Cambria provided guidance on the main ideas and revised the article.

Chapter 6 is published with material from: Jinjie Ni, Vlad Pandelea, Tom Young, Haicang Zhou, Erik Cambria. HiTKG: Towards Goal-Oriented Conversations via Multi-Hierarchy Learning. Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2022), 36(10), 11112-11120.

The contributions of the co-authors are as follows:

- I came up with the key ideas, designed and implemented all source code and experiments, and drafted the whole manuscript.
- Vlad Pandelea and Tom Young provided valuable advice on the experiment design and paper organization and helped revise the manuscript.
- Haicang Zhou provided help on the graph distance computation part and gave valuable suggestions on the other ideas.
- A/P Erik Cambria provided guidance on the main ideas and revised the article.

30/Dec/2022

Date

Jinjie Ni

Acknowledgements

I wish to express my greatest gratitude to my supervisor Prof. Erik Cambria for his mentorship and support. His passionate guidance and encouragement made it much easier for me to become a qualified researcher. His kindness and patience gave me the freedom to explore the truth, which I will keep in mind forever.

I also wish to express my gratitude to the members of the Sentic team and CIL who made my Ph.D. journey enjoyable and memorable. Special thanks to my friends & teachers: Tom Young, Vlad Pandelea, Zonglin Yang, Dr. Rui Mao, Fuzhao Xue, Jiaxing Xu, and Haicang Zhou, whose collaboration provided valuable guidance and encouragement to my research.

I need to express my gratitude to my colleagues during the internship at Alibaba DAMO, who helped me grow as a researcher with their valuable experience and strong expertise. Special thanks to Dr. Yukun Ma, Dr. Wen Wang, Dianwen Ng, Dr. Trung Hieu Nguyen, Dr. Chong Zhang, Dr. Qian Chen, and Dr. Bin. Ma. Dr. Yukun provided valuable guidance and mentorship to my research at DAMO.

I wish to express my deepest gratitude to my parents Xiaohui Ni and Jing Xu, who offered their greatest love and care from my birth to today. Xiaohui Ni is the best father who taught me to stay cautious and curious in my life, and his way of teaching makes all of my learning processes filled with fun. Jing Xu is the best mother who taught me to always pay attention to details and be patient, and it is her belief that encourages me to check myself again and again after thousands of failures. It's heartbreaking that she leaves us alone, but I always believe that she lives in another dimension we cannot see. Rest in peace in this world, my dear Mom, and wish you the best happiness in another.

Most importantly, I express my deepest gratitude and love to my wife Han Lei. It's her trust and understanding that drive me forward in this challenging journey. She taught me how to become a sincere person that reveals the real me to this world. I believe that she is the one that brings me the most relief and joy, the one that witnesses all my merriment and sorrow, and the one that supports all my dream and faith. Life with you is the greatest journey I will ever be on.

"From sacrifice comes meaning. From struggle comes purpose."

—Ryan Holiday

To my dear family

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviations	Full Name
GHA	Grouped Head Attention
GHA-PS	Grouped Head Attention - Pillars of Strength
GHT	Grouped Head Transformer
GHT-PS	Grouped Head Transformer - Pillars of Strength
PS	Pillars of Strength
GCT	Group-Constrained Training
V2S	Voting-to-Stay
PAD	Prior-informed Adaptive knowledge Distillation
ASP	Attention-based Significance Priors
AASA	Anchor-based Adaptive Span Aggregation
HITKG	Hierarchical Transformer based Knowledge Graph Walker
MDI	Multi-source Decoding Inputs
OLH	Output-level Length Head
HAGD	Hierarchical Attention based Graph Decoder
MP	MetaPath

Abstract

The capability for machines to transduce, understand, and reason with natural language lives at the heart of Artificial Intelligence not only because natural language is one of the main mediums for information delivery, residing in documents, daily chats, and databases of various languages, but also because it involves many key aspects of intelligence (e.g., logic, understanding, abstraction, etc.). Empowering the machine with more linguistic intelligence may benefit a wide range of real-world applications such as Machine Translation, Natural Language Understanding, Dialogue Systems, etc.

At present, there are two popular streams of approaches for building intelligent Natural Language Processing (NLP) systems, i.e., sub-symbolic and neural-symbolic approaches. Sub-symbolic approaches learn implicit representations on the corpus that is unstructured, which is massive in amount but results in poor interpretability and reasoning ability of the learned models; neural-symbolic approaches integrate neural and symbolic architectures to incorporate structured symbolic data (e.g., semantic nets, knowledge graphs, etc.) as an external knowledge source, which makes the learned model more interpretable and logical, but the structured symbolic data is hard to be fully represented and it is comparatively scarce. As a result, both streams of approaches deserve studying, since they have their respective strengths and weaknesses, working complementarily in different tasks/scenarios.

Meanwhile, attention-based models, such as Transformers, have achieved huge success in many NLP tasks such as Machine Translation, Language Modeling, Question Answering, etc. However, the attention itself has many issues, such as redundancy, quadratic complexity, weak inductive bias, etc. Besides, the previous applications of attention-based models in various NLP tasks are problematic, e.g., omitting the prior attention distribution, large computation complexity, weak long-term reasoning capability, etc.

To this end, this thesis explores novel attention architectures for NLP tasks that are currently based mainly on sub-symbolic or neural-symbolic approaches to solve the existing issues and advance the state-of-the-art. In particular, for sub-symbolic-based tasks, we study Machine Translation, Language Modeling, Abstractive Summarization, and Spoken Language Understanding; for neural-symbolic-based tasks,

we study Dialogue Commonsense Reasoning. The following lists the main contributions of this thesis:

- We study the redundancy and over-parameterization issues of Multi-Head Attention (MHA). We find that, in a certain range, higher compactness of attention heads (i.e., the intra-group heads become closer to each other and the inter-group ones become farther) improves the performance of MHA, which forces the MHA to focus on the most representative and distinctive features, providing guidance for future architectural designs. Accordingly, we propose a divide-and-conquer strategy that consists of Group-Constrained Training (GCT) and Voting to Stay (V2S). It mitigates the redundancy and over-parameterization issues of MHA. Our method uses fewer parameters and achieves better performance, outperforming the existing MHA redundancy/parameter reduction methods. We verify our methods on three well-established NLP tasks (i.e., Machine Translation, Language Modeling, and Abstractive Summarization). The superior results on datasets with multiple languages, domains, and data sizes demonstrate the effectiveness of our method.

- We ease the modality and granularity inconsistency problem when distilling knowledge from the teacher understanding model to the student ones, by refining the attention hidden states based on the attention map distribution. We propose to apply the Attention-based Significance Priors (ASP) to improve the semantic knowledge transfer from text to speech. We further propose the Anchor-based Adaptive Span Aggregation algorithm (AASA) that narrows the modal granularity gap of alignments. To the best of our knowledge, we are the first that evaluate multiple different alignment strategies beyond vanilla global and local alignments to study the feasibility of metric-based speech-text distillations. The results on three spoken language understanding benchmarks (i.e., Intent Detection, Slot Filling, and Emotion Recognition) verify our assumptions and claims.

- We improve the multi-source and long-term Dialogue Commonsense Reasoning (DCR) process, which is a new and difficult problem in NLP, by presenting a hierarchical attention-based decoding block. We propose the first Transformer-based KG walker that attentively reads multiscale inputs for graph decoding. Specifically, Multi-source Decoding Inputs (MDI) and Output-level Length Head (OLH) are presented to strengthen the controllability and multi-hop reasoning ability of the Hierarchical Attention-based Graph Decoder (HAGD). We further propose a two-hierarchy learning framework to train the proposed hierarchical attention-based KG walker, in order

to learn both turn-level and global-level KG entities as conversation topics. This is the first attempt to learn models to make natural transitions towards the global topic in KG, where we present a distance embedding to incorporate distance information. Moreover, we propose MetaPath (MP) to concurrently exploit entity and relation information when reasoning, which is proved essential as the backbone method for KG path representation, providing a paradigm for KG reasoning. The results on the DCR dataset OpendialKG show that HiTKG achieves a significant improvement in the performance of turn-level reasoning compared with state-of-the-art baselines. Additionally, both automatic and human evaluation prove the effectiveness of the two-hierarchy learning framework for both short-term and long-term DCR.

Chapter 1

Introduction

1.1 Background and Motivation

Intelligence has been defined in many aspects: the capacity for logic, understanding, abstraction, self-awareness, emotional knowledge, reasoning, learning, planning, critical thinking, creativity, and problem-solving. In a broader sense, it might be defined as the capacity for information perception, inference, and retaining information as knowledge for application to adaptive behaviors in a given environment or context [8]. The ability for machines to transduce, understand, and reason with language lives at the heart of Artificial Intelligence not only because language represents the most common form of information, residing in documents, daily chats, and databases of various forms, but also because it involves many key aspects of intelligence (e.g., logic, understanding, abstraction, etc.). Building intelligent systems that process natural language like human beings would benefit a broad range of real-world applications such as Machine Translation, Question Answering, Dialogue Systems, etc. Currently, sub-symbolic and neural-symbolic approaches are the two main streams for building intelligent Natural Language Processing (NLP) systems.

Sub-symbolic NLP approaches [9] mainly refer to data-driven neural approaches that learn implicit representations for NLP tasks in an end-to-end way. Most recent developments in neural approaches have given NLP applications an unprecedented performance increase, piquing the interest of the whole NLP community to study sub-symbolic approaches. For instance, Neural Machine Translation (NMT),

which utilizes deep neural networks as the core architecture, has gradually replaced phrase-based statistical techniques in Machine Translation (MT) since NMT performs better. Similarly to this, recurrent architectures and deep learning models have supplanted early approaches for Named Entity Recognition (NER) that were based on dictionaries, ontologies, and syntactic grammar rules. Large neural networks have been shown to be superior to conventional machine learning methods such as Support Vector Machines (SVM). First off, these models are convenient to implement since they can often be trained using a single end-to-end architecture and do not demand conventional task-specific feature engineering. Second, the large representation space of neural models can fit a broad range of data, being more powerful to represent the enormous feature space of real-world data [10].

Neural-symbolic NLP approaches [11] integrate neural and symbolic approaches. Symbolic NLP approaches assume that symbols are the fundamental building blocks of human intellect and that the cognitive process consists of a sequence of explicit judgments based solely on symbolic representations such as semantic nets, dependency trees, knowledge graphs, etc. Generally, the readability and interpretability of symbolic models are strong. The finite and discrete symbolic representations, however, are not tolerant of ambiguous and noisy data and are insufficient to describe all the fundamental relationships among the data. Whereas neural-symbolic approaches complement this drawback by integrating symbolic knowledge with neural architectures, being robust, reliable, and interpretable at the same time [12]. Thus, they can be applied to solve more high-level tasks such as understanding and reasoning.

Note that sub-symbolic and neural-symbolic approaches have their respective advantages and drawbacks, which leads to their co-existence in the current NLP research community. Sub-symbolic models can be trained in an end-to-end way with a large amount of annotated/unannotated unstructured data, which is easier to obtain and huge in number, making it easy to learn enormous knowledge. However, sub-symbolic models suffer from poor interpretability and reasoning ability [12], and thus are unreliable in some cases. For example, given different inputs, it's hard to control the commonsense facts in the output to be consistent. By contrast, neural-symbolic approaches are more interpretable and logical because of their external structured knowledge. However, high-level structured data is difficult to be fully represented [13] (e.g., their structural information is hard to represent) and

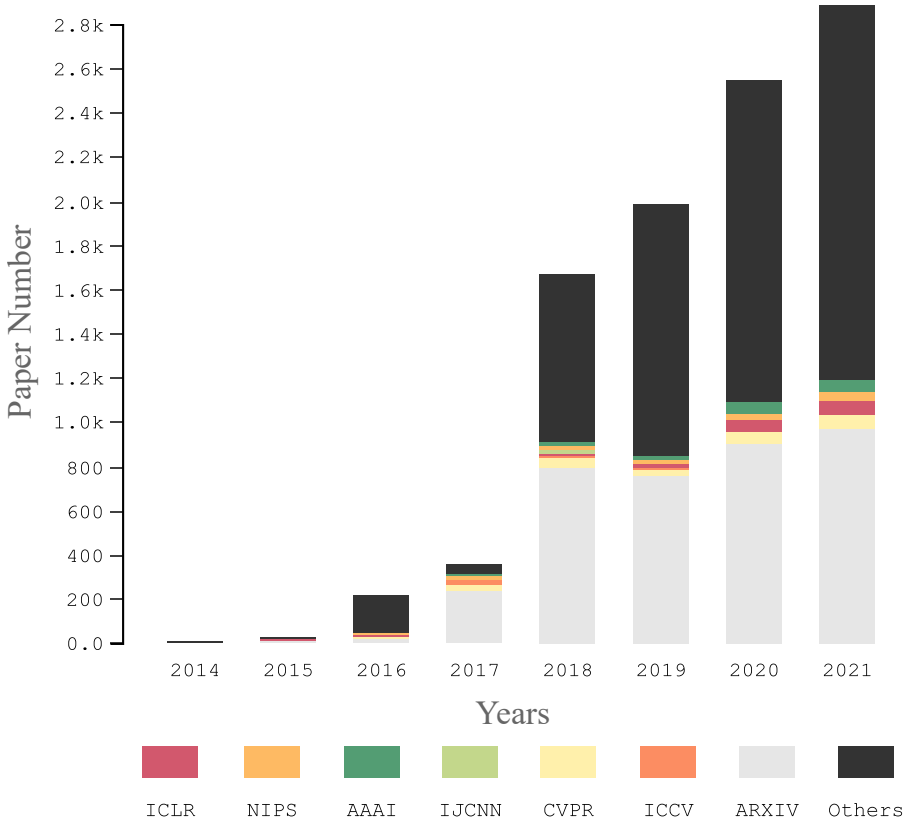


FIGURE 1.1: The number of papers studying attention-based architectures and their applications by years (2014-2021). The statistics are mainly from ICLR, NIPS, AAAI, IJCNN, CVPR, ICCV, and ArXiv. The category of *Others* refers to these sources: ACL, ACM, EMNLP, ICML, ICLR, ICASSP, ECCV, ACCV, CORR, ICRA, ICPR, IEEE ACCESS, and Neurocomputing.

it is scarce compared to unstructured ones [10]. These issues restrict their power to a large extent. As a result, in this thesis, we study both sub-symbolic-based and neural-symbolic-based NLP tasks to offer a more extensive exploration[1]. Here sub- or neural-symbolic-based tasks refer to the tasks whose recent approaches in its corresponding research community are mainly sub- or neural-symbolic-based.

As shown in Figure 1.1, the work studying attention-based architectures and their applications is rapidly increasing over the past 8 years and is expected to keep growing. The attention-based models, such as Transformer [5], have been a go-to backbone in both sub-symbolic-based [14] and neural-symbolic-based [13] NLP tasks. Attention performs a relevance-based pooling operation that joins all tokens of a sequence, presenting a graph-like inductive bias. Though effective, the

[1]Traditional Machine Learning approaches, symbolic approaches, and approaches that combine them are no longer popular in the NLP community for their limited performance.

attention mechanism itself has been known for many problems such as redundancy [15, 16], quadratic complexity [17, 18], weak inductive bias [14, 19], etc., impeding the model's scalability and performance in many circumstances. Besides, the previous applications of attention-based models in various NLP tasks are problematic, e.g., omitting the prior attention distribution [20], large computation complexity [21], weak long-term reasoning capability [22], etc. Recent years have witnessed a multitude of works that design new attention paradigms for NLP tasks, which eases the issues to some extent. However, this line of research is still in its infancy, which drives us to optimize attention-based architectures and their training methods to solve the challenging NLP problems.

1.2 Research Objective

The main objectives of this thesis are outlined as follows:

- Shed light on the general designs of high-performing attention mechanisms for sub-symbolic-based NLP tasks such as Machine Translation, Language Modeling, Abstractive Summarization, and Spoken Language Understanding, and advance the state-of-the-art.
- Explore the application of attention-based architectures in neural-symbolic-based NLP tasks such as Dialogue Commonsense Reasoning, providing new paradigms for future work.

1.3 Problem Overview and Research Scope

In this thesis, we optimize the attention mechanisms for both sub-symbolic-based and neural-symbolic-based tasks to solve their existing problems and fully study the effectiveness of attention mechanism variants. In particular, we study a set of core sub-symbolic-based NLP tasks: Machine Translation, Language Modeling, Abstractive Summarization, and Spoken Language Understanding; for neural-symbolic-based tasks, we study Dialogue Commonsense Reasoning, which is a representative task in this category.

The focused areas of this thesis are introduced in the following subsections.

1.3.1 Multi-Head Attention Optimization

The attention is the key module of Transformer [23]. Single-head attention projects the input sequence into three matrices which serve as the packed queries (Q), keys (K), and values (V). It computes the dot products between the queries and keys, divides each by $\sqrt{d_k}$, applies a softmax function to compute the weights, and performs a weighted sum on the values:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1.1}$$

It is proved beneficial to linearly project the queries, keys, and values h times with different, learnable linear projections to a smaller dimension $d_k = d_{model}/h$. The h projected versions of queries, keys, and values perform the attention function in parallel, and then perform a concatenation and projection on the resulting values:

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \tag{1.2}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{1.3}$$

where the W_i^Q, W_i^K, W_i^V, and W^O are learnable matrices. Compared to single-head attention, Multi-Head Attention (MHA) jointly attends to information from h representation subspaces at different positions, which allows it to learn a task from multiple perspectives [23].

Regarding the relationships between the source and target, there are two categories of attention that are commonly used:

- **Self-attention.** Self-attention relates different positions of a sequence to form new contextual representations. In MHA, we set $Q = K = V = X$ in Eq. 1.2, where X is the output representation of the previous layer. Both the source and target sequence are the input itself.
- **Cross-attention.** Cross-attention mixes two different sequences to form the new contextual representations. It takes the target sequence to project into queries (Q) and the source sequence to project into keys (K) and values (V).

Attention is a powerful representation reconstruction mechanism that has been proven effective in many works of different fields [5, 24, 25].

However, attention has been found to have many issues, such as complexity issues, lack of priors, parallel MHA design, etc., that harm its performance, which drives us to study the attention optimization problem. We discuss the issues of attention in detail in §2.1.

In addition, different NLP tasks require different inductive biases, and thus adapting the attention mechanism for different tasks is necessary to achieve better performance. For transduction tasks with short inputs, the main concern is to look at the whole sequence and have each head looking at different aspects [26]; whereas for transduction tasks with long inputs, the main issue is the quadratic complexity of the attention map computation [17, 18], hence local receptive fields are preferred to reduce computation complexity and out-of-memory issues. For language understanding tasks, attention mechanisms with adaptive granularity are preferred, so that the attention is able to catch both global and local semantics for better understanding, either when learning from datasets [27] or teacher models [20]. For language reasoning tasks, one of the main concerns is how to better utilize the current input information and history reasoning trajectory to achieve long-term future predictions, so attention-based architectures that deal with multiple inputs and have long-term memories are preferred [13]. Moreover, tasks from different fields, such as NLP and Computer Vision (CV), require different model inductive biases as well. For example, NLP tasks assume that temporal dependencies exist between tokens, whereas CV tasks require more spatial dependencies.

1.3.2 Machine Translation

Machine translation (MT) is a sub-field of computational linguistics that studies algorithms translating text or speech from one language to another. Given a source language sequence $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_{n1}\}$, the task is to learn a mapping to convert $\mathbf{X}$ into the target language sequence $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_{n2}\}$.

The studies on MT have a long history [28]. In recent years, this field is still active and numerous effective methods have been proposed. Recent advancements in Deep Learning have led to the emergence of a branch known as Neural Machine

Translation (NMT). NMT has a significant advantage over more established approaches like Phrase-Based Statistical Machine Translation (PBSMT) due to its straightforward architecture and capacity to capture complex sentence dependencies, becoming a new trend in the community. Numerous NMT models have been proposed, some of which made significant advancements with state-of-the-art outcomes. As the two most impactful attention-based approaches, Bahdanau et al. [4] and Vaswani et al. [5] built the foundation for the developments of NMT in recent years.

MT datasets that are most commonly adopted by the community are from WMT and IWSLT benchmarks, such as WMT 2014 en-de, WMT 2014 en-fr, IWSLT 2014 de-en, etc. BLEU [29] is the main evaluation metric.

1.3.3 Language Modeling

Language Modeling (LM), a well-known problem of probabilistic density estimation, is the most common self-supervised task in NLP. LM is a generic term, and in practice, it often refers to auto-regressive LM or unidirectional LM in particular. Given a text sequence $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, its joint probability $p(\mathbf{X})$ is termed as:

$$p(\mathbf{X}) = \prod_{t=2}^{n} p(\mathbf{x}_t | \mathbf{x}_1, ..., \mathbf{x}_{t-1}) \tag{1.4}$$

We can apply a neural encoder f_{enc} to model the text context $\{\mathbf{x}_1, ..., \mathbf{x}_{t-1}\}$, which is used to predict the conditional probability $p(\mathbf{x}_t | \mathbf{x}_1, ..., \mathbf{x}_{t-1})$:

$$p(\mathbf{x}_t | \mathbf{x}_1, ..., \mathbf{x}_{t-1}) = f_{LM}(f_{enc}(\{\mathbf{x}_1, ..., \mathbf{x}_{t-1}\})) \tag{1.5}$$

Where f_{LM} is a predictive neural model. With a large corpus, we may use Maximum Likelihood Estimation (MLE) [30] to train the entire network to maximize the likelihood from Eq.1.4. Each token in unidirectional LM only encodes itself and the leftward context tokens, which is a disadvantage. Better contextual representations of text should incorporate both forward and backward contextual information. Bidirectional LM (BiLM), which combines two unidirectional LMs: a forward left-to-right LM and a backward right-to-left LM, offers an enhanced solution. [31]

suggested a two-tower model for the BiLM, with the forward tower controlling the left-to-right LM and the backward tower controlling the right-to-left LM. In addition, attention-based Pre-Trained Models (PTMs) have recently empowered the LM task greatly. They do not need to be trained from scratch for each LM dataset because they have learned universal language representations from a vast corpus of data. A small amount of data from the downstream LM task is enough to achieve good results.

The WikiText-103 [32] and Billion Word [33] are popular datasets of LM. The perplexity is adopted as a common evaluation metric for LM.

1.3.4 Abstractive Summarization

Abstractive Summarization (AS) is known as creating a brief and succinct summary that encapsulates the key concepts of the source text. The generated summaries might include new words and phrases that are not in the original text. Given a source document sequence $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_{n1}\}$, the task is to learn a mapping to convert the source document sequence $\mathbf{X}$ into the target summary sequence $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_{n2}\}$.

Like MT and LM, recent advances in AS are also empowered by attention-based models. The difference is that its input is too long to compute its self-attention. Thus, the AS works either truncate the inputs, or develop efficient [34] / recurrent [35] attention mechanisms. As has longer inputs and requires more abstraction ability, it is considered more difficult than MT and LM [36].

The CNN-DailyMail [37] is a common AS dataset, and the F-1 ROUGE [38] is the standard evaluation metric for this task.

1.3.5 Spoken Language Understanding

Spoken Language Understanding (SLU) is a set of tasks in NLP intending to learn models to understand the content of speech input. In this thesis, we focus on three tasks of SLU: Intent Detection (IC), Emotion Recognition (ER), and Slot Filling (SF).

Intent Detection. In the Intent Detection (ID) task, the model classifies user utterances $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_{n1}\}$ into specified classes of user intentions $c_i \in C$, where $C = \{c_1, ..., c_{n2}\}$. Normally, the Fluent Speech Commands dataset [39] is adopted, where each utterance is annotated with three labels of intent: action, object, and location. Accuracy (ACC) is usually chosen as the evaluation metric.

Slot Filling. In the Slot Filling (SF) task, a series of semantic slot-types $\mathbf{Y}^1 = \{\mathbf{y}_1^1, ..., \mathbf{y}_{n2}^1\}$ and slot-values $\mathbf{Y}^2 = \{\mathbf{y}_1^2, ..., \mathbf{y}_{n2}^2\}$ are predicted from an utterance $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_{n1}\}$, such as a slot-type *FromLocation* of the slot-value *London*. An SLU system cannot operate without both slot-types and slot-values. Slot-type F1 score and slot-value CER are normally used as the evaluation metrics [40]. The popular dataset Audio SNIPS synthesizes multi-speaker utterances for SNIPS [41]. As a standard setting, US-accent speakers are chosen for training, while others are chosen for validation/testing.

Emotion Recognition. In the Emotion Recognition (ER) task, each utterance $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_{n1}\}$ is assigned an emotion class $c_i \in C$, where $C = \{c_1, ..., .c_{n2}\}$. A common ER dataset is IEMOCAP [42]. Following the traditional settings, the unbalanced emotion classes are dropped to leave the final four classes (neutral, happy, sad, and furious) with a comparable number of data points. Normally, accuracy (ACC) is the evaluation metric.

1.3.6 Dialogue Commonsense Reasoning

Commonsense knowledge is information about everyday life that is generally accepted by the majority of people and includes real-world experience. Building models for natural language understanding and, more broadly, AI systems that can reason about the world in the same way as humans do usually requires the assistance of external commonsense knowledge bases such as Knowledge Graph (KG), which contains commonsense relationships between real-world entities.

We study the Dialogue Commonsense Reasoning (DCR) task that predicts a topic sequence $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_n\}$ on KG, given the dialogue history $\mathbf{X}_d = \{\mathbf{x}_d^1, ..., \mathbf{x}_d^i\}$ and topic trajectory $\mathbf{X}_t = \{\mathbf{x}_t^1, ..., \mathbf{x}_t^j\}$ [13]. DCR aims to analyze and create conversation topics considering both the dialogue flow and commonsense knowledge,

because human conversations are usually guided by several topics that are in line with the current chat context and are mutually related according to commonsense knowledge. Such KG reasoning models are usually named dialogue graph walkers.

Current dialogue graph walkers can generally be divided into recurrent walkers [22, 43–45] and graph attention-based walkers [21]. Recurrent walkers decode KG paths depending on a fixed-length vector, which creates a bottleneck for performance. Graph attention-based KG walkers are good at achieving optimal performance since they reserve all potential paths, but such a mechanism is too high in computation complexity to be scalable to multi-hop reasoning. In addition, both recurrent and graph attention-based walkers neglect the hierarchical structure of the input source and make separate predictions for entity and relation paths, which harms their performance. Besides, these walkers only plan turn-level topics based on the dialogue history, which means that their reasoning is local and undirected, whereas conversations between humans are guided by an ultimate topic. Considering the above-mentioned problems, this task is still in its infancy and deserves more study.

For DCR, the OpenDialKG [22] is a popular dataset. F1 score and human evaluation are normally applied to evaluate the generated sequences.

1.4 Major Contributions

In general, this thesis optimizes the attention mechanisms for sub-symbolic-based and neural-symbolic-based NLP tasks. The attention mechanisms are transformed to solve the inherent issues of the attention mechanism itself such as redundancy and over-parameterization, or issues that exist in specific tasks such as the granularity gap problems in SLU and the short-sighted problems in DCR.

More specifically, they can be stated in the following three sub-sections.:

1.4.1 Grouped Head Attention for Language Transduction

We study the redundancy and over-parameterization issues of multi-head attention. Three core tasks of NLP are involved: Machine Translation, Language Modeling, and Abstractive Summarization. We find that the high compactness of attention

heads (i.e., the intra-group heads become closer to each other and the inter-group ones become farther) improves MHA's performance, which leads the MHA to focus on the most representative and distinctive features, providing guidance for future architectural designs. Accordingly, we propose a divide-and-conquer strategy that consists of GCT and V2S. It mitigates the redundancy and over-parameterization issues of MHA. Our method uses fewer parameters and achieves better performance, outperforming the existing MHA redundancy/parameter reduction methods. We verify our methods on three well-established NLP tasks. The superior results on datasets with multiple languages, domains, and data sizes demonstrate the effectiveness of our method.

1.4.2 Significance Prior Refined Attention for Language Understanding

We ease the modality and granularity inconsistency problem when distilling knowledge from the teacher understanding model to the student ones, by refining the attention hidden states based on the attention map distribution. The Spoken Language Understanding tasks including Intent Detection, Slot Filling, and Emotion Recognition are involved. We propose to apply the Attention-based Significance Priors (ASP) to improve the semantic knowledge transfer from text to speech. We further propose the Anchor-based Adaptive Span Aggregation algorithm (AASA) to narrow the modal granularity gap of alignments. To the best of our knowledge, we are the first to evaluate multiple different alignment strategies beyond vanilla global and local alignments to study the feasibility of metric-based speech-text distillations. The results on three spoken language understanding benchmarks verify our assumptions and claims.

1.4.3 Multi-Hierarchy Attention for Commonsense Reasoning

We improve the multi-source and long-term Dialogue Commonsense Reasoning process, which is a new and difficult problem in NLP, by presenting a hierarchical attention-based decoding block. We propose the first Transformer-based KG walker

that attentively reads multiscale inputs for graph decoding. Specifically, we propose Multi-source Decoding Inputs (MDI) and Output-level Length Head (OLH) to strengthen the controllability and multi-hop reasoning ability of the Hierarchical Attention-based Graph Decoder (HAGD). We further propose a two-hierarchy learning framework to train the proposed hierarchical attention-based KG walker, in order to learn both turn-level and global-level conversation topics. This is the first attempt to learn models to make natural transitions towards the global topic in KG, where we propose a distance embedding to incorporate distance information. To concurrently exploit entity and relation information when reasoning, we further propose MetaPath (MP), which has proved essential as the backbone method for KG path representation, providing a paradigm for KG reasoning. The results on the OpendialKG dataset show that HiTKG achieves a significant improvement in the performance of turn-level reasoning compared with state-of-the-art baselines. Additionally, both automatic and human evaluation show the effectiveness of the two-hierarchy learning framework for both short-term and long-term reasoning.

1.5 Outline of the Thesis

Chapter 1 introduces the background, motivation, and objective of this thesis. Further, the studied problems and research scope are clarified. The main contributions and the thesis organization are briefly stated as well.

Chapter 2 reviews the recent advances in the tasks involved to clarify the research context and development trend. The issues of the attention mechanism are also revealed and we review the important works that mitigate these problems. Moreover, the main backbone variants for sequence encoding and transduction, including Convolutional Neural Networks, Recurrent Neural Networks, Sequence-to-sequence Models, Memory Networks, Attention Mechanisms, Transformers, Pointer Net, and CopyNet, are introduced. The model backgrounds and principles are introduced in detail. In addition, the evolution of these backbone models is analyzed to reveal the relationships between them.

Chapter 3 proposes Grouped Head Attention (GHA), trained with a self-supervised group constraint that group attention heads, where each group focuses on an essential but distinctive feature subset. It additionally proposes a Voting-to-Stay (V2S)

procedure to remove redundant parameters and obtain GHA-PS (PS denotes the Pillars of Strength), thus achieving a Transformer with lighter weights. The proposed method achieves significant performance gains on three well-established tasks (Machine Translation, Language Modeling, and Abstractive Summarization) and concurrently compresses considerable parameters (the light architecture reduces 63.6% parameters against vanilla Transformer).

Chapter 4 presents the Significance Prior Refined Attention. It studies the SLU problems by metric-based knowledge distillation that aligns the embedding space of text and speech with only a small amount of data without modifying the model structure. Since the semantic and granularity gap between text and speech has been omitted in literature, which impairs the distillation effectiveness, it proposes the Prior-informed Adaptive knowledge Distillation (PAD) that adaptively leverages text/speech units of variable granularity and prior significance distributions to achieve better global and local alignments between text and speech pre-trained models. It evaluates tasks of three spoken language understanding benchmarks to show that PAD is more effective in transferring linguistic knowledge than other metric-based distillation approaches.

Chapter 5 presents HiTKG, a multi-hierarchy attention-based graph walker that leverages multiscale inputs to make precise and flexible predictions on KG paths. Furthermore, it proposes a two-hierarchy learning framework that employs two stages to learn both turn-level (short-term) and global-level (long-term) conversation topics. Specifically, in the first stage, HiTKG is trained in a supervised fashion to learn how to plan turn-level topic sequences; in the second stage, HiTKG tries to naturally approach the assigned global topic via reinforcement learning. In addition, it proposes MetaPath as the backbone method for KG path representation to exploit the entity and relation information concurrently. It further proposes Multi-source Decoding Inputs and Output-level Length Head to improve the decoding controllability. Our experiments show that HiTKG achieves a significant improvement in the performance of turn-level goal learning compared with state-of-the-art baselines. Additionally, both automatic and human evaluation prove the effectiveness of the two-hierarchy learning framework for both short-term and long-term topic reasoning.

Chapter 6 summarizes this thesis and presents some possible research trends for future work.

Chapter 2

Literature Review

With the success of deep learning and the development of computation powers, neural models have dominated almost the whole NLP community. Recently, the Transformer [5], as an attention-based neural architecture, has been the go-to model for many tasks, such as Machine Translation [5], Language Modeling [46], Dialogue Systems [13], etc.

In this chapter, we clarify the research context with a focus on deep learning-based approaches. We review the attention variants proposed in recent years that solve their respective issues. In addition, we review the core sub-symbolic-based and neural-symbolic-based NLP tasks that require different levels of semantic dependencies [47]. More specifically, for sub-symbolic-based tasks, we review Machine Translation, Language Modeling, Abstractive Summarization, and Spoken Language Understanding; for neural-symbolic-based tasks, we review Dialogue Commonsense Reasoning.

2.1 Attention Variants Regarding Their Issues

2.1.1 Complexity Issue

In the standard attention mechanism, each token attends to all positions in the input, which increases the computation complexity quadratically, thus limiting the attention's application circumstance within the tasks with short inputs. The high

time and memory complexity prohibit the model from being trained efficiently when the input is long. Thus, there evolve many variants to solving this issue.

Many observations found that the trained attention maps were very sparse, which made it possible to limit the query-key computations to make the attention efficient. Different sparse patterns were created for various sorts of data in the Sparse Transformer [48]. It employed a combination of band attention and strided attention for data with a periodic structure (such as photos). By contrast, it employed a composition of block local attention along with global attention, where global nodes were drawn from fixed points in the input sequence, for data lacking a periodic structure (such as text). Band attention and internal global-node attention were both used by Longformer [34]. For classification tasks, the global nodes were selected as $[CLS]$ tokens. To expand the receptive field without boosting computation, they also swapped out some of the band attention heads in the top layers for dilated window attention. Extended Transformer Construction (ETC) [49] used a combination of band attention and external global-node attention. To handle structured inputs and modify Contrastive Predictive Coding (CPC) [50] for pretraining, ETC additionally had a masking mechanism. BigBird [51] used additional random attention to approximate full attention in addition to the band and global attention. Their theoretical research also showed that any Turing Machine may be simulated using a sparse encoder and a sparse decoder, which accounted for the effectiveness of such sparse attention models.

The quadratic complexity of $\mathbf{QK}^T$ computation can be linearized, as long as the $softmax(\mathbf{QK}^T)$ can be disentangled into $\hat{\mathbf{Q}}\hat{\mathbf{K}}^T$ and then the attention output can be obtained by computing the production of $\mathbf{K}$ and $\mathbf{V}$ first: $\hat{\mathbf{Q}}\hat{\mathbf{K}}^T\mathbf{V} \rightarrow \hat{\mathbf{Q}}(\hat{\mathbf{K}}^T\mathbf{V})$, which leads to a complexity of $\mathrm{O}(n)$, n being the input length. Linear Transformer [18] suggested using the feature map $\phi(\mathbf{x}_i) = elu(\mathbf{x}_i) + 1$ for disentanglement. This feature map was empirically proved to perform on par with the standard Transformer, although it actually did not try to approximate the dot product attention. Random feature maps were used by Performer [52] to approximate the scoring function. Accordingly, Peng et al. [53] and Choromanski et al. [54] tried to approximate order-1 arc-cosine kernel apart from using random feature maps which approximate standard dot product attention. It was demonstrated that this feature map works well for a variety of tasks, including protein sequence modeling and MT.

The self-attention matrix was shown to be low-rank, according to Wang et al. [17]. This characteristic has two implications: (1) with parameterization, it is possible to explicitly model the low-rank property; (2) we can substitute a low-rank approximation for the self-attention matrix. Guo et al. [55] divided the self-attention into two components: a band attention that captures local dependencies and a low-rank attention with a small D_k that extracts long-range non-local interactions. In addition, the above-mentioned kernel-based methods (such as Performer) provide a low-rank matrix approximation, which was followed up by some works [56, 57].

2.1.2 Lack of Priors

Attention weights represent the attention distribution over the values. The distribution is often created from inputs (e.g., $softmax(\mathbf{QK})$ in a classic Transformer). Such a setting is not optimal since it does not utilize all possible information of a given task, e.g., the task inductive bias. There exist other sources, which we refer to as priors, that form the attention distribution. Attention distribution based on priors can be used in addition to or instead of the distribution derived from inputs. Usually, two attention distributions can be fused by first computing a weighted sum of the prior scores and the generated attention scores, and then computing the softmax function.

Some data formats, like text, can show an obvious preference for the locality. It is possible to directly encode this attribute as prior-based attention. Using a Gaussian distribution across locations would be a direct intuition. Yang et al. [58] suggested using a feedforward network to determine a central point for the mean of the Gaussian distribution. More directly, the Gaussian Transformer [59] used the position of the query as the central position and achieved comparable results.

The attention distributions in neighboring layers of the Transformer are found to be alike. Therefore, using the attention distribution from the preceding layer as a prior for attention computation makes sense. According to Predictive Attention Transformer [60], earlier attention scores were convolved by a 2D-convolutional layer and the final attention scores should be computed as a convex combination of the produced attention scores and the convolved scores. The experiments demonstrated gains above baseline models, whether they are training the model from

start or fine-tuning it on the pre-trained BERT. Realformer [61] simulated a residual skip link on attention maps by directly adding the past attention scores to the produced attention scores. For this model, they conducted pre-trainings. The experiments demonstrated that this model performed better than the baseline BERT in many datasets and beat the baseline model even when pre-training budgets are much smaller. More radically, Lazyformer [62] suggested sharing attention maps between many nearby layers. The advantage of this method is that the computing cost decreased because the attention maps were generated just once and repeatedly utilized in subsequent layers. Their pre-training experiments demonstrated that the final model was still useful while having a substantially higher computational efficiency.

Some studies drew attention from other sources, which performed comparatively to the attention solely based on inputs. Average Attention Network, proposed by Zhang et al. [63], is an effective Transformer decoder that only draws attention distribution from discrete uniform distributions. As a result, the values are aggregated as a cumulative average of all values. On top of the average attention module, they also added a feed-forward gating layer to increase the network's expressiveness. This method avoids the $O(n^2)$ complexity in decoding by allowing the customized Transformer decoder to train in parallel like normal Transformers do and decode similarly to an RNN. According to Synthesizer [64], generated attention scores can be replaced by: (1) learnable attention scores with random initialization; and (2) attention scores produced by a feed-forward network that is solely conditioned on the querying input itself. These variations performed on par with the original Transformer, according to evaluations on MT and LM. Although the empirical findings are fascinating, it is not clear why these variants are effective.

2.1.3 The Parallel Multi-Head Mechanism

To enable the model to jointly attend to input from several representation subspaces at various positions is a fundamental justification for adopting the multi-head design in Multi-Head Attention (MHA) [5]. To ensure distinct behavior between attention heads, however, or for heads to communicate with one another, it requires explicit methods apart from the original parallel design of the vanilla MHA. By providing more complex mechanisms that direct the activity of attention heads

or permit interaction between attention heads, this field of research aims to enhance multi-head processes.

Kovaleva et al. [65] identified numerous inappropriate attention patterns in BERT. For instance, many attention heads merely focused on the $[CLS]$ and $[SEP]$ special tokens. In order to optimize the labor division among attention heads, additional mechanisms should be introduced. Deshpande and Narasimhan [66] suggested using an auxiliary loss, defined as the Frobenius norm between attention distribution maps and predetermined attention patterns, to achieve this goal. Li et al. [26] added extra disagreement regularization terms to the loss function to increase diversity among the several attention heads. The first two regularization terms maximized the cosine distances between the input subspaces and the output representations, while the last regularization term diversifies the locations that the heads attend to by performing element-wise multiplications between head attention matrices. By using a talking head method, Talking-head Attention [67] linearly projected the produced attention scores from h_k to h heads, applied softmax in that space, and then projected to h_v heads for value aggregation. The goal was to support the model's ability to switch between attention heads in a teachable manner.

The redundancy problem of MHA arises recently. Michel et al. [15] and Voita et al. [16] found that only a subset of the attention heads have significant utilities in the Transformer, while others could be pruned, where the important heads could be identified by Expected Sensitivity and Layer-wise Relevance Propagation (LRP) [68]. Upon this, Li et al. [69] learned per-head importance scores and pruned the heads. Cordonnier et al. [70] homogenized the attention heads by sharing a part of the weights between heads, which lowered the number of parameters but sacrificed performance. As mentioned above, Li et al. [26] found that diversifying attention heads by adding regularization terms can force MHA to reduce inter-head redundancy, yielding performance gains for Machine Translation.

2.2 Machine Translation

The study of Machine Translation (MT) has a lengthy history, dating back to the 17th century. Rene Descartes developed a global language in 1629 that used one

set of signs to convey the same meaning across several languages. The first researcher on the subject, Yehoshua Bar-Hillel, started his study at MIT in 1951 and convened the first International Conference on Machine Translation in 1952, which is when the particular research on MT started. Rule-based Machine Translation [71], Statistical Machine Translation [72], and Neural Machine Translation (NMT) [73] are the three main streams of MT research that have occurred since. In this thesis, we selectively review the recent advances of NMT.

Transformer has greatly improved the performance of various tasks, therefore researchers are paying close attention to its applications on MT. The widely acknowledged shortcomings of the standard Transformer include the absence of recurrence modeling, the fact that it is not theoretically Turing-complete, the difficulty of capturing positional information, and the complexity of the model. The performance of its translation has been hampered by all these flaws. Modifications have been proposed as a solution to these issues in order to obtain better performance.

Some works made changes to the model design concentrating on the network composition and the depth of the attention layer. With a more sophisticated attention mechanism, Bapna et al. [74] proposed 2-3x deeper Transformer. The model is able to propagate the gradient flow to various encoder layers because the enhanced attention mechanism extends its connection to each encoder layer, similar to weighted residual connections throughout the encoder depth. Similar to this, Wang et al. [75] suggested a deeper Transformer model (25 layers of encoder), which was built on the above-mentioned work [74] by appropriately implementing layer normalization and employing a special output aggregation method.

In contrast to the fixed-layer NMT models, Dehghani et al. [76] proposed Universal Transformers that dynamically adjusted the number of layers, which improved the original self-attention-based representation for better recursive transformations. It combines recurrent inductive bias of RNNs and Adaptive Computation Time Halting mechanism. Notably, this modification has allowed the model to be demonstrated as Turing-complete under specific assumptions.

While the majority of modifications concentrate on directly modifying the model structure, some recent work has opted to use an alternative input representation for MT in order to enhance the model's performance. Since the vanilla Transformer has difficulty in attending to positional information, one straightforward solution is

employing improved Positional Encoding for sequence order injection. Shaw et al. [77] augmented the self-attention mechanism with knowledge of relative locations, improving the performance on two MT tasks.

Meanwhile, attempts have been made to use pre-initialized input representations for various NLP tasks. Edunov et al. [78] used a pre-trained ELMo [79] for the encoder of NMT model. In addition, the Transformer-based contextual input representations, such as BERT (Bidirectional Encoder Representation from Transformers) [80] and GPT (Generative Pre-trained Transformer) [81], have been presented, which were applied to bring improvements to MT [82].

2.3 Language Modeling

Language Modeling (LM) is the most common self-supervised task in NLP. The traditional Statistical Language Modeling predicts the probability distribution based on the word frequency in the corpus. Recently, the neural approaches have greatly boosted the development of LM in the NLP community. The neural approaches are able to assign high probabilities for those linguistically flawless sequences that never exist in the training corpus.

Note that the recent advances in Language Modeling are also greatly empowered by the models discussed in the last section [27, 36, 80, 81]. To avoid repeated discussions, we focus on the advances in Masked Language Modeling (MLM) of the LM field, which is also one of the most commonly adopted pre-training tasks at present [80]. The idea of MLM, often known as a Cloze problem, was initially put forward by Taylor [83]. This task was modified by Devlin et al. [80] as a brand-new pre-training task to address the problem of the traditional unidirectional LM. MLM trains the model to predict the tokens that have been masked out of the input phrases based on the remaining tokens, where the masked tokens are replaced with the special token $[MASK]$. The $[MASK]$ token does not present during the fine-tuning phase, hence this pre-training approach will result in a mismatch between the pre-training and fine-tuning phases. To mitigate this mismatch, BERT [80] performed masking 80% of the time using a specific $[MASK]$ token, 10% of the time using a random token, and 10% of the time using the original token.

Typically, MLM is resolved as a classification task. In order to predict the masked token, we send the masked sequences to a neural encoder, such as BERT, whose output vectors are then fed into a softmax classifier. Apart from such encoder-only methods, the MLM can be performed with an encoder-decoder architecture as well, also known as a sequence-to-sequence architecture (introduced in §2.7.2. As such, the encoder receives a masked sequence and the decoder successively generates the masked tokens in an auto-regressive fashion. Such an approach, which is usually referred to as sequence-to-sequence MLM or Seq2Seq MLM, was utilized in some famous works such as MASS [84] and T5 [85]. Moreover, the Seq2Seq-style downstream tasks, such as Question Answering, Summarization, and Machine Translation, can benefit from Seq2Seq MLM.

There were several studies that proposed improved MLM variations to boost BERT. RoBERTa [86] enhanced BERT with dynamic masking in contrast to static masking. UniLM [87] augmented the MLM with unidirectional, bidirectional, and sequence-to-sequence language modeling. Translation Language Modeling (TLM) was proposed in XLM [88], which performed MLM on a concatenation of parallel bilingual sentence pairs. To consider structural information in pre-training, SpanBERT [89] substituted MLM with Random Contiguous Words Masking and Span Boundary Objective (SBO), which prompted the model to predict masked spans based on span boundaries. Additionally, the Span Order Recovery task was introduced by StructBERT [90] to further include linguistic structures.

2.4 Abstractive Summarization

The goal of Abstractive Summarization (AS) is to produce brief, accessible phrases that convey the essence of the source texts. Due to the introduction of Seq2Seq models [91] and the attention mechanism [4], neural networks have shown impressive results in AS. A pointer network was used by See et al. [92], Paulus et al. [93], and Gehrmann et al. [94] to address the out-of-vocabulary problem. Additionally, to avoid repetition, See et al. [92] applied a coverage mechanism [95]; Paulus et al. [93] and Chen and Bansal [96] performed reinforcement learning in an end-to-end setting. Pre-trained language models have recently made great progress in a number of NLP tasks [80, 97, 98]. The use of pre-trained language models in the

AS task has been the core of several research that yielded state-of-the-art results [99–102].

The issue of factual inconsistency in AS models has been brought up in a number of works [103–106]. Similar to a natural language inference job, Kryscinski et al. [103] suggested training a neural network to determine if a summary is factually compatible with a particular source material. Li et al. [107] suggested employing unlikelihood to suppress logically incorrect replies in the dialogue generation task. The strategy of Nan et al. [108] that enhanced entity-level metrics of summaries is also in line with controlled AS [109], where a user may specify a list of named entities they wish to appear in the summary by passing it as input.

2.5 Spoken Language Understanding

2.5.1 Intent Detection

The Intent Detection approaches classify the intent of users given their utterances. [110] and [111] were the first who greatly improved the recognition accuracy of dialogue intent. They built deep convex networks to combine the predictions of a prior network and the current utterances as an integrated input of a current network. A deep learning framework was also applied by Yann et al. [112] to classify the intent in a semi-supervised fashion. To solve the difficulty of training a deep neural network for intent prediction, Restricted Boltzmann Machine (RBM) and Deep Belief Networks (DBNs) were applied to initialize the parameters of deep neural networks [113]. To make use of the strengths of RNNs in sequence processing, some works used RNNs as utterance encoders and made predictions for intent and domain categories [114, 115]. [116] used a CNN to extract hierarchical features for intent detection and illustrated the sequence classification capabilities of CNNs. [117] proposed a model for intent classification of short utterances. Short utterances are hard for intent detection because of the lack of information in a single dialogue turn. This paper used RNN and CNN architectures to incorporate the dialogue history, thus obtaining the context information as an additional input besides the current turn's message. The model achieved promising performances on three intent classification datasets. More recently, [118] pre-trained Task-Oriented

Dialogue BERT (TOD-BERT) and significantly improved the accuracy of the intent detection task. The proposed model also exhibited a strong capability of few-shot learning and could effectively alleviate the data insufficiency issue in a specific domain. [119] introduced duel sentence encoders for efficient intent detection. Their methods were effective in low-resource situations. [120] proposed an NLU framework for argumentative dialogue systems in the information-seeking and opinion-building domain. They used a BERT+BiLSTM to inject commonsense knowledge into the framework to better understand the user intent. [121] combined Transformer with capsule networks, and found their model achieved better performance than original capsule-NLU network implementations.

2.5.2 Slot Filling

The SF problem is also called semantic tagging, a sequence tagging problem. It is more challenging for that the model needs to predict multiple objects at a time. DBNs exhibited promising capabilities in the learning of deep architectures and have been applied in many tasks including slot filling. [122] used a DBN-initialized neural network to complete slot filling in the call-routing task. [123] built a DBN-based sequence tagger. In addition to the NER input features used in traditional taggers, they also combined part of speech (POS) and syntactic features as a part of the input. The recurrent architectures benefited the SF task in that they could keep track of the information along past timesteps to make the most of the sequential information. [124] first argued that instead of simply predicting words, RNN Language Models (RNN-LMs) could be applied in slot filling. On the output side of RNN-LMs, tag labels were predicted instead of normal vocabularies. [125] and [126] further investigated the impact of different recurrent architectures in the Slot Filling task and found that all RNNs outperformed the Conditional Random Field (CRF) baseline. As a powerful recurrent model, LSTM showed promising tagging accuracy on the ATIS dataset owing to the memory control of its gate mechanism [127]. [128] argued that the shallow output representations of traditional slot filling lacked the ability to represent the structured dialogue information. To improve, they treated the Slot Filling task as a template-based tree decoding task by iteratively generating and filling in the templates. Different from traditional SF methods, [129] tackled the task by treating it as a turn-based span extraction task. They applied the conversational pre-trained model ConveRT and

utilized the rich semantic information embedded in the pre-trained vectors to solve the problem of in-domain data insufficiency. The inputs of ConveRT were the requested slots and the utterance, while the output was a span of interest as the slot value.

2.5.3 Emotion Recognition

Emotion Recognition aims to classify the emotion polarities of the given input. The emotion does not evenly spread across every input timestep. However, in traditional deep learning algorithms for Emotion Recognition, all places of a particular sentence receive equal attention. In the attention mechanism, the model focuses on some specific positions of the provided samples that are emotionally salient, based on the attention weights given to each position of the data. Bidirectional LSTM using a weighted-polling strategy was proposed by Mirsamadi et al. [130] in an effort to identify more insightful aspects of emotion as opposed to conventional Low-Level Descriptors (LLD) and High-level Statistical aggregation Functions (HSF). This technique was inspired by the attention mechanism, which enables the network to concentrate on emotionally significant sentence fragments while ignoring silent acoustic frames. Its experiments on the IEMOCAP dataset demonstrated that weighted pooling with local attention may outperform LSTM with mean pooling, by balancing short-term characteristics at the frame level and long-term aggregation at the utterance level. Later in 2019, Li et al. [131] introduced a self-attentional CNN-BLSTM that outperformed the multi-channel CNN [132] on the IEMOCAP dataset by 7.7%, which leveraged the attention mechanism and multitask learning. Self-attention was used to enable the CNN-BLSTM model to concentrate on the emotionally salient time steps and gender categorization was used as an auxiliary task. A model based on a modified LSTM was proposed by Xie et al. [133]. Their approaches were able to minimize the computational complexity by replacing the LSTM's forgetting gate with an attention gate. By using the attention mechanism on both the time and feature dimensions as opposed to only forwarding the results of the previous LSTM iteration, they further improved the representations. In this study, the output of the LSTM was conditioned on a number of time steps produced by the attention mechanism, as opposed to only being determined by the outcome of the previous step. The features were then treated in a similar way by forwarding them to the fully connected layer for classification.

2.6 Dialogue Commonsense Reasoning

In all facets of Artificial Intelligence, from NLP to Computer Vision (CV), commonsense knowledge reasoning is essential. According to Davis and Marcus [134], there are many difficulties in commonsense reasoning, ranging from the difficulty in understanding and formulating commonsense knowledge for particular or general domains, to the complexity in different types of reasoning and their integration for problem-solving. Davis and Marcus [134] argued that in order to advance the field, the community needs methods that can combine various forms of reasoning (such as symbolic reasoning through deduction and statistical reasoning based on a lot of data), as well as benchmarks and evaluation metrics that can quantitatively assess the work of this field.

In this thesis, we study the neural-symbolic methods for Dialogue Commonsense Reasoning (DCR). It is a new field of study that reasons the dialogue process based on commonsense knowledge and dialogue history [22]. The commonsense knowledge of most recent works on DCR is based on the symbolic information of knowledge graphs. In general, these studies can be divided into two groups based on the depth of the candidate knowledge explored.

By combining their shallow (i.e., 1-hop or 2-hop) neighborhood information in the knowledge graph, the first line of studies, termed as breadth-centric techniques, tend to focus on enhancing dialog context with entity representations [43, 135–137]. Zhou et al. [138] propose to attentively attend to all 1-hop relations of each initial entity that appears in the user's utterance in order to encode an auxiliary knowledge vector. The knowledge encoding technique proposed by Zhang et al. [139] expanded the work to 2-hop relations, encoding all starting entities and their 1-hop neighbors using two separate attention processes. These works are effective at finding appropriate relationships of the given entities inside the knowledge graph, but they fall short when it comes to fetching a narrow set of knowledge pathways specifically relevant to the conversation or generalizing to multi-hop interactions.

The second line of work, however, uses a depth-centric search across potential knowledge pathways. They focus on exploring just a narrow range of entities and relations that are specifically useful for response generation, rather than augmenting entity representation with a shallow but broad range of knowledge. A policy network was used by Liu et al. [140] to traverse the KG, which was formulated as a

Partially Observed Markov Decision Process. Moon et al. [22] proposed a recurrent route decoder that selected the following item from a set of accessible nodes using a hidden state vector. These models are capable of inferring multi-hop relations, but they disregard the rich relational information of nodes and edges that they do not explicitly choose to traverse. Jung et al. [21] compensated for the flaw by not pre-selecting an ideal node; instead, it spread attention to every entity that can be reached before determining an ideal path from the output attention distribution. The KG walker in Jung et al. [21] was good at achieving optimal performance since they reserved all potential paths, but such a mechanism is too high in computation complexity to be scalable to multi-hop reasoning. In addition, these reasoning strategies neglected the hierarchical structure of the input source and made separate predictions for entity and relation paths, which affected their performance. Ni et al. [13] mitigated these issues by proposing a hierarchical attention-based Transformer that attentively leveraged the hierarchical dialogue history and KG information with moderate computational complexity, being scalable to lengthy sequences.

2.7 Backbones for Sequence Encoding and Transduction

There have evolved many architectures [47] in the NLP community. Despite the tremendous amount, most of them are based on Convolutional Neural Networks, Recurrent Neural Networks, or Attention, which are the three main backbone models. Some of them [141], though being proposed earlier in time, still play an important role in this field.

In this section, we introduce the main backbone variants for natural language encoding and transduction, including Convolutional Neural Networks, Recurrent Neural Networks, Sequence-to-sequence Models, Memory Networks, Attention Networks, Transformers, Pointer Net, and CopyNet. The model background and principles are introduced in detail. In addition, the evolution of these backbone models is analyzed to reveal the relationships between them.

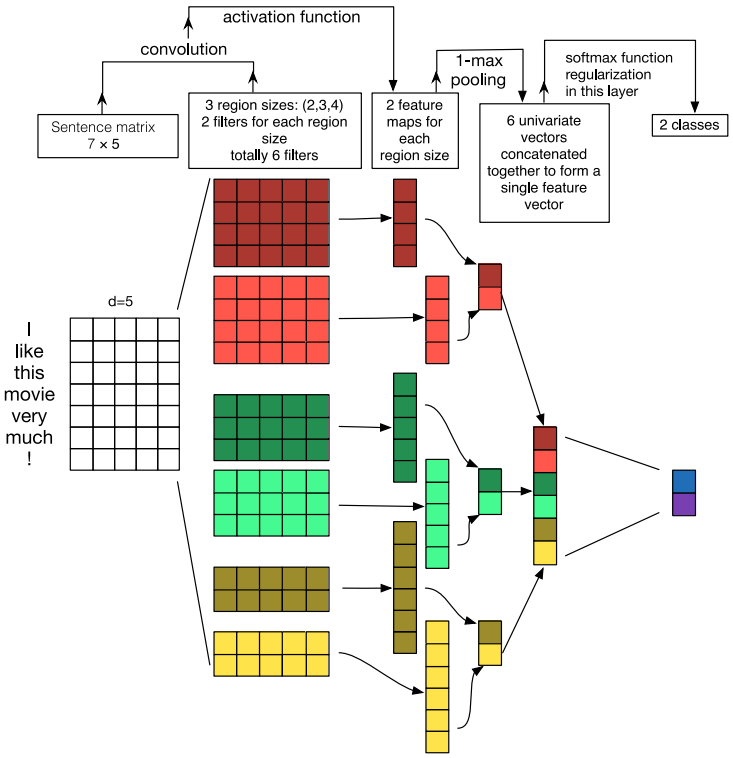


FIGURE 2.1: A CNN architecture for text classification [1]

2.7.1 Convolutional Neural Networks

Deep neural networks have been considered as one of the most powerful models. 'Deep' refers to the fact that they are multilayer, which extracts features by stacking feed-forward layers. Feed-forward layers can be defined as: $y = \sigma(Wx + b)$. Where the σ is an activation function; W and b are trainable parameters. The feed-forward layers are powerful due to the activation function, which makes the otherwise linear operation, non-linear. Whereas there exist some problems when using feed-forward layers. Firstly, the operations of feed-forward layers or multilayer neural networks are just template matching, where they do not consider the specific structure of data. Furthermore, the fully connected mechanism of traditional multilayer neural networks causes an explosion in the number of parameters and thus leads to generalization problems. [142] proposed LeNet-5, an early Convolutional Neural Network (CNN). The invention of CNNs mitigates the above problems to some extent.

CNNs (Figure 2.1) usually consist of convolutional layers, pooling layers, and feed-forward layers. Convolutional layers apply convolution kernels to perform the convolution operation:

$$G(m,n) = (f * h)(m,n) = \sum_{j} \sum_{k} h(j,k) f(m-j, n-k) \tag{2.1}$$

Where m and n are respectively the indexes of rows and columns of the result matrix. f denotes the input matrix and h denotes the convolutional kernel. The pooling layers perform down-sampling on the result of convolutional layers to get a higher level of features and the feed-forward layers map them into a probability distribution to predict class scores.

A sliding window feature enables convolution layers to capture local features and the pooling layers can produce hierarchical features. These two mechanisms give CNNs the local perception and global perception ability, helping to capture some specific inner structures of data. The parameter sharing mechanism eases the parameter explosion problem and overfitting problem because the reduction of trainable parameters leads to less model complexity, improving the generalization ability.

Due to these good properties, CNNs have been widely applied in many works. Among them, the Computer Vision tasks benefit the most for that the Spatio-temporal data structures of images or videos are perfectly captured by CNNs. For more detailed mechanism illustrations and other variants of CNNs, readers can refer to these representative algorithm papers or surveys: [143–149].

Recent years have seen a dramatic increase in applications of CNNs in NLP. Many tasks take words as basic units. However, phrases, sentences, or even paragraphs are also useful for semantic representations. As a result, CNNs are an ideal tool for the hierarchical modeling of language [150].

2.7.2 Recurrent Neural Networks and Sequence-to-sequence Models

NLP tasks including dialogue-related tasks try to process and analyze sequential language data points. Even though standard neural networks, as well as CNNs, are powerful learning models, they have two main limitations [151]. One is that they assume the data points are independent of each other. While it is reasonable if the data points are produced independently, essential information can be missed when processing interrelated data points (e.g., text, audio, video). Additionally, their inputs are usually of fixed length, which is a limitation when processing sequential data varying in length. Thus, a sequential model being able to represent the sequential information flow is desirable.

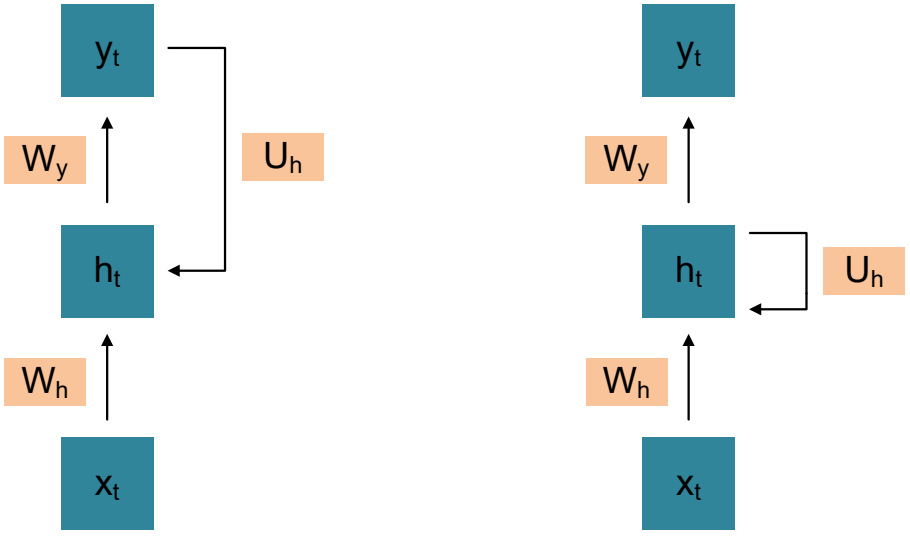
Markov models like Hidden Markov Models (HMMs) are traditional sequential models, but due to the time complexity of the inference algorithm [152] and because the size of the transition matrix grows significantly with the increase of the discrete state space, in practice they are not applicable in dealing with problems involving large possible hidden states. The property that the hidden states of Markov models are only affected by the immediate hidden states further limits the power of this model.

RNN models are not proposed recently, but they greatly solve the above problems and some variants can amazingly achieve state-of-the-art performance in dialogue-related tasks as well as many other NLP tasks. The inductive bias of recurrent models is non-replaceable in many scenarios, and many up-to-date models incorporate the recurrence.

2.7.2.1 Jordan-Type and Elman-Type RNNs

In 1982, Hopfield introduced an early family of RNNs to solve pattern recognition tasks [153]. [154] and [155] introduced two kinds of RNN architectures respectively. Generally, modern RNNs can be classified into Jordan-type RNNs and Elman-type RNNs.

The Jordan-type RNNs are shown in Figure 2.2a. x_t, h_t, and y_t are the inputs, hidden state, and output of time step t, respectively. W_h, W_y, and U_h are weight



(A) Jordan-type RNNs (B) Elman-type RNNs

FIGURE 2.2: Graphical models of two basic types of RNNs

matrices. Each update of the hidden state is decided by the current input and the output of the last time step while each output is decided by the current hidden state. Thus the hidden state and output of time step t can be calculated as:

$$h_t = \sigma_h(W_h x_t + U_h y_{t-1} + b_h) \tag{2.2}$$

$$y_t = \sigma_y(W_y h_t + b_y) \tag{2.3}$$

Where b_h and b_y are biases. σ_h and σ_y are activation functions.

The Elman-type RNNs are shown in Figure 2.2b. The difference is that each hidden state is decided by the current input and the hidden state of the last time step. Thus the hidden state and output of time step t can be calculated as:

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \tag{2.4}$$

$$y_t = \sigma_y(W_y h_t + b_y) \tag{2.5}$$

Simple RNNs can model long-term dependencies theoretically. But in practical training, long-range dependencies are difficult to learn [156, 157]. When backpropagating errors over many time steps, simple RNNs suffer from problems known as gradient vanishing and gradient explosion [141]. Some solutions were proposed

to solve these problems [158, 159], which led to the invention of some variants of traditional recurrent networks.

2.7.2.2 LSTM

[141] introduced gate mechanisms in LSTM mainly to address the gradient vanishing problem. Input gate, forget gate, and output gate were introduced to decide how much information from new inputs and past memories should be reserved. The model can be described by the following equations:

$$\hat{h}^{(t)} = tanh\left(W^{\hat{h}x}x^{(t)} + W^{\hat{h}h}h^{(t-1)} + b_{\hat{h}}\right) \tag{2.6}$$

$$i^{(t)} = \sigma\left(W^{ix}x^{(t)} + W^{ih}h^{(t-1)} + b_i\right) \tag{2.7}$$

$$f^{(t)} = \sigma\left(W^{fx}x^{(t)} + W^{fh}h^{(t-1)} + b_f\right) \tag{2.8}$$

$$o^{(t)} = \sigma\left(W^{ox}x^{(t)} + W^{oh}h^{(t-1)} + b_o\right) \tag{2.9}$$

$$s^{(t)} = \hat{h}^{(t)} \odot i^{(t)} + s^{(t-1)} \odot f^{(t)} \tag{2.10}$$

$$h^{(t)} = tanh(s^{(t)}) \odot o^{(t)} \tag{2.11}$$

Where t represents time step t. i, f, and o are gates, denoting input gate, forget gate, and output gate respectively. x, $\hat{h}$, s, and h are input, short-term memory, long-term memory, and output respectively. b is bias and W is weight matrix. $\odot$ denotes element-wise multiplication.

The intuition of the term "Long Short-Term Memory" is that the proposed model applies both long-term and short-term memory vectors to encode the sequential data, and uses gate mechanisms to control the information flow. The performance of LSTM is impressive since it achieved state-of-the-art results in many NLP tasks as a backbone model although this model was proposed in 1997.

2.7.2.3 GRU

Inspired by the gating mechanism, [160] proposed Gated Recurrent Unit (GRU), which can be modeled by the equations:

$$z^{(t)} = \sigma\left(W^z x^{(t)} + U^z h^{(t-1)} + b_z\right) \tag{2.12}$$

$$r^{(t)} = \sigma\left(W^r x^{(t)} + U^r h^{(t-1)} + b_r\right) \tag{2.13}$$

$$\hat{h}^{(t)} = tanh\left(W^h x^{(t)} + U^h(r^{(t)} \odot h^{(t-1)}) + b_h\right) \tag{2.14}$$

$$h^{(t)} = (1 - z^{(t)}) \odot h^{(t-1)} + z^{(t)} \odot \hat{h}^{(t)} \tag{2.15}$$

Where t represents time step t. z and r are gates, denoting the update gate and reset gate respectively. x, $\hat{h}$ and h are input, candidate activation vector and output respectively. b is bias while W and U are weight matrices. $\odot$ denotes element-wise multiplication.

LSTM and GRU, as two types of gating units, are very similar to each other [161]. The most prominent common point between them is that from time step t to time step $t+1$, an additive component is introduced to update the state whereas simple RNNs always replace the activation. Both LSTM and GRU keep certain old components and mix them with new contents. This property enables the units to remember the information of history steps farther back and, more importantly, avoid gradient vanishing problems when backpropagating the error.

There also exist several differences between them. LSTM exposes its memory content under the control of the output gate, while the same content in GRU is in an uncontrolled manner. Additionally, different from LSTM, GRU does not independently gate the amount of new memory content being added. And if looking from the experimental perspective, GRU has fewer parameters, which contributes to its faster convergence and better generalization ability. It has also been shown that GRU can achieve better performance in smaller datasets [161]. However, [162] showed that LSTM cells exhibited consistently better performance in a large-scale analysis of Neural Machine Translation.

2.7.2.4 Bidirectional Recurrent Neural Networks

In sequence learning, not only the past information is essential to the model inference, but the future information should also be considered to achieve a better inference ability. [163] proposed the bi-directional recurrent neural networks (BRNNs), which had two kinds of hidden layers: the first encoded information from past time steps while the second encoded information in a flipped direction. The model can be described using the equations:

$$h^{(t)} = \sigma\left(W^{hx}x^{(t)} + W^{hh}h^{(t-1)} + b_h\right) \tag{2.16}$$

$$z^{(t)} = \sigma\left(W^{zx}x^{(t)} + W^{zz}z^{(t+1)} + b_z\right) \tag{2.17}$$

$$\hat{y}^{(t)} = softmax\left(W^{yh}h^{(t)} + W^{yz}z^{(t)} + b_y\right) \tag{2.18}$$

Where h and z are the two hidden layers. Other variables are defined in the same way as in the case of LSTMs and GRUs.

2.7.2.5 Vanilla Sequence-to-sequence Models (Encoder-decoder Models)

[164] first proposed the sequence-to-sequence model to solve the machine translation tasks. The sequence-to-sequence model aimed to map an input sequence to an output sequence by first using an encoder to map the input sequence into an intermediate vector and a decoder further generated the output based on the intermediate vector and history generated by the decoder. The equations below illustrate the encoder-decoder model:

$$Encoder : h_t = E(h_{t-1}, x_t) \tag{2.19}$$

$$Decoder : y_t = D(h_t, y_{t-1}) \tag{2.20}$$

Where t is the time step, h is the hidden vector and y is the output vector. E and D are the sequential cells used by the encoder and decoder respectively. The last hidden state of the encoder is the intermediate vector, and this vector is usually

used to initialize the first hidden state of the decoder. At encoding time, each hidden state is decided by the hidden state of the previous time step and the input at the current time step, while at decoding time, each hidden state is decided by the current hidden state and the output of the previous time step.

This model is powerful because it is not restricted to fixed-length inputs and outputs. Instead, the length of the source sequence and target sequence can differ. Based on this model, many more advanced sequence-to-sequence models have been developed, which will be discussed in this and subsequent sections.

2.7.2.6 Hierarchical Recurrent Encoder-Decoder (HRED)

Hierarchical Recurrent Encoder-Decoder (HRED) is a context-aware sequence-to-sequence model. It was first proposed by [165] to address the context-aware online query suggestion problem. It was designed to be aware of historical queries and the proposed model can provide rare and high-quality results.

With the popularity of the sequence-to-sequence model, [2] extended HRED to the dialogue domain and built an end-to-end context-aware dialogue system. HRED achieved noticeable improvements in dialogue and end-to-end question answering. This work attracted even more attention than the original paper for that dialogue systems are a perfect setting for the application of HRED. Traditional dialogue systems [166] generated responses based on the single-turn messages, which sacrificed the information in the dialogue history. [167] combined dialogue history turns with a window size of 3 as the input of a sequence-to-sequence model for response generation, which is limited as well for that they encode the dialogue history only in token-level. The "turn-by-turn" characteristic of dialogue indicated that the turn-level information also matters. The HRED learned both token-level and turn-level representation, thus exhibiting promising dialogue context awareness.

Figure 2.3 represents the HRED in a dialogue setting. HRED models the token-level and turn-level sequences hierarchically with two levels of RNNs: a token-level RNN consisting of an encoder and a decoder, and a turn-level context RNN. The encoder RNN encodes the utterance of each turn token by token into a hidden state. This hidden state is then taken as the input of the context RNN at each turn-level time step. Thus the turn-level context RNN iteratively keeps track of the history

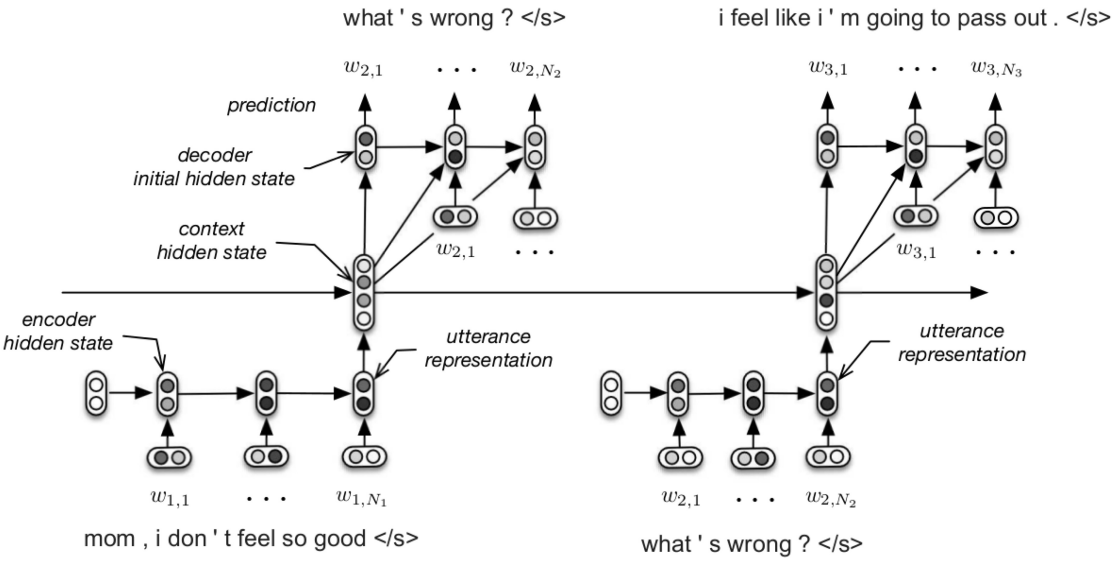


FIGURE 2.3: The HRED model in a dialogue setting [2]

utterances. The hidden state of context RNN at turn t represents a summary of the utterances up to turn t and is used to initialize the first hidden state of decoder RNN, which is similar to a standard decoder in sequence-to-sequence models [164]. All of the three RNNs described above apply GRU cells as the recurrent unit, and the parameters of the encoder and decoder are shared for each utterance.

[168] further proposed Latent Variable Hierarchical Recurrent Encoder-Decoder (VHRED) to model complex dependencies between sequences. Based on HRED, VHRED combined a latent variable into the decoder and turned the decoding process into a two-step generation process: sampling a latent variable at the first step and then generating the response conditionally. VHRED was trained with a variational lower bound on the log-likelihood and exhibited promising improvement in the diversity, length, and quality of generated responses.

2.7.3 Memory Networks

Memory is a crucial component when addressing problems regarding past experiences or outside knowledge sources. The hippocampus of human brains and the hard disk of computers are the components that humans and computers depend on for reading and writing memories. Traditional models rarely have a memory

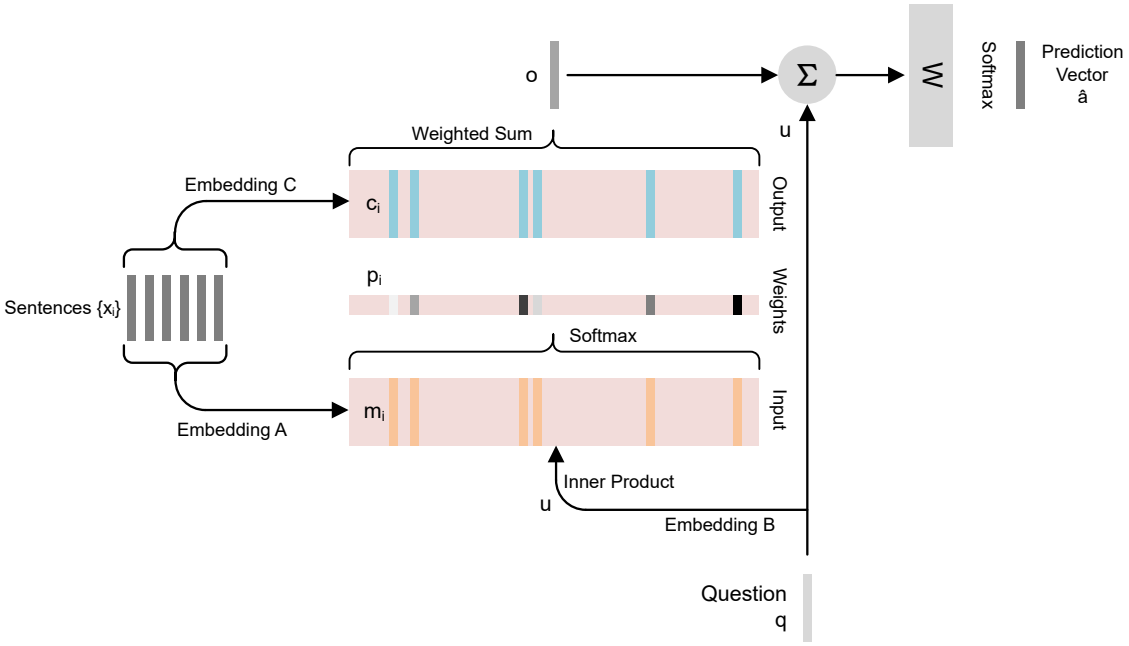


FIGURE 2.4: The structure of end-to-end memory networks [3]

component, thus lacking the ability of knowledge reusing and reasoning. RNNs iteratively pass history information across time steps, which, to some extent, can be viewed as a memory model. However, even for LSTM, which is a powerful variant of RNN equipped with long-term and short-term memory, the memory module is too small and facts are not explicitly discriminated, thus not being able to compress specific knowledge facts and reuse them in tasks.

[169] proposed memory networks, a model that is endowed with a memory component. As described in their work, a memory network has five modules: a memory module that stores the representations of memory facts; an 'I' module that maps the input memory facts into embedded representations; a 'G' module that decides the update of the memory module; an 'O' module which generates the output conditioned on the input representation and memory representation; an 'R' module which organizes the final response based on the output of 'O' module. This model needs a strong supervision signal for each module and thus is not practical to train in an end-to-end fashion.

[3] extended their prior work to an end-to-end memory network, which was commonly accepted as a standard memory network being easy to train and apply.

Figure 2.4 represents the proposed end-to-end memory networks. Its architecture consists of three stages: weight calculation, memory selection, and final prediction.

Weight calculation. The model first converts the input memory set $\{x_i\}$ into memory representations $\{m_i\}$ using a representation model A. Then it maps the input query into its embedding space using another representation model B, obtaining an embedding vector u. The final weights are calculated as follows:

$$p_i = Softmax(u^T m_i) \tag{2.21}$$

Where p_i is the weight corresponding to each input memory x_i conditioned on the query.

Memory selection. Before generating the final prediction, a selected memory vector is generated by first encoding the input memory x_i into an embedded vector c_i using another representation model C, then calculating the weighted sum over the $\{c_i\}$ using the weights calculated in the previous stage:

$$o = \sum_i p_i c_i \tag{2.22}$$

Where o represents the selected memory vector. This vector cannot be found in memory representations. The soft memory selection facilitates differentiability in gradient computing, which makes the whole model end-to-end trainable.

Final prediction. The final prediction is obtained by mapping the sum vector of the selected memory o and the embedded query u into a probability vector $\hat{a}$:

$$\hat{\alpha} = Softmax(W(o + u)) \tag{2.23}$$

2.7.4 Attention and Transformer

As introduced in § 2.7.2, traditional sequence-to-sequence models decode the token conditioning on the current hidden state and output vector of the last time step,

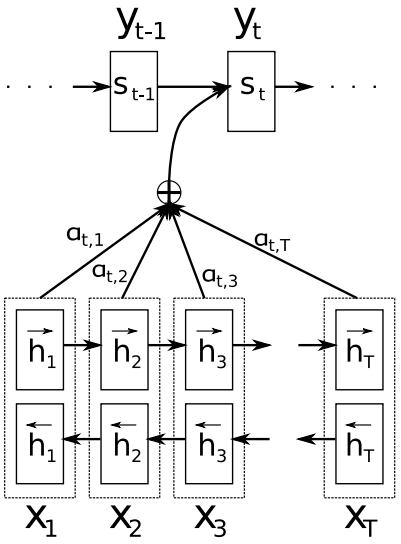


FIGURE 2.5: The graphical illustration of the attention model [4]

which is formulated as:

$$P(y_i|y_1, ..., y_{i-1}, x) = g(y_{i-1}, h_i) \tag{2.24}$$

Where g is a sequential model which maps the input vectors into a probability vector.

However, such a decoding scheme is limited when the input sentence is long. RNNs are not able to encode all information into a fixed-length hidden vector. [73] proved via experiments that a sequence-to-sequence model performed worse when the input sequence got longer. Also, for the limited-expression ability of a fixed-length hidden vector, the performance of the decoding scheme in Equation (2.24) largely depends on the first few steps of decoding, and if the decoder fails to have a good start, the whole sequence would be negatively affected.

2.7.4.1 Attention

[4] proposed the attention mechanism in the machine translation task. They described the method as "jointly align and translate", which illustrated the sequence-to-sequence translation model as an encoder-decoder model with attention. At the decoding stage, each decoding state would consider which parts of the encoded source sentence are correlated, instead of depending only on the immediate prior

output token. The output probability distribution can be described as:

$$P(y_i|y_1, ..., y_{i-1}, x) = g(y_{i-1}, s_i, c_i) \tag{2.25}$$

Where i denotes the i^{th} time step; y_i is the output token, s_i is the decoder hidden state and c_i is the weighted source sentence:

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \tag{2.26}$$

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{2.27}$$

Where α_{ij} is the normalized weight score:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{2.28}$$

e_{ij} is the similarity score between s_{i-1} and j^{th} encoder hidden state h_j, where the score is predicted by the similarity model a:

$$e_{ij} = a(s_{i-1}, h_j) \tag{2.29}$$

Figure 2.5 illustrates the attention model, where t and T denote the time steps of the decoder and encoder respectively.

Memory networks are similar to attention networks in the way they operate, except for the choice of the similarity model. In memory networks, the encoded memory can be viewed as the encoded source sentence in attention. However, the memory model proposed by [3] chose cosine distance as the similarity model while the attention proposed by [4] used a feed-forward network which is trainable together with the whole sequence-to-sequence model.

2.7.4.2 Transformer

Before Transformers, most works combined attention with recurrent units, except for a few works such as [170] and [171]. Recurrent models condition each hidden state on the previous hidden state and the current input and are flexible in sequence length. However, due to their sequential nature, recurrent models cannot be trained

in parallel, which severely undermines their potential. [5] proposed Transformer, which entirely utilized attention mechanisms without any recurrent units and deployed more parallelization to speed up training. It applied self-attention and encoder-decoder attention to achieve local and global dependencies respectively.

Figure 2.6 represents the Transformer. The following details its key mechanisms.

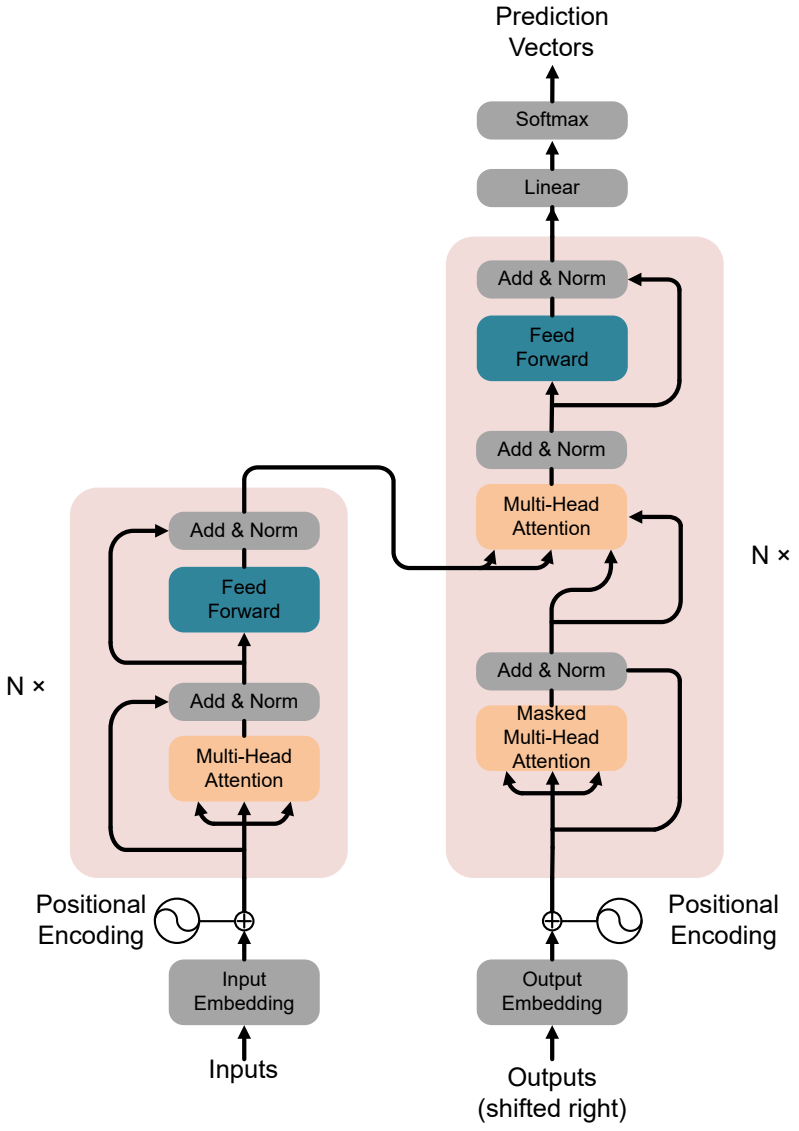


FIGURE 2.6: The Transformer model [5]

Encoder-decoder The Transformer consists of an encoder and a decoder. The encoder maps the input sequence $(x_1, \ldots, x_n)$ into continuous hidden states $(z_1, \ldots, z_n)$. The decoder further generates the output sequence $(y_1, \ldots, y_n)$ based on the hidden states of the encoder. The probability model of the Transformer is in the same

form as that of the vanilla sequence-to-sequence model introduced in § 2.7.2.5. [5] stacked 6 identical encoder layers and 6 identical decoder layers. An encoder layer consists of a multi-head attention component and a simple feed-forward network, both of which apply residual structure. The structure of a decoder layer is almost the same as that of an encoder layer, except for an additional encoder-decoder attention layer, which computes the attention between the decoder hidden states of the current time step and the encoder output vectors. The input of the decoder is partially masked to make sure that each prediction is based on the previous tokens, avoiding predicting with the presence of future information. Both inputs of the encoder and decoder use a positional encoding mechanism.

Self-attention For an input sentence $x = (x_1, \ldots, x_n)$, each token x_i corresponds to three vectors: query, key, and value. The self-attention computes the attention weight for every token x_i against all other tokens in x by multiplying the query of x_i with the keys of all the remaining tokens one by one. For parallel computing, the query, key, and value vectors of all tokens are combined into three matrices: Query (Q), Key (K), and Value (V). The self-attention of an input sentence x is computed by the following formula:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.30}$$

Where d_k is the dimension of queries or keys.

Multi-head attention To jointly consider the information from different subspaces of embedding, query, key, and value vectors are mapped into h vectors of identical shapes by using different linear transformations, where h denotes the number of heads. Attention is computed on each of these vectors in parallel, and the results are concatenated and further projected. The multi-head attention can be described as:

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \tag{2.31}$$

Where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$ and W denotes the linear transformations.

Positional encoding The proposed Transformer architecture has no recurrent units, which means that the order information of the sequence is dismissed. The positional encoding is added with input embeddings to provide positional information. Cosine functions were chosen for positional encoding:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \tag{2.32}$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \tag{2.33}$$

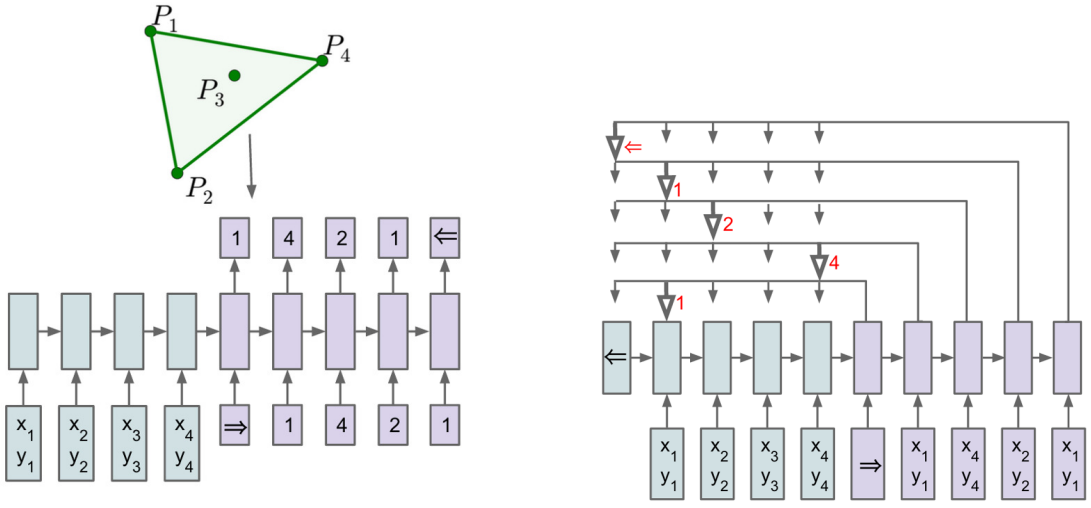
Where *pos* denotes the position of the target token and i denotes the dimension, which means that each dimension of the positional matrix uses a different wavelength for encoding.

Transformer-based pre-train models and Transformer variants Recently, many Transformer-based pre-train models have been developed. Unlike Embeddings from Language Model (ELMo) proposed by [97], which is an LSTM-based contextual embedding model, Transformer-based pre-train models are more powerful. Two representative models are GPT-2 [172] and BERT [173]. GPT-2 and BERT both consist of 12 Transformer blocks and BERT is further improved by making the training bi-directional. They are powerful due to their capability of adapting to new tasks after pre-training. This property helped achieve significant improvements in many NLP tasks. There also evolve many Transformer variants [174–176], which are designed to reduce the model parameters/computational complexity or improve the performance of the original Transformer in diverse scenarios. [177] and [178] systematically summarize the state-of-the-art Transformer variants for academics that are interested.

2.7.5 Pointer Net and CopyNet

2.7.5.1 Pointer Net

In some NLP tasks like dialogue systems and question-answering, the agents sometimes need to directly quote from the user message. Pointer Net [6] (Figure 2.7) solved the problem of directly copying tokens from the input sentence.



(A) Sequence-to-sequence (B) Pointer Net

FIGURE 2.7: **(a)** *Sequence-to-sequence* - The RNN (blue) processes the input sequence to produce a code vector, which is then used by the probability chain rule and another RNN to generate the output sequence (purple). The dimensionality of the problem determines the output dimensionality, which remains constant through training and inference. **(b)** *Pointer Net* - The input sequence is converted to a code (blue) by an encoding RNN, which is fed to the generating network (purple). The generating network generates a vector at each step that modulates a content-based attention process across inputs. The attention mechanism produces a softmax distribution with a dictionary size equal to the input length. [6]

Traditional sequence-to-sequence models [164, 179] with an encoder-decoder structure map a source sentence to a target sentence. Generally, these models first map the source sentence into hidden state vectors with an encoder and then predict the output sequence based on the hidden states. The sequence prediction is accomplished step-by-step, with each step predicting one token using greedy search or beam search. The overall sequence-to-sequence model can be described by the following probability model:

$$P(C^P|P;\theta) = \prod_{i=1}^{m(P)} p(C_i|C_1,...,C_{i-1},P;\theta) \tag{2.34}$$

Where (P, C_p) constitutes a training pair, $P = \{P_1, ..., P_n\}$ denotes the input sequence and $C_p = \{C_1, ..., C_{m(p)}\}$ denotes the ground target sequence. θ is a decoder model.

The sequence-to-sequence models have vanilla backbones and attention-based backbones. Vanilla models predict the target sequence based only on the last hidden state of the encoder and pass it across different decoder time steps. Such a mechanism restricts the information received by the decoder at each decoding stage. Attention-based models consider all hidden states of the encoder at each decoding step and calculate their importance when utilizing them. To compare the mechanism of Pointer Net and Attention, we present the equations explained in Section 2.7.2 here again. The decoder predicts the token conditioned partially on the weighted sum of encoder hidden states d_i:

$$d_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{2.35}$$

Where α_{ij} is the normalized weight score:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{2.36}$$

e_{ij} is the similarity score between s_{i-1} and jth encoder hidden state h_j, where the score is predicted by the similarity model a:

$$e_{ij} = a(s_{i-1}, h_j) \tag{2.37}$$

At each decoding step, both vanilla and attention-based sequence-to-sequence models predict a distribution over a fixed dictionary $X = \{x_1, ..., x_n\}$, where x_i denotes the tokens and n denotes the total count of different tokens in the training corpus. However, when copying words from the input sentence, we do not need such a large dictionary. Instead, n equals to the number of tokens in the input sequence (including repeated ones) and is not fixed since it changes according to the length of the input sequence. Pointer Net made a simple change to the attention-based sequence-to-sequence models: instead of predicting the token distribution based on the weighted sum of encoder hidden states d_i, it directly used the normalized weights α_i as predicted distribution:

$$P(C_i|C_1, ..., C_{i-1}, P) = \alpha_i \tag{2.38}$$

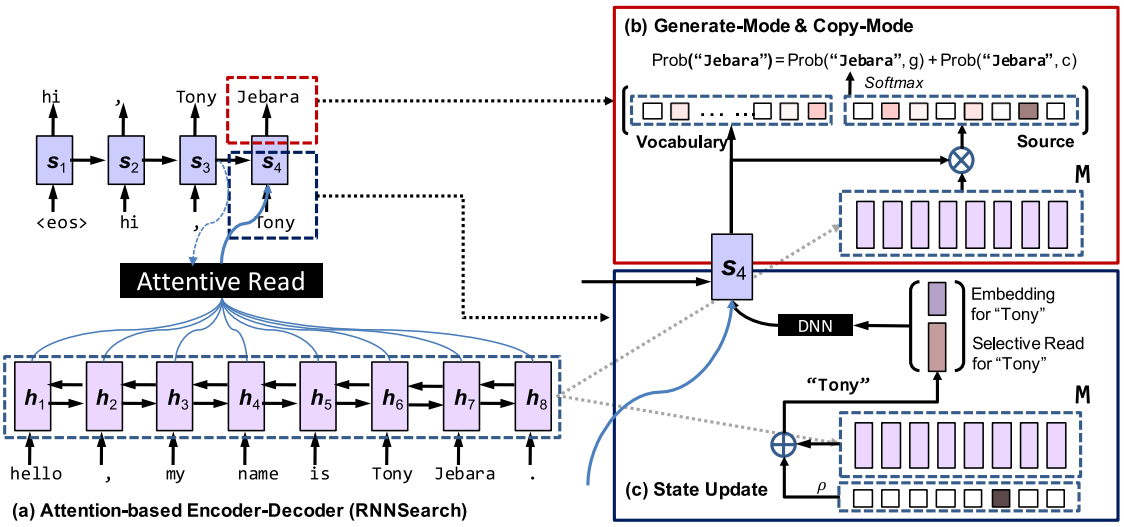


FIGURE 2.8: The overall architecture of CopyNet [7]

Where α_i is a set of probability numbers $\{\alpha_i^1, ..., \alpha_i^j\}$ which represents the probability distribution over the tokens of the input sequence. Obviously, the *token prediction* problem is now transformed into *position prediction* problem, where the model only needs to predict a position in the input sequence. This mechanism is like a pointer that points to its target, hence the name "Pointer Net".

2.7.5.2 CopyNet

In real-world applications, simply copying from the source message is not enough. Instead, in tasks like dialogue systems and QA, agents also require the ability to generate words that are not in the source sentence. CopyNet [7] (Figure 2.8) was proposed to incorporate the copy mechanism into traditional sequence-to-sequence models. The model decides at each decoding stage whether to copy from the source or generate a new token not in the source.

The encoder of CopyNet is the same as that of a traditional sequence-to-sequence model, whereas the decoder has some differences compared with a traditional attention-based decoder. When predicting the token at time step t, it combines the probabilistic models of generate-mode and copy-mode:

$$P(y_t|s_t, y_{t-1}, c_t, M) = P_g(y_t|s_t, y_{t-1}, c_t, M) + P_c(y_t|s_t, y_{t-1}, c_t, M) \tag{2.39}$$

Where t is the time step. s_t is the decoder hidden state and y_t is the predicted token. c_t and M represent the weighted sum of encoder hidden states and encoder hidden states respectively. g and c are generate-mode and copy-mode respectively.

Besides, though it still uses y_{t-1} and weighted attention vector c_t to update the decoder hidden state, y_{t-1} is uniquely encoded with both its embedding and its location-specific hidden state; also, CopyNet combines attentive read and selective read to capture information from the encoder hidden states, where the selective read is the same method used in Pointer Net. Different from the Neural Turing Machines [179, 180], the CopyNet has a location-based mechanism that enables the model to be aware of some specific details in training data in a more subtle way.

2.8 Summary

In this chapter, we discuss the attention optimization works that studied the inherent issues of the attention mechanism. In addition, we review the recent advances in sub-symbolic-based and neural-symbolic-based NLP tasks. More specifically, for sub-symbolic-based tasks, we discuss Machine Translation, Language Modeling, Abstractive Summarization, and Spoken Language Understanding; for neural-symbolic-based tasks, we discuss Dialogue Commonsense Reasoning. Generally, for all NLP tasks discussed, attention-based models have become a better choice. Moreover, pre-trained attention-based models are becoming a trend in the whole NLP community, since they learn massive knowledge on a large corpus and greatly advance the state of the arts.

Moreover, we introduce the main backbone variants for sequence encoding and transduction, including Convolutional Neural Networks, Recurrent Neural Networks, Sequence-to-sequence Models, Memory Networks, Attention Mechanisms, Transformers, Pointer Net, and CopyNet. The model backgrounds and principles are introduced in detail. In addition, the evolution of these backbone models is analyzed to reveal the relationships between them.

Chapter 3

Grouped Head Attention for Language Transduction

3.1 Introduction

Transformer [23] has shown promising performance across various tasks . However, it has some issues, e.g., redundancy and over-parameterization, which is mainly caused by Multi-Head Attention (MHA) [15, 16] and Feed-Forward Network (FFN) [27, 36, 182] of Transformer. We mean to mitigate the redundancy and over-parameterization issues by optimizing the MHA module. The multi-heads were originally designed to attend to different representation subspaces of input [23]. However, prior works [15, 16] have shown that the attention heads are highly redundant and over-parameterized after training because some heads can be switched off with a negligible performance drop. Such an issue is probably caused by their parallel design: the heads mean to work in the same way and likely attend to similar feature subspaces [183].

The prior redundancy optimization methods were mainly based on homogenization, diversification, and head significance. However, they all have some limits. The

* This chapter is published with material from: Jinjie Ni, Rui Mao, Zonglin Yang, Han Lei, Erik Cambria. Finding the Pillars of Strength for Multi-Head Attention. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2023) [181].

homogenization-based method mitigates redundancy and over-parameterization by homogenizing the heads and removing unnecessary parameters. The diversification-based method diversifies the heads to enrich features and reduce inter-head redundancy. Specifically, Cordonnier et al. [70] homogenized attention heads by sharing most weights between all heads, which reduced the redundant information but sacrificed the performance somewhat because of the lack of diversity; Li et al. [26] found that diversifying attention heads by adding a regularization could force MHA to reduce inter-head information redundancy, yielding performance gains to Machine Translation. Whereas, keeping all feature subsets is sub-optimal, because it does not address the issue that MHA is over-parameterized. The significance-based methods [15, 16, 69] learn significance scores for the heads to prune down insignificant ones. However, the remaining important heads still remain inter-head redundancy without diversifying them.

To solve the issues of the above-mentioned methods and inspired by the minimum-redundancy feature selection [184], we hypothesize that attending to the most representative and distinctive feature subsets can achieve more effective and efficient MHAs. Accordingly, we propose a divide-and-conquer strategy, including Group-Constrained Training (GCT) and a Voting-to-Stay (V2S) procedure to mitigate the redundancy and over-parameterization issues. We illustrate the strategy below.

We first propose a strategy to group and distinguish attention heads. Grouped Head Attention (GHA) is obtained via self-supervised Group-Constrained Training (GCT). By encouraging homogenization within a group and diversification between groups, the MHA is forced to divide its heads to work in several separate groups, where each group focuses on an essential but unique feature subset. In GCT, both homogenization and diversification reduce the redundancy of MHA. Note that the essence of redundancy is the model's learning ability being more than enough to process the current information [70]. Accordingly, homogenization reduces parameter redundancy by removing extra parameters to lower the model's learning ability; whereas diversification reduces information redundancy by forcing the model to attend to more diversified information so that the information to process matches the learning ability. The post-grouped attention (GHA) heads take advantage of the following two aspects:

- The inter-group diversification reduces information redundancy, forcing different groups to attend to the subspaces with very different features, resolving

the drawbacks of homogenization- and significance-based methods.

- The intra-group homogenization reduces parameter redundancy by making the intra-group heads become similar. Thus, they can be pruned later for better parameter efficiency, compared to previous diversification-based methods.

Next, we show that GHA-PS (GHA with the Pillar of Strength), a lighter-weight GHA, can be achieved by excluding the redundant parameters of GHA via the V2S procedure. V2S culls the redundant heads that share similar patterns with the most representative head (PS head) of a group. The PS head of a group is selected by voting on different training batches. Note that upon the convergence of the GCT, the heads are highly homogenized within a group, and thus being redundant because they process similar information. As a result, once the redundant heads are culled, the PS heads can still achieve the essential utility of the original attention layer and yield comparable performance to the unculled model. The Lottery Ticket hypothesis [185] argued that there exist subnetworks in an over-parameterized neural network, which can converge faster and achieve comparable or even better performance than the original network. Our GHA-PS achieving greater results is also in line with this hypothesis.

We evaluate our method on three benchmarking tasks. We denote the corresponding Transformer architectures of GHA and GHA-PS as Grouped Head Transformers (GHT) and Grouped Head Transformers with the Pillars of Strength (GHT-PS), respectively. GHT and GHT-PS achieve significant improvements over the strong baselines in Machine Translation BLEU scores (+3.8% and +4.4% averaged on 7 translation tasks), Language Modeling perplexity (-2.8% and -2.9%), and Abstractive Summarization F1-Rouge (+6.7% and +7.0% on average). GHT-PS exhibits higher efficiency in model size, inference speed, and floating-point operations (FLOPs). The light architecture of GHT-PS reduces 63.6% parameters of the vanilla Transformer and yields comparable performance.

Our contribution is summarized as threefold:

- We find that, in a certain range, higher compactness of attention heads (i.e., the intra-group heads become closer to each other and the inter-group ones become farther) improves MHA's performance, which forces the MHA to focus on the most representative and distinctive features. It provides guidance for future architectural designs.

- We propose a divide-and-conquer strategy that consists of GCT and V2S. It mitigates the redundancy and over-parameterization issues of MHA. Our method uses fewer parameters and achieves better performance, outperforming the existing MHA redundancy/parameter reduction methods.
- We verify our methods on three well-established NLP tasks. The superior results on datasets with multiple languages, domains, and data sizes demonstrate the effectiveness of our method.

3.2 Related Work

Parameter efficiency. Different methods were proposed to achieve lightweight Transformers: (a) replacing attention with lightweight modules, e.g., convolution modules, such as Dynamic Conv [36] and Lite Transformer [27]; (b) removing or replacing the feed-forward layers, such as Sukhbaatar et al. [182] and Wu et al. [27]; (c) pruning the model, such as Michel et al. [15], Voita et al. [16], and Li et al. [69].

Modified multi-head mechanism. Ahmed et al. [186] learned to weight the projected output of different heads, performing weighted sum over them. Li et al. [187] aggregated the output of different heads by dynamic routing; Cui et al. [188] used different attention mechanisms, e.g., global/local and forward/backward attention for different heads; Shazeer et al. [67] mixed different heads before and after the softmax operation in an attention function to achieve communication between heads.

Head redundancy optimization. Michel et al. [15] and Voita et al. [16] found that only a subset of the attention heads have significant utilities in Transformer, where the important heads could be identified by Expected Sensitivity and Layer-wise Relevance Propagation (LRP) [68]. Upon this, Li et al. [69] learned per-head importance scores and pruned the heads. Cordonnier et al. [70] homogenized the attention heads by sharing a part of the weights between heads, which lowered the number of parameters but sacrificed performance. Li et al. [26] found that

diversifying attention heads by adding a regularization can force MHA to reduce inter-head redundancy, yielding performance gains for Machine Translation.

As argued before, previous methods either traded performance for efficiency or retained extra redundancy/parameters for effectiveness.

3.3 Methodology

There are two core components in our method, namely the Group-Constrained Training (GCT) and the Voting-to-Stay (V2S) procedure. GHA is developed with GCT that removes head redundancy; GHA-PS is developed by removing the redundant parameters of GHA in V2S. In this section, we detail the process of developing the GHA and finding its Pillars of Strength (PS).

3.3.1 Grouped Head Attention with Hidden Units

First, we detail the core module of GHT, the GHA with hidden units, where heads in a layer are grouped via GCT. GCT enhances the group patterns of attention, performing homogenization and diversification on MHA. Heads within a group will be more similar, whereas heads between groups will be more different. Thus, MHA is forced to divide its heads to work in several separate groups, each group focusing on an essential but unique feature subset, where the head redundancy is reduced. We will demonstrate the effectiveness in § 3.5.

Given a Transformer model $f(\mathbf{x};\theta)$ with n attention layers, the set of heads at attention layer l is denoted as $\mathbf{H}_l = \{\mathbf{h}_{1,l}, ..., \mathbf{h}_{k,l}\}$, where k is the number of heads. The outputs of attention heads are concatenated and projected with W^{out}, where the i-th head output $\mathbf{o}_{i,l}$ in layer l results from the computation of the projection matrices $W^{\mathbf{Q}}_{i,l}$, $W^{\mathbf{K}}_{i,l}$, and $W^{\mathbf{V}}_{i,l}$ of this head:

$$MHA_l(\mathbf{Q},\mathbf{K},\mathbf{V}) = Concate(\mathbf{o}_{1,l}, ..., \mathbf{o}_{k,l})W^{out} \tag{3.1}$$

$$\mathbf{o}_{i,l} = softmax(\frac{(\mathbf{Q}W^{\mathbf{Q}}_{i,l})(\mathbf{K}W^{\mathbf{K}}_{i,l})^T}{\sqrt{d_k}})(\mathbf{V}W^{\mathbf{V}}_{i,l}) \tag{3.2}$$

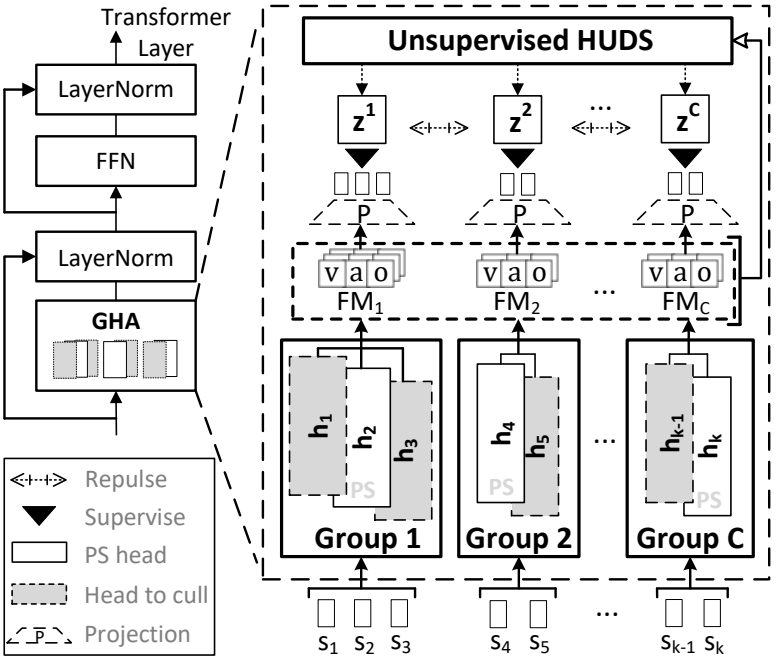


FIGURE 3.1: The Grouped Head Attention. The heads in a group are under self-supervision of the discovered group hidden units (Eq.3.4). The non-PS heads (gray dashed boxes in a group) will be culled in the VS procedure (Algorithm 1). S_k denotes the k-th representation subspace; FM_C denotes the C-th feature map group.

Three feature maps (FMs) of GHA are extracted for the self-supervised GCT: the output of $\mathbf{V}W_l^{\mathbf{V}}$, denoted as $\hat{\mathbf{V}}_l = \{\mathbf{v}_{1,l}, ..., \mathbf{v}_{k,l}\}$ (the value FM); the attention weights of the l-th layer, denoted as $\mathbf{A}_l = \{\mathbf{a}_{1,l}, ..., \mathbf{a}_{k,l}\}$ (the attention FM); the output of the l-th layer before the output projection W^{out}, denoted as $\mathbf{O}_l = \{\mathbf{o}_{1,l}, ..., \mathbf{o}_{k,l}\}$ (the head output FM). $\hat{\mathbf{V}} = \{\hat{\mathbf{V}}_1, ..., \hat{\mathbf{V}}_l\}$, $\mathbf{A} = \{\mathbf{A}_1, ..., \mathbf{A}_l\}$, $\mathbf{O} = \{\mathbf{O}_1, ..., \mathbf{O}_l\}$. Given the FMs, a Hidden Unit Discovery System (HUDS) Ω assigns a hidden unit $\mathbf{z}_{i,l}^j$ for each head (i denotes the i-th head; j denotes the j-th group hidden unit, $\mathbf{z}_{i,l}^j \in \{\mathbf{z}_l^1, ..., \mathbf{z}_l^C\}$) to represent its group property. The discovered hidden units are denoted as $\mathbf{Z}_l = \{\mathbf{z}_{i,l}, ..., \mathbf{z}_{i,l}\}$. $\mathbf{Z}_l = \Omega(\mathbf{E}_l)$, where $\mathbf{E}_l$ denotes either one of the $\hat{\mathbf{V}}_l$, $\mathbf{A}_l$, or $\mathbf{O}_l$. $\Omega(\cdot)$ is an unsupervised algorithm that divides the heads into C groups given their FMs, such as K-means:

$$\Omega(\mathbf{E}_l) = \arg\min_{\mathbf{Z}_l} \sum_{i=1}^{C} \sum_{\mathbf{x} \in \hat{\mathbf{E}}_l^i} ||\mathbf{x} - \mu_i||^2 \tag{3.3}$$

Where $\hat{\mathbf{E}}_l^i$ is the set of feature maps of the i-th head group in the l-th attention layer, and thus the feature map groups are denoted as $\hat{\mathbf{E}}_l = \{\hat{\mathbf{E}}_l^1, ..., \hat{\mathbf{E}}_l^i, ..., \hat{\mathbf{E}}_l^C\}$. μ_i

is the mean of feature map vectors in $\hat{\mathbf{E}}_l^i$.

The hidden units are C-class categorical variables (Eq.3.4(A)) or continuous vectors (Eq.3.4(B)) to supervise the GCT. The objective of the self-supervised GCT is termed as:

$$L_z(\mathbf{f}; \mathbf{A}, \hat{\mathbf{V}}, \mathbf{O}, \mathbf{Z}) =$$
$$\begin{cases} -\frac{1}{kn}\alpha \sum_{l=1}^{n}\sum_{i=1}^{k} log\ p_z(\mathbf{z}_{i,l}|\mathbf{v}_{i,l}, \mathbf{a}_{i,l}, \mathbf{o}_{i,l}) + \frac{1}{(C-1)kn}\beta \sum_{l=1}^{n}\sum_{i=1}^{k}\sum_{j_2 \neq j_1} log\ p_z(\mathbf{z}_l^{j_2}|\mathbf{v}_{i,l}^{j_1}, \mathbf{a}_{i,l}^{j_1}, \mathbf{o}_{i,l}^{j_1}) & \text{(A)} \\ \frac{1}{kn}\alpha \sum_{l=1}^{n}\sum_{i=1}^{k} \varphi(\mathbf{v}_{i,l}, \mathbf{a}_{i,l}, \mathbf{o}_{i,l}; \mathbf{z}_{i,l}) - \frac{1}{\binom{C}{2}n}\beta \sum_{l=1}^{n}\sum_{j_1=1}^{C}\sum_{j_2=j_1+1}^{C} \varphi(\mathbf{z}_l^{j_1}; \mathbf{z}_l^{j_2}) & \text{(B)} \end{cases} \tag{3.4}$$

Either when $\mathbf{z}$ is categorical variables (Eq.3.4(A)) or continuous vectors (Eq.3.4(B)), the objective is composed of a homogenization term and a diversification term. $\mathbf{v}_{i,l}^j$, $\mathbf{a}_{i,l}^j$, and $\mathbf{o}_{i,l}^j$ denote the feature maps of the i-th head belonging to the j-th group. $p_z(\mathbf{z}_{i,l}|\mathbf{v}_{i,l}, \mathbf{a}_{i,l}, \mathbf{o}_{i,l})$ denotes the probability of predictions on the hidden variable $\mathbf{z}_{i,l}$ given $\mathbf{v}_{i,l}$, $\mathbf{a}_{i,l}$, and $\mathbf{o}_{i,l}$. $\varphi(\mathbf{x}; \mathbf{y})$ denotes a cosine similarity measurement between $\mathbf{x}$ and $\mathbf{y}$ (following Li et al. [26]). $\varphi(\mathbf{v}_{i,l}, \mathbf{a}_{i,l}, \mathbf{o}_{i,l}; \mathbf{z}_{i,l}) = \tau_1\varphi(\mathbf{v}_{i,l};\ \mathbf{z}_{i,l}) + \tau_2\varphi(\mathbf{a}_{i,l};\ \mathbf{z}_{i,l}) + \tau_3\varphi(\mathbf{o}_{i,l};\ \mathbf{z}_{i,l})$, where τ is a coefficient, determined by the specific settings for each dataset & task. When $\mathbf{z}$ is a categorical variable, the grouping is a classification task whose classification heads project the output into C classes. When $\mathbf{z}$ is a continuous vector, the grouping process is a metric learning task whose similarity computations are conducted between $\mathbf{z}$ and the projected FM representations. In both conditions, GHA is supervised by $\mathbf{z}$ to make the heads in the same group yield similar patterns, whereas those in different groups repulse from each other. The overall objective is given by $L = L_t + L_z$, where L_t is a task-specific objective. The coefficients α and β of Eq.3.4 respectively control the intra-group homology and inter-group diversity degrees to achieve different group intensities in different tasks/datasets.

3.3.2 The Pillars of Strength

Given the Lottery Ticket hypothesis [185], we establish the GHT-PS from GHT as its subnetwork by removing redundant parameters from GHA, which is the core module of GHT. We propose the V2S procedure that finds out the Pillars of Strength (PS) heads that constitute the core of the GHA and removes other

Algorithm 1 The Voting-to-Stay (V2S) algorithm

Procedure Voting-to-Stay($\mathbf{f}, \hat{\mathbf{V}}, \mathbf{A}, \mathbf{O}, \mathbf{Z}$)
if satisfy ρ, and $\mathbf{m}$ is none **then**
    Start voting epoch; Freeze $\mathbf{f}$.
    $\mathbf{\Gamma}_l \leftarrow [\ ]$ ▷ Creat $\mathbf{\Gamma}_l$ to store votes
    for batch b in B training batches **do**
        for layer l in L layers **do**
            for $\mathbf{E}_l$ in $\{\hat{\mathbf{V}}_l, \mathbf{A}_l, \mathbf{O}_l\}$ **do**
                Create $\mathbf{m}^b_{l,v}, \mathbf{m}^b_{l,a}, \mathbf{m}^b_{l,o}$.
            Add $\mathbf{m}^b_{l,v}$, $\mathbf{m}^b_{l,a}$, $\mathbf{m}^b_{l,o}$ to $\mathbf{\Gamma}_l$.
    for l in n **do** ▷ Vote at each attn layer
        $\mathbf{m}_l \leftarrow VOTE(\{\ \mathbf{m}_{l,v},\ \mathbf{m}_{l,a},\ \mathbf{m}_{l,o}\})$
    $\mathbf{m} \leftarrow [\mathbf{m}_1, ..., \mathbf{m}_n]$ ▷ Stack layer votes
    Unfreeze $\mathbf{f}$; end voting epoch.
$\mathbf{f} = \mathbf{f} \odot \mathbf{m}$ ▷ Mask GHT attn outputs with $\mathbf{m}$

heads. In GHA, the heads within each group exhibit similar patterns after the Group-Constrained Training. Thus, we keep the heads showing the most explicit group patterns (the PS heads) via V2S, and switch off the other ones within the same group. Following Michel et al. [15], Voita et al. [16], we mask out the output of heads as the equivalent operation of head removal[2].

The V2S procedure is performed on fully converged GHAs. The main idea is to vote on all heads of the GHA, and only retain one head for each group - the head receiving the most votes. Specifically, it takes an entire epoch to collect the votes $\mathbf{m}^b_l \in \{0,1\}^k$ from the whole training set (a batch of data (b) creates k votes per attention layer, k being the head number), where 0 indicates that the corresponding head should be switched off and 1 indicates that a head is retained. We assume that there are B mini-batches in the training set, then each head of an attention layer receives B votes, with each vote denoted by either 0 or 1. For each group, the head receiving top-k count for '1's are assigned a '1' in the final head mask $\mathbf{m}_l \in \{0,1\}^k$ for attention layer l, meaning that this head will be retained.

The V2S procedure is outlined in Algorithm 1. We detail some of its definitions below. ρ indicates the full convergence of GHT, i.e., the hidden units found by Ω have a center shift less than a threshold. In Step 7-8, given feature maps $\hat{\mathbf{V}}_l$, $\mathbf{A}_l$, and $\mathbf{O}_l$ of the l-th attention layer, the vote vectors $\mathbf{m}^b_{l,v}$, $\mathbf{m}^b_{l,a}$, and $\mathbf{m}^b_{l,o} \in$

[2]We perform real head removal when test inference speed.

$\{0,1\}^k$ are determined by the group pattern scores (η) of each head, indicating the explicitness of group patterns, where $\eta = p_z(\mathbf{z}_{i,l}|\mathbf{e}_{i,l})$ if $\mathbf{z}$ is categorical; otherwise $\eta = -\varphi(\mathbf{e}_{i,l}; \mathbf{z}_{i,l})$. $\mathbf{e}_{i,l}$ denotes the i-th head feature map (either one of the $\mathbf{v}_{i,l}$, $\mathbf{a}_{i,l}$, or $\mathbf{o}_{i,l}$). We set the corresponding digit in the vote vectors as 1 for the head achieving the highest η in its group, indicating the most representative head of the group. $VOTE$ means counting the '1's for each head based on the *0-1* votes in $\mathbf{\Gamma}_l$ and only keeping the heads with the highest counts[3].

GHT-PS compresses considerable parameters. In the case of two head groups, GHT-PS reduces 75% parameters for an attention layer and 32.1% for the entire model. We will show that V2S removing non-PS heads does not sacrifice model performance. Instead, it brings accuracy gains in some cases and improves inference speed.

3.4 Experimental Setup

In this section, we detail the key architecture configurations of our experiments. Further model, dataset, and evaluation setups are detailed in the Appendices.

3.4.1 Architecture Setup

We follow the Transformer architecture of Vaswani et al. [23] as a backbone architecture for all datasets and tasks in our experiments. For Machine Translation and Abstractive Summarization, we adopt the same 8-head encoder-decoder architecture with 6 layers for both encoder and decoder; the model dimension d_{model} = 512 and feed-forward dimension d_f = 2048. For Language modeling, we adopt the 16-head decoder-only architecture with 16 layers; the model dimension d_{model} = 1024 and feed-forward dimension d_f = 4096. The layer normalization is applied before the residual connection of each layer. The parameters of decoder input and output projections are shared. Our models are based on fairseq[4] Transformer implementations.

[3]Besides voting, there is an alternative way to create the mask. Instead of using *0-1* number as a discrete voting unit, the group pattern scores can be added up to rank the head pattern explicitness. We find that the two ways perform similarly.

[4]`https://github.com/facebookresearch/fairseq`

TABLE 3.1: Benchmark with vanilla Transformer (backbone) on IWSLT and WMT Machine Translation datasets, measured by BLEU. All improvements are statistically significant with $p < 0.05$ under t-test.

Model	Param ↓	BLEU ↑	*IWSLT*				*WMT*	
		de-en	it-en	en-de	en-it	en-fr	en-de	en-fr
Vanilla Transformer [23]	44M	34.4	32.3	28.0	30.8	40.1	27.3	38.1
GHT (ours)	44M	**35.4**	**32.8**	**29.1**	**31.6**	**41.5**	**28.6**	**40.7**
Transformer-Lite1	30M	33.8	31.9	27.9	29.3	39.9	26.9	37.7
Transformer-Lite2	30M	34.0	32.2	28.2	29.5	40.0	26.7	37.8
GHT-PS (ours)	30M	**35.2**	**32.7**	**28.9**	**31.6**	**41.4**	**28.2**	**40.5**

TABLE 3.2: Benchmark with state-of-the-art MHA redundancy/parameter optimization baselines on IWSLT and WMT Machine Translation datasets at the same parameter level, measured by BLEU. * denotes the improvement is statistical significant with $p < 0.05$ under t-test.

Model	Param ↓	BLEU ↑	*IWSLT*				*WMT*	
		de-en	it-en	en-de	en-it	en-fr	en-de	en-fr
Cordonnier et al. [70]	44M	34.4	31.8	28.2	31.0	40.7	27.6	38.5
Li et al. [26]	44M	34.7	31.8	28.5	30.7	40.7	27.3	39.3
GHT (ours)	44M	**35.4***	**32.8***	**29.1***	**31.6***	**41.5***	**28.6***	**40.7***
Voita et al. [16]	30M	32.2	30.8	26.5	30.3	39.8	22.0	34.0
Li et al. [69]	30M	33.2	31.3	27.5	30.0	39.7	20.5	33.6
Dynamic conv [36]	30M	34.8	**32.7**	28.7	31.1	40.6	24.0	36.5
Lite Transformer [27]	30M	33.3	31.4	27.5	29.8	39.4	24.9	37.4
GHT-PS (ours)	30M	**35.2***	**32.7**	**28.9***	**31.6***	**41.4***	**28.2***	**40.5***

We perform the GCT as a metric learning task because it does not introduce additional projection layers when the shapes of similarity inputs are identical (Eq.3.4(B)), which makes GHT weight-lighter. In addition, it performs better in our experiments compared to the classification-based grouping. The training settings are detailed in Appendix A.1. Different α, β, and head feature maps ($\hat{\mathbf{V}}$, $\mathbf{A}$, and $\mathbf{O}$) are preferred for different datasets to achieve optimal performance, we detail the configurations in Appendix A.2.

3.5 Results and Analysis

3.5.1 Machine Translation

Ours vs. vanilla Transformer. We first report results by comparing GHT and GHT-PS with the vanilla Transformer [23] which is the backbone of our model. As shown in Table 3.1, the models are compared at the same parameter levels[5]. GHT does not have weight reduction, keeping the same parameter size as the vanilla Transformer (44M, the same setting as Transformer base [23]). In contrast, GHT-PS is compressed to 30M parameters via V2S. For a fair comparison, we first compare GHT-PS with two lite architectures, Transformer-Lite1 and Transformer-Lite2, whose parameter numbers are 30M as well. Keeping other settings unchanged, the encoder and decoder of Transformer-Lite1 are reduced to 4 layers, respectively. Transformer-Lite2 reduces the model dimension d_{model} to 424, and d_f to 1696.

GHT and GHT-PS consistently and significantly outperform their backbone models at the same parameter level across all datasets. On average, the GHT surpasses 44M vanilla Transformer by 3.8% in BLEU [29]; GHT-PS surpasses Lite1 and Lite2 by 4.9% and 4.4%, respectively. Although GHT-PS reduces 32.1% parameters, it significantly outperforms both 44M and 30M vanilla Transformers, which is comparable to GHT on all datasets. It shows that V2S can reduce the parameter size of the original Transformer without sacrificing accuracy on MT.

Ours vs. lite Transformers. We compare GHT with two state-of-the-art (SOTA) Machine Translation baselines that optimize MHA redundancy, and compare GHT-PS with four SOTA baselines that made major contributions to attention parameter compression and redundancy optimization[6]. Cordonnier et al. [70] and Li et al. [26] are homogenization- and diversification-based methods respectively for MHA redundancy optimization. Voita et al. [16] and Li et al. [69] are pruning-based methods. Dynamic Conv [36] and Lite Transformer [27] modify Transformer architectures to reduce parameters.

[5]The parameters analyzed in this thesis exclude the embedding layer since they vary a lot between different datasets when the vocabulary sizes are different.

[6]We do not compare to the post-pruning of Michel et al. [15], because their method performs extremely bad when the parameter level is low, e.g., 30M [69].

TABLE 3.3: Ablation study on IWSLT'14. The results are generated with beam width 5. All improvements are statistically significant with $p < 0.05$ under t-test.

Model	BLEU ↑				
	de-en	it-en	en-de	en-it	en-fr
GHT	**35.4**	**32.8**	**29.1**	**31.6**	**41.5**
- w/o Diversifying	34.7	31.8	28.5	30.7	40.7
- w/o Homologizing	34.3	32.0	28.2	30.9	40.2
GHT-PS	**35.2**	**32.7**	**28.9**	**31.6**	**41.4**
- w/o GCT	33.8	31.9	28.1	30.5	39.8
- w/o GC	34.0	32.0	28.4	31.0	40.2
- w/o HUDS	33.7	32.0	28.1	30.9	40.3
- w/o PS stay	33.6	31.7	27.9	30.7	40.2
- w/ stage 2 GC	33.2	31.8	28.1	30.8	40.3
- w/ stage 1& 2 GC	33.4	31.9	27.7	30.6	40.2

Table 3.2 shows that GHT outperforms all its baselines on all datasets, exceeding the strongest baseline by 2.9% in averaged BLEU scores. GHT-PS outperforms all its baselines on 6 out of 7 datasets, exceeding the strongest baseline by 4.4% on average. The model compression of the baselines may sacrifice performance (especially on large datasets, e.g., WMT 2014 en-de and en-fr), while GHT-PS is more robust, achieving comparable results to GHT.

Ablation Study. We evaluate the impacts of the variants of GHT and GHT-PS by ablating their features (Table 3.3). We first ablate the diversification/homogenization term of GCT (see Eq.3.4), which lowers the BLEU scores. Next, we show the importance of GCT for V2S. **w/o GCT** denotes that we directly use V2S at the very beginning without GCT. **w/o GC** denotes that V2S is employed after normal training without Group Constrain (GC). Both ablation models yield lower BLEU, because GCT homogenizes unnecessary heads to achieve a sparsified MHA. Next, we validate the power of Pillars of Strength. **w/o HUDS** denotes we replace HUDS with randomly switching off heads after GCT; **w/o PS stay** denotes we keep random group members instead of the Pillars of Strength after GCT. We observe lower BLEU in **w/o HUDS** and **w/o PS stay**. Finally, we find that GC only needs to be added before V2S. We denote the training stages before and after V2S as stages 1 and 2. We compare the proposed Stage 1-based GHT-PS with

TABLE 3.4: Efficiency comparison by parameter sizes, inference speed (averaged on five runs), and FLOPs. All results are generated by beam size 5, batch size 256, and max length 10 on a single NVIDIA Quadro RTX A6000.

Model	BLEU↑	Param↓	Infer Speed↑	FLOPs↓
Transformer base	25.8	44M	1016.4 sent/s	1996M
Transformer big	26.4	176M	707.1 sent/s	6635M
Lite Conv	26.6	166M	722.1 sent/s	6184M
Dynamic Conv	26.9	176M	713.9 sent/s	1913M
GHT-PS-LITE (ours)	26.6	**16M**	**1170.2** sent/s	**1181M**
GHT-PS (ours)	29.4	30M	1122.1 sent/s	1558M

models that perform GCT at Stage 2 (**w/ stage 2 GC**) and at both stages (**w/ stage 1& 2 GC**). BLEU scores of both models decrease.

Efficiency analysis. We analyze the efficiency of GHT-PS by controlling BLEU scores on newstest2013 in Table 3.4, where all the selected baselines achieve similar BLEU[7]. The GHT-PS-LITE is a light version of GHT-PS that has a d_f of 1024. Given BLEU ranging from 25.8 to 26.9, GHT-PS-LITE is more efficient than the baselines, e.g., achieving 90.36% fewer parameters, 62.05% faster inference speed, and 80.90% fewer FLOPs against Lite Conv (the model achieving exactly the same BLEU with GHT-PS-LITE). Besides, GHT-PS achieves 2.5 BLEU increase, 82.95% fewer parameters, 57.14% faster inference speed, and 18.56% fewer FLOPs compared with Dynamic Conv which yields the highest BLEU among the baselines.

Effect of group compactness. The group patterns of MHA become more compact as the GCT goes on (will be shown in a later analysis). We hypothesize that more compact group patterns bring performance gains to Transformers. Figure 3.2 shows the correlation between the group pattern compactness and the BLEU scores achieved on 5 IWSLT'14 datasets when the model is fully converged in GCT. We choose Silhouette Coefficient (SC) [189] and Dunn's Index (DI) [190] as the measurements of group pattern compactness, both of which increase as the intra-group

[7]Not all baseline papers reported close BLEU scores on newstest2013/newstest2014. Tuning a baseline model to achieve a certain BLEU is impractical. Thus, we only select the SOTA models reporting similar BLEU scores.

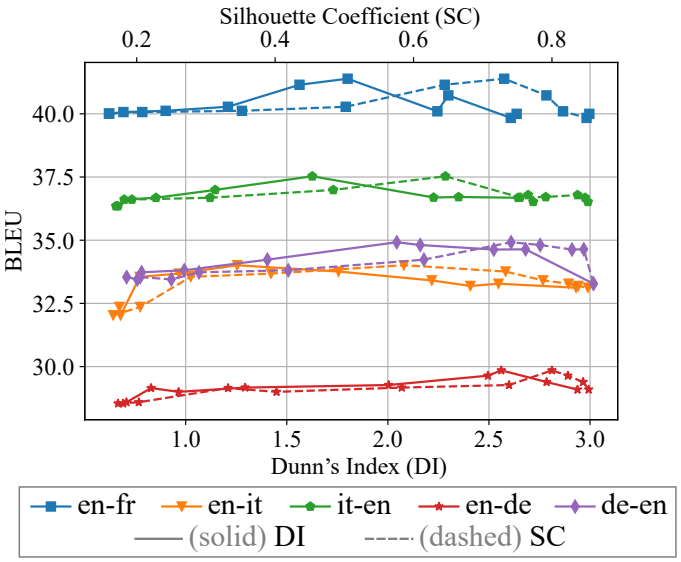


FIGURE 3.2: The BLEU scores of GHT first rise and then drop on IWSLT'14, as the group patterns become more compact (indicated by the increasing SC and DI scores).

samples become more similar and the inter-group ones become more separated. The SC and DI are computed on the FMs in § 3.3.1. Figure 3.2 shows that within the normal range, the BLEU scores rise by higher SC/DI scores, which is in line with our assumption. The BLEUs start to drop as the SC/DI scores increase after the peak, because the very heavy group constraint prohibits the model from learning useful task-specific knowledge.

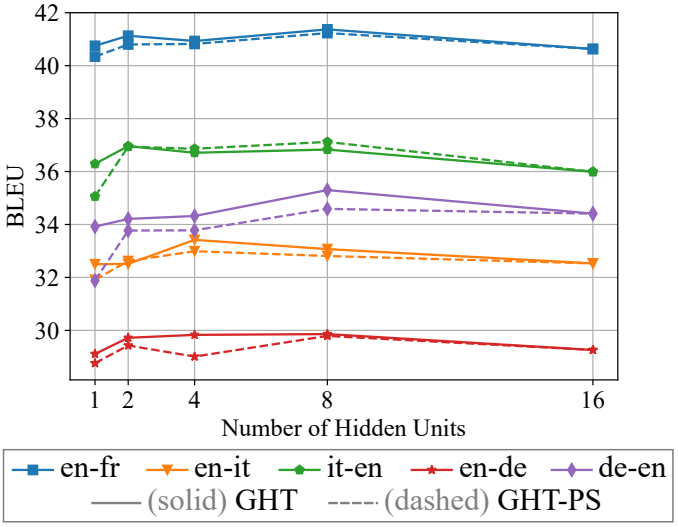


FIGURE 3.3: The BLEUs of GHT and GHT-PS by different numbers of hidden units (groups) on IWSLT'14.

Effect of group number. Figure 3.3 shows the performance trends of 16-head GHT and GHT-PS by different numbers of group hidden units. For GHT, different datasets have different optimal hidden unit quantities, while a similar trend is observed. The optimal group number is between 2 and 8, which is in line with the claim that our group strategy is superior to solely homogenization (1 group) or diversification (16 groups) strategies. For GHT-PS, when the group number is larger than 1, it shows comparable performance to GHT on most datasets. This also verifies that non-PS heads can be switched off without sacrificing performance.

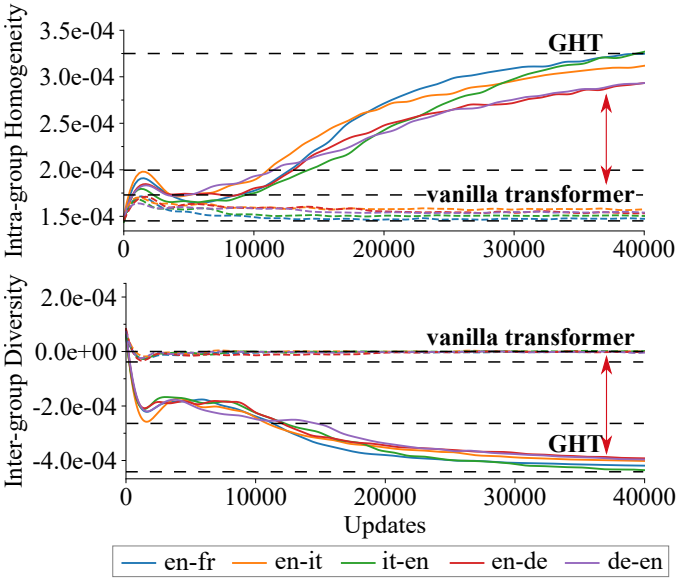


FIGURE 3.4: Intra-group homogeneity (upper) and inter-group diversity (lower) of GHT and vanilla Transformer by training steps.

Group pattern trends. Figure 3.4 exhibits the trends of intra-group homogeneity (the *1st* term of Eq.3.4(B)) and inter-group diversity (the *2nd* term of Eq.3.4(B)) degree of GHT and vanilla Transformer in the training process on five IWSLT datasets. By training, GHT shows higher intra-group homogeneity and lower inter-group diversity, leading to more compact groups, while the vanilla Transformer shows inversed and flattened trends. It proves that GCT can effectively homogenize intra-group heads and diversify inter-group heads for MHA.

3.5.2 Abstractive Summarization

We evaluate the ability of our model to process longer inputs via AS on the CNN-DailyMail dataset. We take vanilla Transformer as the backbone. Table 3.5 shows

TABLE 3.5: Abstractive Summarization results on CNN-DailyMail in terms of F1-Rouge and efficiency (parameter, inference speed, and FLOPs). All improvements are statistically significant with $p < 0.05$ under t-test.

Model	Param ↓	R-1 ↑	R-2 ↑	R-L ↑	Infer Spd ↑	FLOPs ↓
LSTM [93]	-	38.30	14.81	35.49	-	-
CNN [109]	-	39.06	15.38	35.77	-	-
Light Conv [36]	86M	39.52	15.97	36.51	-	-
Dynamic Conv [36]	87M	39.84	16.25	36.73	-	-
Vanilla Transformer [16]	44M	38.45	17.97	36.03	208.77 sent/s	1996M
GHT (ours)	44M	40.00	21.10	37.51	208.77 sent/s	1996M
GHT-PS (ours)	**30M**	**40.01**	**21.31**	**37.62**	**257.62 sent/s**	**1558M**

that both GHT and GHT-PS achieve higher F1-Rouge scores [38] on this task. GHT-PS achieves 4.1% higher Rouge-1, 18.6% higher Rouge-2, and 4.4% higher Rouge-L against vanilla Transformer; 0.4% higher Rouge-1, 31.1% higher Rouge-2 and 2.4% higher Rouge-L against the best performing baseline (Dynamic Conv). Meanwhile, GHT-PS only takes 68.18% parameters of the vanilla Transformer and exhibits higher inference speed and fewer FLOPs.

3.5.3 Language Modeling

TABLE 3.6: Language modeling results on WIKITEXT-103 by perplexity and efficiency (parameter, inference speed, and FLOPs). VT w/ AI denotes vanilla Transformer with adaptive input. All improvements against the baselines are statistically significant with $p < 0.05$ under t-test.

Model	Param↓	Valid↓	Test↓	Infer Spd↑	FLOPs↓
S4	249M	19.69	20.95	-	-
BERT-L-CAS	395M	19.67	20.42	-	-
GPT-2 Large	762M	-	22.05	-	-
VT w/ AI	201M	19.03	19.14	9.9 tok/s	6106M
GHT (ours)	201M	**18.57**	18.60	9.9 tok/s	6106M
GHT-PS (ours)	**167M**	18.58	**18.59**	**19.0 tok/s**	**4573M**

We evaluated our models by an LM task, based on WIKITTEXT-103 dataset. The backbone is a decoder-only Transformer with 16 layers and adaptive inputs [46].

We compare with the backbone model, as well as comparable SOTA LM models, including S4 [191], BERT-Large-CAS [192], and GPT-2 Large [172]. Table 3.6 shows that both GHT and GHT-PS achieve lower perplexity [193] than the baselines on both validation and test sets (2.9% and 9.0% less perplexity against the backbone and the best performing baseline, respectively). Meanwhile, GHT-PS achieves 16.92% model shrinkage, 2× inference speed, and 75% FLOPs compared with the backbone.

3.6 Summary

We assume that only focusing on the most essential and different features may mitigate the redundant and over-parameterization issues of MHA. In particular, we propose the Grouped Head Attention (GHA) trained with a self-supervised group constraint that forces MHA to divide its heads to work in several separate groups, where each group focuses on an essential but unique feature subset. We further propose a Voting-to-Stay procedure to remove redundant parameters in GHA and obtain GHA-PS, a lighter-weight attention. The improvements on three tasks and the extensive analysis verify our hypothesis and the effectiveness of our redundancy optimization methods. Our study provides guidance on future MHA design and training to achieve higher performance.

Chapter 4

Significance Prior Refined Attention for Language Understanding

4.1 Introduction

Recent years witness a great success of pre-trained models in various domains, e.g., natural language [80, 85, 86, 195, 196] and speech [197–199]. The pre-trained models learn to encode the input into high-level features via self-supervised learning tasks such as masked language modeling (MLM), next sentence prediction, etc., and bring considerable performance gain to the downstream tasks [80, 195]. Spoken Language Understanding (SLU) is a set of downstream tasks in the speech domain intending to learn models to understand the content of speech input.

The state-of-the-art end-to-end SLU models are mostly achieved by finetuning on the speech pre-trained models [200]. Despite that the pre-trained speech encoder

* This chapter is published with material from: Jinjie Ni, Yukun Ma, Wen Wang, Qian Chen, Dianwen Ng, Han Lei, Trung Hieu Nguyen, Chong Zhang, Bin Ma, Erik Cambria. Adaptive Knowledge Distillation between Text and Speech Pre-trained Models. 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023), pp. 1-5, doi: 10.1109/ICASSP49357.2023.10096950. [194].

is powerful in preserving content-based information, it still falls short in encoding linguistic information due to the absence of symbolic representation. To complement the speech pre-training, a speech encoder could benefit from transferring knowledge from language models that are pre-trained on unlabelled texts. However, the knowledge obtained during the pre-training stage of speech and text has a natural gap caused by the modal bias and the difference in pre-training methods [201]. This motivates us to study the problem of distilling knowledge from text pre-trained models to speech ones to enhance their performance on understanding tasks.

We categorize existing work on leveraging text knowledge for end-to-end SLU into two classes: one-tower and two-tower. The one-tower approaches [201, 202] encode uni-modal inputs via a shared encoder and is flexible in taking either uni-modal or multi-modal inputs at inference stage. The two-tower methods use modal-specific encoders for speech and text learning, and rely on either additional layers/codebooks [201, 203–206] or alignment losses to align the multi-modal embedding space.

This work falls under the two-tower category aligning the text and speech embedding space by designing metric objectives [20, 207], which we call metric-based distillation. Compared with the one-tower methods, metric-based distillation requires only a small amount of data to distill text knowledge [20] with an explicit alignment objective. It does not introduce any additional parameters, which ensures the model's efficiency and separability. However, due to the large semantic and granularity gap between speech and text, it is challenging to design appropriate metrics for metric-based distillation. Prior works [20, 207] use simple metrics (L2 distance and cosine similarity) to perform metric-based distillation from text to speech ignoring the semantic and granularity difference between them, which harms the distillation effectiveness. To this end, we propose the **P**rior-informed **A**daptive knowledge **D**istillation (PAD) that aligns text-speech representations at adaptive granularity, with the alignment process being informed by the prior knowledge reflecting the semantic significance.

We made the following **contributions**:

- We propose to apply the Attention-based Significance Priors (ASP) to ease the semantic knowledge transfer from texts to speech.

- We propose the Anchor-based Adaptive Span Aggregation algorithm (AASA) that narrows the modal granularity gap of alignments.
- To the best of our knowledge, we are the first that evaluate multiple different alignment strategies beyond vanilla global and local alignments to study the feasibility of metric-based speech-text distillations. The results on three spoken language understanding benchmarks verify our assumptions and claims.

4.2 Method

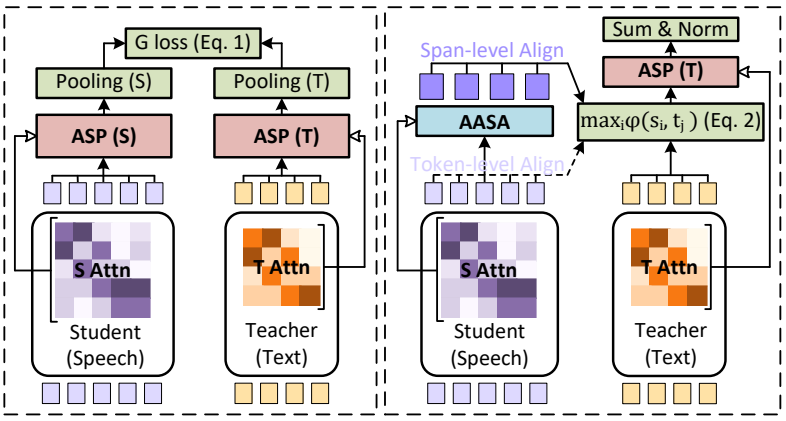
4.2.1 Preliminary: Global and Local Alignment

Metric-based distillation distills text knowledge by aligning the representation space of speech and text pre-trained models. In general, there are two aligning strategies: global and local alignments. Global-level alignment narrows the gap between sequence-level representations and local-level alignment similarizes the local unit representations. Once the alignment training is finished, the speech representation space is closer to that of the text and thus text knowledge is transferred. Given a pair of speech and text data, the output of speech and text pre-trained models are denoted as $\mathbf{S} = [\mathbf{s}_1, ..., \mathbf{s}_n]$ and $\mathbf{T} = [\mathbf{t}_1, ..., \mathbf{t}_n]$ respectively, where $\mathbf{s}_m \in \mathbb{R}^{d_s}$ and $\mathbf{t}_m \in \mathbb{R}^{d_t}$. d_s and d_t are the model dimension of speech and text modules.

Global-level alignment. Sequence-level representations can be extracted from the output representations of the pre-trained models. We denote the global-level output representation of speech and text model as $\hat{\mathbf{s}} \in \mathbb{R}^{d_s}$ and $\hat{\mathbf{t}} \in \mathbb{R}^{d_t}$, which are downsampled embeddings that represent the sentence such as the cls embedding in BERT or the averaged embeddings. The L1 distance of the two sequence representations is computed for alignment:

$$\mathcal{L}_{Glob} = ||\hat{\mathbf{s}} - \hat{\mathbf{t}}||_1 \tag{4.1}$$

During global-level alignment training, the text module is frozen and only the speech module is updated.



(a) Global Alignment (b) Local Alignment

FIGURE 4.1: The global and local PAD. The global and local alignments (Section 4.2.1) are both informed by the **ASP** (Section 4.2.2) to narrow the semantic gap. The **AASA** (Section 4.2.3) adaptively reorganizes the speech sequence to narrow the granularity gap.

Local-level alignment. A finer alignment can be conducted locally. There is a mapping between the speech and text signals which are represented as tokens. The local token units of speech and text module are aligned by maximizing the sum of the maximum text-speech similarities regarding all text units:

$$\mathcal{L}_{Loc} = -\frac{1}{n}\sum_{j=0}^{n} max_i \phi(\mathbf{s}_i, \mathbf{t}_j) \tag{4.2}$$

Where ϕ denotes the similarity metric such as cosine similarity. Same as the global-level alignment, the text module is fixed and only the speech module is updated.

4.2.2 Attention-based Significance Priors

Both global and local alignments introduced in Section 4.2.1 treat tokens in a sequence equally. However, the modal bias of speech and text causes some alignments to be meaningless, e.g., blank and noise signals of acoustic sequences cannot be found in the text, and trying to align these signals with text tokens may not be beneficial to the model's understanding ability. Thus, we assume that the alignment between acoustic features and their text counterpart should be focused on only the most semantic-relevant tokens or frames, for example, the non-blank speech tokens or the keywords in the text sequence. As a result, we need to properly score the significance of tokens (or frames).

[20] used the inverse document frequency (idf) to re-weight the significance of frequent token-frame pairs. However, idf has two problems: (a) it is fixed for each dataset, whereas the significance of a token varies with different contexts; (b) it is not applicable to speech features that are continuous-valued. We propose Attention-based Significance Priors (ASP) that extracts the significance distribution $\mathcal{P}_{sig} \in \mathbb{R}^n$ of a sequence $\mathbf{H} = [\mathbf{h}_1, .., \mathbf{h}_n]$ from the self-attention map of the pre-trained Transformer models[2] (Figure 4.1):

$$\mathcal{P}_{sig}(\mathbf{H}) = \frac{1}{L_0} \sum_{l=0}^{L_0} \frac{1}{\mathbf{e}A(\mathbf{H})^l\mathbf{e}^T} \sum_{m=0}^{n} A(\mathbf{H})^l_m \tag{4.3}$$

The significance score of $\mathbf{h}_m$ is a normalized mean over the attention weights of the units attending to $\mathbf{h}_m$. $A(\mathbf{H})^l = [A(\mathbf{H})^l_1, ..., A(\mathbf{H})^l_n]$ denotes the l-th layer self-attention map; $A(\mathbf{H})^l_m \in \{a \in \mathbb{R} | 0 < a < 1\}^n$ denotes the attention weights of the units that attend to the m-th unit of $\mathbf{H}$, which is the m-th row of $A(\mathbf{H})^l$; L_0 is the number of layers of the Transformer model; $\mathbf{e} = \{1\}^n$. The prior distribution is contextual and we will show that the priors of both speech and text modules narrow the modal semantic gap and contribute to better alignment results. For global alignments, we obtain $\mathbf{S}'$ and $\mathbf{T}'$ after applying the significance prior to the output sequence of speech and text module: $\mathbf{S}' = \mathbf{S} \odot \mathcal{P}_{sig}(\mathbf{S})$, $\mathbf{T}' = \mathbf{T} \odot \mathcal{P}_{sig}(\mathbf{T})$. For local alignments, according to Eq.4.2, we obtain the similarity sequence $\mathbf{\Phi} = [\varphi_1, ..., \varphi_n]$ regarding the text output $\mathbf{T}$, and we obtain $\mathbf{\Phi}'$ constrained by prior: $\mathbf{\Phi}' = \mathbf{\Phi} \odot \mathcal{P}_{sig}(\mathbf{\Phi})$. For PAD, we replace the original sequences with their prior constrained ones in Eq.4.1 and Eq.4.2.

4.2.3 Anchor-based Adaptive Span Aggregation

In reality, a text token may correspond to many speech tokens since a word consists of multiple phonemes. However, the local alignment defined in Section 4.2.1 aligns each text token with the corresponding speech token, which is not consistent in granularity. Such alignment is possible because the attention representations are contextual - each speech embedding potentially represents the contextualized phoneme. However, attention scores are still highly localized for Transformers

[2] We find that the last-layer attention map achieves a better estimation of the ASP in some cases, which led to better performance.

[27, 65, 208], meaning that the embeddings tend to only represent their close neighbors. As a result, there still remains a granularity gap between the representations of speech and text tokens, thus directly aligning them is not optimal.

Algorithm 2 Anchor-based Adaptive Span Aggregation

Procedure AASA $(\mathbf{S}, \xi)$ ▷ Input: $\mathbf{S} \in \mathbb{R}^{n \times d_s}$
Anchor points $\mathbf{\Gamma} \leftarrow [id_{e0}]$; span pools $\tilde{\mathbf{S}} \leftarrow [\ ]$.
$P'_{sig}(\mathbf{S})$, $Index_{sorted} \leftarrow$ Descending Sort $P_{sig}(\mathbf{S})$.
▷ Sample anchor pts by their significance scores (Eq.4.3).
for id in $Index_{sorted}$ **do**
  if $\forall id_e \in \mathbf{\Gamma}, |id - id_e| > \xi/2$ **then** ▷ Control density.
    $span_m \leftarrow \{\mathbf{s}_{id-s_m}, ..., \mathbf{s}_{id+s_m}\}$ for $m \in 1, ..., c$
    ▷ $s_m \in \{\xi/2, \xi, 2\xi, ...\}$; ξ denotes the base scale.
    $\tilde{\mathbf{s}}_{id} \leftarrow \{(pooling(span_1), ..., pooling(span_c)\}$
    Add id to $\mathbf{\Gamma}$; add $\tilde{\mathbf{s}}_{id}$ to $\tilde{\mathbf{S}}$.
Return $\tilde{\mathbf{S}}$ ▷ Output: $\tilde{S} \in \mathbb{R}^{k \times c \times d_s}$

We propose to adaptively generate local spans where each span is aggregated by several speech embeddings to narrow the granularity gap. The alignment process is the same as Eq.4.2, only being different in that the original speech sequence is replaced with the local span pools generated by the Anchor-based Adaptive Span Aggregation (AASA) procedure (Algorithm 2). We denote the local span pools of a speech sequence $\mathbf{S}$ as $\tilde{\mathbf{S}} = [\tilde{\mathbf{S}}_1, ..., \tilde{\mathbf{S}}_k]$, where $\tilde{\mathbf{S}}_m \in \mathbb{R}^{c \times d_s}$. Each span pool $\tilde{\mathbf{S}}_m$ consists of c spans of different scales, and the absolute locations of the span pools are decided by the anchor points. In line with local-level alignment (Eq.4.2), each text token aligns with one of the spans $\tilde{\mathbf{s}}_m$ in $\tilde{\mathbf{S}}$, maximizing the sum of the maximum text-span similarities regarding all text tokens.

4.3 Experimental Setup

We use *wav2vec2-base* and *bert-base-uncased* checkpoints from the Hugging Face[3] as the speech and text modules. We choose 768 dimensions for both models and cut down the BERT vocabulary size to 5000. Following [20], the alignment training uses a 10-hour transcripted data randomly selected from the *train-clean-360* subset of Librispeech 960 [209] dataset. For the evaluation of all downstream tasks, we follow the SUPERB [200] benchmark using S3PRL[4] to achieve fair comparison.

[3] `https://huggingface.co`
[4] `https://github.com/s3prl/s3prl`

TABLE 4.1: Results on IC, ER, and SF in terms of accuracy and F1 score. Our PAD outperforms all the alignment baselines.

Models	Align Variant	Intent Classification	Emotion Recognition	Slot Filling
		Acc ↑	Acc ↑	F1 ↑
wav2vec 2.0 Base [198]	-	94.40	63.02	87.94
Glob - cls [20]	G	97.39	64.19	86.30
TLocal - idf [20]	T	94.91	58.49	82.04
Glob - avr	G	97.30	64.65	86.00
TLocal	T	95.90	59.18	82.27
TLocal - CW	T	96.81	60.27	84.88
TLocal - CG	T	96.76	61.97	84.89
TLocal - OR	T	90.30	59.32	77.36
SLocal - OR	S	97.57	63.87	85.12
PAD-Glob	G	97.68	**64.91**	87.67
PAD-TLocal	T	95.90	62.46	**88.40**
PAD-SLocal	S	**97.86**	62.74	87.19

We evaluate three SLU tasks: Intent Detection (IC), Emotion Recognition (ER), and Slot Filling (SF) on Fluent Speech Commands, IEMOCAP, and Audio SNIPS datasets respectively. SUPERB freezes the upstream pre-trained model and controls all other settings to be identical. [20] finetune the whole upstream model, and they compare with baselines without specifying the settings to be identical. Thus, their alignment effectiveness is not easily comparable and our results based on SUPERB are fairer and more reproducible for future work. We re-implement their alignment methods for comparison.

4.4 Results and Analysis

4.4.1 Results against Various Alignment Methods

We report the results of three levels of PAD against the baselines in Table 4.1. G, T, and S denote global, token-level local, and span-level local alignment respectively. For global-level alignment (Eq.4.1), we compare with 2 variants of sentence embeddings: *cls* embedding (**Glob-cls**) [20] and averaged pooling (**Glob-avr**). For

token-level local alignment, we compare with 5 variants: vanilla token-level alignment (**TLocal**) as in Eq.4.1; token-level alignment using idf to ignore the frequent words (**TLocal-idf**) [20]; first training a speech token recognizer with CTC loss using the same 10-hour data, then the prediction probabilities on text vocabulary are used to weight (**TLocal-CW**, *by constraining the alignment as ASP*) or guide (**TLocal-CG**, *achieve a speech-text mapping by max probability, and align them according to the mapping instead of the max similarity of Eq.4.2*) the alignment; **TLocal-OR** treats the similarity distribution of Eq.4.2 as the probability distribution in CTC calculation and computes the alignment loss as a CTC loss to achieve alignments based on maximum expectations. For span-level local alignment, we compare with the span-level ordered align (**SLocal-OR**), which is the same as the **TLocal-OR** except for its granularity.

The results in Table 4.1 show that our three alignments outperform all baselines in the downstream tasks. Whereas there exist performance trade-offs on different tasks for global and local alignments. The models under global alignments are better at classification tasks (Intent Classification and Emotion Recognition), whereas the ones under local alignments are better at sequence generation tasks (Slot Filling). Classification tasks require better global-level semantics whereas sequence generation tasks require finer token-level semantics. Compared with the token-level alignment, the span-level alignment achieves good performance on both kinds of tasks, which illustrates the effectiveness of AASP that narrows the gap between speech and text granularities.

4.4.2 Ablation Study

We evaluate the impact of the various choices we made for PAD by ablating its features (Table 4.2). For PAD-Glob, we remove the speech and text significance priors one by one and the consistent performance drop verifies their importance. Similarly, we also verify the effectiveness of the significance priors for PAD-TLocal and PAD-SLocal. Additionally, for PAD-SLocal, we find that either (a) *removing the span pools and using fixed spans* or (b) *using even-stride spans instead of anchor-based ones* decreases the performance.

TABLE 4.2: Ablation study on IC, ER, and SF by removing the features one by one.

Align Variants	Align Variant	Intent Classification	Emotion Recognition	Slot Filling
		Acc ↑	Acc ↑	F1 ↑
PAD-Glob	G	**97.68**	**64.91**	**87.67**
- w/o s prior	G	97.55	64.39	87.17
- w/o t prior	G	97.34	64.14	87.59
- w/o both	G	97.30	63.97	86.29
PAD-TLocal	T	**95.90**	**62.46**	**88.40**
- w/o prior	T	**95.90**	59.18	82.27
PAD-SLocal	S	**97.86**	**62.74**	**87.19**
- w/o prior	S	97.52	60.37	83.04
- w/o span pool	S	97.86	62.72	87.06
- w/o anch pts	S	96.73	60.88	86.10

TABLE 4.3: Analysis of the joint alignment combinations.

Align Variants	Align Variant	Intent Classification	Emotion Recognition	Slot Filling
		Acc ↑	Acc ↑	F1 ↑
PAD-Glob	G	97.68	**64.91**	87.67
PAD-TLocal	T	95.90	62.46	**88.40**
PAD-SLocal	S	**97.86**	62.74	87.19
PAD-G&T	G+T	68.63	61.42	77.98
PAD-G&S	G+S	97.86	62.93	87.25
PAD-T&S	T+S	97.68	62.91	86.49
PAD-all joint	G+T+S	97.50	62.40	85.47

4.4.3 Analysis of Joint Alignments

We also evaluate all the joint alignment combinations that perform multiple alignments concurrently (Table 4.3). We find that all of them perform worse than the combined results of separate alignments. Furthermore, we observe that the performances of PAD-G&S and PAD-T&S > PAD-all joint > PAD-G&T, which is in line with our assumptions: the global-level and token-level alignment may be incompatible in essence because of their granularity gap, whereas the granularity of both of them are close to the span-level alignment because of the adaptive aggregation of AASA (Eq.2).

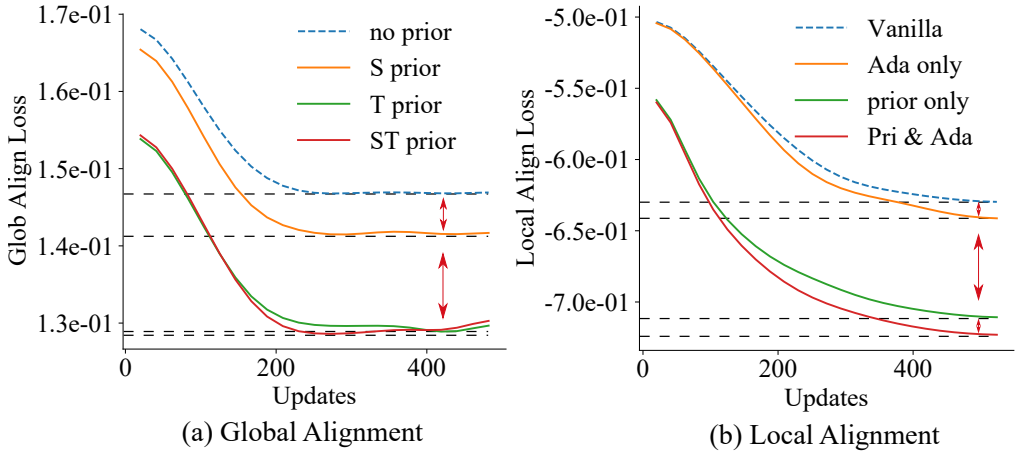


FIGURE 4.2: The global and local-level alignment loss on dev set. S and T denote speech and text respectively.

4.4.4 Analysis of Alignment Loss

Apart from the downstream task accuracies, the alignment loss is also a good metric for comparing the alignment effectiveness since they directly reflect the distance of embedding space. With all entries being properly scaled, we plot the validation loss curves of global and local alignment training from epoch 1 in Figure 4.2. Figure 4.2(a) shows that the three alignments constrained by the prior show lower minimum loss and faster gradient descent compared with the vanilla version, which is in line with the results obtained above, illustrating the effectiveness of the speech and text priors. In Figure 4.2(b), the local alignments that are constrained by the priors or leverage the adaptive spans show lower loss and faster gradient descent, which is also in line with the effectiveness of the significance prior and adaptive span in local-level alignment experiments.

4.5 Summary

We propose the Prior-informed Adaptive knowledge Distillation (PAD) that leverages adaptive spans and the in-stored significance prior distribution to achieve better alignments between well-established text and speech pre-trained models at both global and local levels. The experimental results and analysis illustrate the effectiveness of PAD. We also observe that the trade-off between global and local alignments always exists because they may be contradictory in essence. We will study how to alleviate this trade-off in future work.

Chapter 5

Multi-Hierarchy Attention for Commonsense Reasoning

5.1 Introduction

Building a human-like dialogue system has been a long-lasting goal in the community of conversational AI [10, 210–216]. In the pursuit of this goal, multiple research topics have emerged: context awareness [217], response coherence [218] and diversity [219], speaker consistency [220], empathetic response [221], conversation topic [222], knowledge-grounded system [223], etc. The conversation goal is one of the most representative elements that reflect human intelligence.

Human conversations are usually guided by several small goals or a global goal. As shown in Fig. 5.1, *Grilled Fish*, *Chinese Dish*, *China Town*, and *Cinema* are turn-level goals, while the *Cinema* is also the global goal at the same time. During the conversation, the agent intends to approach the global goal by naturally transitioning between turn-level goals. However, most dialogue systems passively respond to the user without explicit goals, causing incoherent or illogical responses.

* This chapter is published with material from: Jinjie Ni, Vlad Pandelea, Tom Young, Haicang Zhou, Erik Cambria. HiTKG: Towards Goal-Oriented Conversations via Multi-Hierarchy Learning. Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2022), 36(10), 11112-11120. [13]

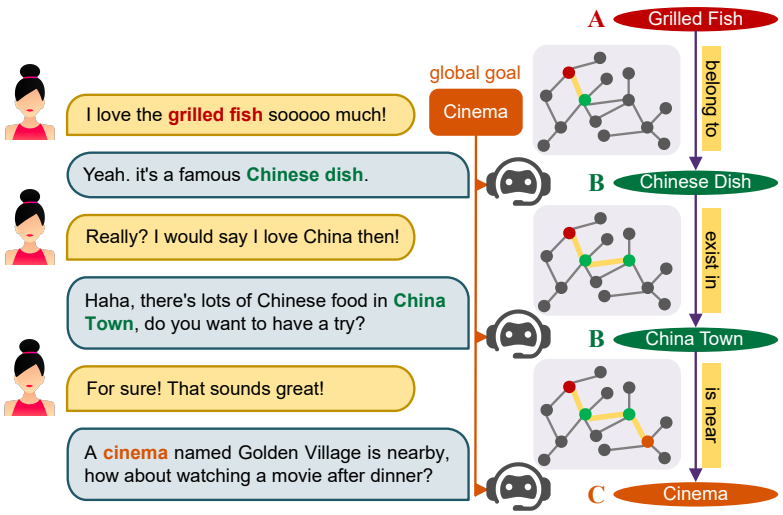


FIGURE 5.1: A goal-driven dialogue sample. Starting from an initial entity (A), the chatbot plans turn-level conversation goals (B) based on dialogue content and history goal trajectory, also trying to naturally direct B to a global goal (C).

In recent years, some researchers attempt to ground dialogue systems on knowledge graphs (KGs) to actively guide conversation topics/goals. KG is a structured knowledge network that consists of vertices, or entities, being connected by edges, or relations [224].

KGs contain commonsense relationships between real-world entities that can also be seen as conversation goals. Generating or retrieving responses according to the walking trajectory in KGs is effective in generating goal-oriented responses. Models reasoning over the KGs based on the dialogue content are called dialogue graph walkers. The current graph walkers can generally be divided into recurrent walkers [22, 43–45] and graph attention based walkers [21]. Recurrent walkers decode KG paths depending on a fixed-length vector, which creates a bottleneck for performance.

Graph attention based KG walkers are good at achieving optimal performance since they reserve all potential paths, but such a mechanism is too high in computation complexity to be scalable to multi-hop reasoning. In addition, these walkers neglect the hierarchical structure of the input source and make separate predictions for entity and relation paths, which affects their performance. Besides, these walkers only plan turn-level goals based on the dialogue history, which means that their reasoning is local and undirected. However, many of the conversations between humans, especially adults, are guided by an ultimate goal.

To this end, we propose **Hi**erarchical **T**ransformer based **K**nowledge **G**raph Walker (**HiTKG**), a graph walker that leverages multiscale inputs to make precise and flexible reasoning on KG paths. We learn this hierarchical model in a two-hierarchy learning framework which employs two stages to learn goal planning. In the first stage, we train HiTKG in a supervised fashion, where it learns how to plan turn-level conversational goal sequences naturally based on the dialogue content; in the second stage, we manually assign a global goal for HiTKG and it learns to approach the global goal via reinforcement learning, where a user simulator is trained to provide user messages for the conversation. In other words, in such two-stage learning, the model learns to approach the target goal without losing the naturalness of the goal sequence. Specifically, HiTKG has a Transformer-based structure, as shown in Fig. 5.2. The graph decoder computes hierarchical attention with different-level memories to obtain a better representation of current states, based on which it reasons over the multi-hop neighbors and decodes a KG path consisting of the related entities and relations. Note that our model is scalable to KG paths of flexible lengths.

To conclude, our contributions are as follows:

- We propose the first Transformer-based KG walker that attentively reads multiscale inputs for graph decoding. We also propose Multi-source Decoding Inputs (**MDI**) and Output-level Length Head (**OLH**) to strengthen the controllability and multi-hop reasoning ability of the Hierarchical Attention based Graph Decoder (**HAGD**).
- We propose a two-hierarchy learning framework to train the proposed hierarchical KG walker, in order to learn both turn-level and global-level conversation goals. This is the first attempt to learn models to make natural transitions towards the global goal in KG, where we propose a distance embedding to incorporate distance information.
- We propose MetaPath (**MP**) to concurrently exploit entity and relation information when reasoning, which is proved essential as the backbone method for KG path representation, providing a paradigm for KG reasoning.

5.2 Related Work

5.2.1 KG-grounded Dialogue Reasoning

Some work augments dialogue inputs with shallow entity and relation information [135, 136, 140, 222, 225, 226]. A knowledge retriever works with the utterance decoder to generate responses based on the retrieved shallow commonsense knowledge entities. These models enjoy rich knowledge augmentation since all short KG paths relating to the user message are encoded, but they lack the ability to extend the topics along with the KG connections, which is essential when organizing a natural dialogue. Another set of work focuses on developing a series of goals/topics for each conversation turn by walking on the KG [21, 22, 44]. These graph walkers are recurrent or graph attention based models that attentively read the dialogues and reason over the KG paths, starting from the entity mentioned in the last user message.

5.2.2 Global Goal Guided Dialogue Reasoning

To the best of our knowledge, we are the first that study global goal oriented KG path transitions in dialogue. Existing similar work either omits explicit approaching [227] or grounds in unstructured knowledge [228]. Xu et al. propose a hierarchical policy model to plan and generate responses of different levels where the high-level policy plans a global topic. However, the low-level policy plans responses that are coherent to this topic instead of approaching it. Tang et al. study guiding the conversation to an assigned target subject, which has a similar purpose as ours, but the transition is grounded in a disordered keyword set that lacks commonsense connections.

5.3 Method

5.3.1 Overview

We define the knowledge graph $\mathbf{G}_{KG} = \mathbf{V}_{KG} \times \mathbf{E}_{KG}$, where the KG is composed of the commonsense vertices (entities) $\mathbf{V}_{KG}$ and edges (relations) $\mathbf{E}_{KG}$ that connect

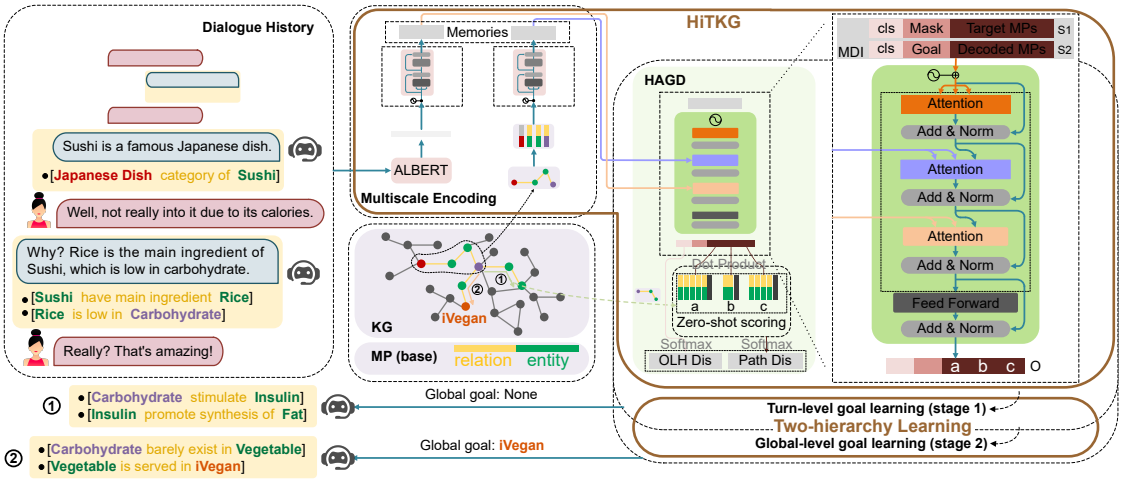


FIGURE 5.2: The overall architecture. HiTKG is composed of multiscale encoders and the Hierarchical Attention based Graph Decoder (HAGD). It first employs two separate Transformers to learn dialogue history and KG path history representations, and then HAGD leverages the multiscale memories to plan KG paths. HiTKG has different reasoning strategies when trained with stage 1 only and with both stages. We optimize the whole HiTKG during training. Note that our task **only** predicts KG paths.

the vertices. We define $\mathbf{V}_{E,n}(v)$ as the set of nodes that are connected to v with n-hop edge connections E.

There are two learning stages for the model to learn different levels of goal planning: turn-level goal learning (stage 1) and global-level goal learning (stage 2). Specifically, at stage 1 (supervised learning), the model takes the multiscale state $\mathbf{x}$ as input, and predicts an n-hop KG path $\mathbf{Y}_p$ which represents the transition of conversation goals. The KG path is made up of the entities and relations in $\mathbf{G}_{KG}$. $\mathbf{x}$ has two scales: $\mathbf{x} = \{\mathbf{x}_p; \mathbf{x}_d\}$, where $\mathbf{x}_p = \{\mathbf{x}_p^{(1)}, \mathbf{x}_p^{(2)}, ..., \mathbf{x}_p^{(i)}\}$ denotes KG path history with a fixed window size i that is generated in previous turns; the $\mathbf{x}_d = \{\mathbf{x}_d^{(1)}, \mathbf{x}_d^{(2)}, ..., \mathbf{x}_d^{(j)}\}$ denotes dialogue history utterances with a fixed window size j that is produced by both speakers. The KG path inference at the t-th dialogue turn of **stage 1** is formulated as:

$$\mathbf{Y}_p^t = \arg\max_{\mathbf{Y}} \prod_{k=1}^{T} P(\mathbf{v}_k^t | \mathbf{x}_p^t, \mathbf{x}_d^t, \mathbf{V}_{E,1}(v_{k-1}^t)) \tag{5.1}$$

Where $\mathbf{Y}_p^t = \{\mathbf{v}_1^t, \mathbf{v}_2^t, ..., \mathbf{v}_T^t\}$ denotes the predicted T-hop KG path. $\mathbf{v}_k^t = [v_{k-1}^t, e_k^t, v_k^t]$ denotes the k-th one-hop KG path of $\mathbf{Y}_p^t$. $v_{k-1}^t \in \mathbf{V}_{KG}$ denotes the starting KG vertex of $\mathbf{v}_k^t$, and $e_k^t \in \mathbf{E}_{KG}$ denotes the edge that connects v_{k-1}^t and v_k^t. At

stage 2 (supervised and reinforcement learning), $\mathbf{x}$ is composed of three scales: $\mathbf{x} = \{\mathbf{x}_p; \mathbf{x}_d; \mathbf{g}\}$, where $\mathbf{g}$ denotes global goal. The model takes $\mathbf{x}$ as input to predict turn-level KG path $\mathbf{v}_{y,k}^t$, and tries to make the entity of $\mathbf{v}_{y,k}^t$ closer to $\mathbf{g}$. Since no n-hop KG path annotation is available at this stage, we predict one-hop paths here to analyze the problem. The KG path inference at **stage 2** is formulated as:

$$\mathbf{v}_{y,k}^t = \arg\max_{\mathbf{v}_k^t} P(\mathbf{v}_k^t | \mathbf{x}_p^t, \mathbf{x}_d^t, \mathbf{g}, \mathbf{V}_{E,1}(v_{k-1}^t)) \tag{5.2}$$

5.3.2 Multiscale Source Representation

The KG path is a series of conversation goals based on which the utterances are organized. We argue that dialogue history is the surface-level representation of a dialogue and KG path history is a higher-level one that can be interpreted as the outline of a conversation. In stage 2, the global goal is the top-level input source that decides the topic flow. As shown in Fig. 5.2, HiTKG encodes multiscale dialogue sources with separate Transformer encoders, where MetaPath is the cornerstone of KG path representation and reasoning. The global goal is represented as a part of the decoder input.

MetaPath (MP) In previous works such as [22] and [21], entities and relations are represented separately in both KG path encoding and decoding. At the decoding stage, KG paths are predicted by scoring entity paths and relation paths respectively, and then rerank. Thus, the model only considers one distribution of the entities or relations at a time, while a KG triple is composed of both. The prediction quality is decided by entity path reasoning, relation path reasoning, and reranking algorithm jointly, which makes it harder to achieve optimum. We propose MetaPath, an effective method to represent and score the KG paths by concurrently considering the entity and relation information. A MetaPath is a flexible combination of embeddings.

Given a KG triple $(\mathbf{v}_1, \mathbf{e}, \mathbf{v}_2)$, a base MP contains the concatenated embeddings of $\mathbf{e}$ and $\mathbf{v}_2$: MP=$[\vec{\mathbf{e}}_e; \vec{\mathbf{e}}_{v_2}]$. Where $\vec{\mathbf{e}}_e, \vec{\mathbf{e}}_{v_2} \in \mathbb{R}^{d_{kg}}$, d_{kg} denoting the KG embedding dimension. Although $\mathbf{v}_1$ is not included, the MP still expresses the full triple, benefiting from the base graph embedding. MP=$[\vec{\mathbf{e}}_e; \vec{\mathbf{e}}_{v_2}]$ instead of $[\vec{\mathbf{e}}_{v_1}; \vec{\mathbf{e}}_e; \vec{\mathbf{e}}_{v_2}]$ because the later form causes information redundancy when encoding KG path

history with multiple MPs (each entity appears twice), causing worse convergence during training. As the cornerstone of KG path reasoning, MP also decides the scoring logic, bringing high efficiency, accuracy, and flexibility.

KG Vertex & Edge Following previous KG walkers, we use TransE [229] to represent the vertices and edges of $\mathbf{G}_{KG}$. TransE is a simple but powerful graph embedding method that is scalable to large KGs and represents multi-scale relationships. The main idea is that given a ground KG triple $(\mathbf{v}_1, \mathbf{e}, \mathbf{v}_2)$, it satisfies $\vec{\mathbf{e}}_{v_1} + \vec{\mathbf{e}}_e \approx \vec{\mathbf{e}}_{v_2}$. The MetaPath and TransE form an ideal combo because according to the principle $\vec{\mathbf{e}}_{v_1} + \vec{\mathbf{e}}_e \approx \vec{\mathbf{e}}_{v_2}$, the embedding of $\mathbf{v}_1$ can be inferred by the model given $\mathbf{e}$ and $\mathbf{v}_2$, which makes up the information loss for a single MP (no loss for MP sequences).

Dialogue History Dialogue history is composed of conversations from past turns and has a fixed window size. The conversation sentences are first encoded with an ALBERT[2] [230] layer ϕ^{al}, which is frozen during training, to obtain contextual representation $\mathbf{e}_d = \phi^{al}(\mathbf{x}_d) = \{\vec{\mathbf{e}}_d^{(1)}, \vec{\mathbf{e}}_d^{(2)}, ..., \vec{\mathbf{e}}_d^{(n_d)}\}$. Memory $\mathbf{h}_d = \{\vec{\mathbf{h}}_d^{(1)}, \vec{\mathbf{h}}_d^{(2)}, ..., \vec{\mathbf{h}}_d^{(n_d)}\}$ is obtained by parallelly applying $\mathbf{h}_d = E_{dial}(\mathbf{e}_d)$. Where E_{dial} is composed of learnable positional embedding [231], multi-head attention α_n [5], and feedforward network (FFN) (we simplify multi-head attention and omit bias terms, and illustrate only one Transformer layer for conciseness):

$$\begin{aligned} E_{dial}(\mathbf{e}_d) &= \psi(\alpha_{self}(\mathbf{e}_d^{pos}) + \mathbf{W}_2\sigma(\mathbf{W}_1\alpha_{self}(\mathbf{e}_d^{pos}))) \\ \alpha_{self}(\mathbf{e}_d^{pos}) &= \alpha_n(\tilde{\mathbf{e}}_d^{pos}|\bar{\mathbf{e}}_d^{pos}) \end{aligned} \tag{5.3}$$

Where $\mathbf{e}_d^{pos}$ denote the position-embedded inputs. $\bar{\mathbf{e}}_d^{pos}$ denotes attention query and $\tilde{\mathbf{e}}_d^{pos}$ denotes attention key/value. ψ denotes layer normalization [232]. $\mathbf{W}_1 \in \mathbb{R}^{d_m \times d_f}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_f \times d_m}$ are the weight matrices of FFN, where d_m and d_f denotes the model and FFN dimension, respectively. We add $[usr1]$ and $[usr2]$ in front of respective messages to identify the utterances from different speakers, which has been proven effective in dialogue tasks [233].

[2] A state-of-the-art contextual representation model. It achieves better performance than BERT-large with fewer parameters.

KG Path History TransE-based MetaPath transformation is denoted as φ^{mp}. We stack all MPs corresponding to their respective one-hop KG paths to represent KG path history $\mathbf{x}_p$. A starting MetaPath MP_{bop}=$[\vec{\mathbf{e}}_{bop}; \vec{\mathbf{e}}_{1st}]$ is placed at the beginning. $\vec{\mathbf{e}}_{bop}$ is a special token embedding indicating the beginning of the path and $\vec{\mathbf{e}}_{1st}$ is the starting entity embedding of $\mathbf{x}_p$. The MP_{bop} ensures that all entities of the original KG path history are presented and indicates the starting point. E_{kg} further aggregates the MP representations of path history $\mathbf{e}_p = \varphi^{mp}(\mathbf{x}_p) = \{\vec{\mathbf{e}}_p^{(1)}, \vec{\mathbf{e}}_p^{(2)}, ..., \vec{\mathbf{e}}_p^{(n_p)}\}$ to get memory $\mathbf{h}_p = \{\vec{\mathbf{h}}_p^{(1)}, \vec{\mathbf{h}}_p^{(2)}, ..., \vec{\mathbf{h}}_p^{(n_p)}\}$ by applying $\mathbf{h}_p = E_{kg}(\mathbf{e}_p)$, where E_{kg} has the same architecture as E_{dial}.

Global Goal The global goal $\mathbf{g}$ is an entity of $\mathbf{G}_{KG}$ that is manually or randomly selected. At the stage of global-level goal learning, the global goal is a significant source of input, while it is not required when learning the turn-level goals. Considering that stage 1 and stage 2 share the same model, they need to keep the same input forms. Thus, we put the global goal in front of the target sequence, jointly as an input of the graph decoder. At stage 1, we use a target mask to mask out the global goal embedding when computing self-attentions for decoder inputs.

5.3.3 Turn-level Goal Learning

We propose **Hierarchical Attention based Graph Decoder (HAGD)** to predict turn-level KG paths and train it in a supervised fashion. Given KG environment $\mathbf{V}_e^{tar}$ of the target sequence, graph decoder ξ decodes KG paths:

$$\mathbf{y}_p = \xi(S_a, \mathbf{h}_d, \mathbf{h}_p || \mathbf{V}_e^{tar}) \tag{5.4}$$

$S_a = [\vec{\mathbf{c}}; \vec{\mathbf{e}}_{gm}; \mathbf{t}]$ denotes turn-level **Multi-source Decoding Inputs (MDI)**, which aggregates static and dynamic states for ξ, being the top level of multi-scale sources. $\vec{\mathbf{c}} \in \mathbb{R}^{d_m}$ is the corresponding *cls* embedding of OLH. $\vec{\mathbf{e}}_{gm} \in \mathbb{R}^{d_m}$ denotes masked global goal embedding (padded). $\mathbf{t} \in \mathbb{R}^{n_s \times d_m}$ is the shifted right target sequence starting with MP_{bop}.

Multi-hierarchy Attention Block ξ has three scales of sources: MDI, dialogue history, and KG path history memories, being denoted as S_a, $\mathbf{h}_d$, and $\mathbf{h}_p$,

respectively. We build a multi-hierarchy attention block to aggregate the multiscale information. Specifically, the proposed HAGD has three attention layers α_{self}, α_{kg}, and α_{dial} that align with MDI, KG path history, and dialogue history, respectively:

$$\begin{aligned} \kappa_s^{top} &= \tau(\alpha_{self}(S_a^{pos})) &&= \tau(\alpha_n(\tilde{S}_a^{pos}|\bar{S}_a^{pos})) \\ \kappa_p^{mid} &= \tau(\alpha_{kg}(\mathbf{h}_p|\kappa_s^{top})) &&= \tau(\alpha_n(\tilde{\mathbf{h}}_p|\bar{\kappa}_s^{top})) \\ \kappa_d^{bot} &= \tau(\alpha_{dial}(\mathbf{h}_d|\kappa_p^{mid})) &&= \tau(\alpha_n(\tilde{\mathbf{h}}_d|\bar{\kappa}_p^{mid})) \end{aligned} \tag{5.5}$$

τ denotes the residual operation $\tau(y(x)) = x + y(x)$. A self-attention layer α_{self} computes attention over the top-level source MDI; the resulting context vectors $\kappa_s^{top} = \{\vec{\kappa}_{s,1}^{top}, \vec{\kappa}_{s,2}^{top}, ..., \vec{\kappa}_{s,n_s}^{top}\}$ interact with $\mathbf{h}_p$ at the middle layer α_{kg}; then taking the resulting residual state $\kappa_p^{mid} = \{\vec{\kappa}_{p,1}^{mid}, \vec{\kappa}_{p,2}^{mid}, ..., \vec{\kappa}_{p,n_s}^{mid}\}$ from α_{kg}, α_{dial} leverages $\mathbf{h}_d$ and obtain $\kappa_d^{bot} = \{\vec{\kappa}_{d,1}^{bot}, \vec{\kappa}_{d,2}^{bot}, ..., \vec{\kappa}_{d,n_s}^{bot}\}$. $\vec{\kappa}_{s,k}^{top}$, $\vec{\kappa}_{p,k}^{mid}$, and $\vec{\kappa}_{d,k}^{bot} \in \mathbb{R}^{d_m}$.

Output-level Length Head (OLH) Humans have a general estimation of how long should the goal trajectory be when they plan the conversation goals, though this could be subconscious. To this end, we place a *cls* token at the beginning of the decoder input and attach the corresponding OLH to the output layer for path length prediction. The output $O = [\vec{\mathbf{c}}_\iota; \vec{\mathbf{g}}_\iota; \mathbf{t}_\iota]$ has the corresponding shape of S. $\vec{\mathbf{c}}_\iota$ is the OLH state representation for KG path length prediction; $\vec{\mathbf{g}}_\iota$ is a placeholder; $\mathbf{t}_\iota$ is the KG path state representation. We compute path length distribution $\mathbf{y}_l$ as follows:

$$\mathbf{y}_l^i = \frac{exp(\mathbf{c}_\iota^i)}{\sum_{k=1}^{N} exp(\mathbf{c}_\iota^k)} \tag{5.6}$$

Where $\mathbf{c}_\iota^k \in \mathbb{R}$. This multi-task learning framework predicts path length and KG paths concurrently and achieves improvement in the performance of KG path prediction.

Zero-shot Scoring Note that different from traditional language models, the total number of candidates M is not fixed at different path positions n. Thus, given the path state representation $\mathbf{t}_\iota$, we compute the zero-shot embedding similarity to score the paths:

$$\mathbf{y}_p^{n,j} = \frac{exp(\vec{\mathbf{t}}_\iota^n \cdot \vec{\pi}_n^j)}{\sum_{k=1}^{M} exp(\vec{\mathbf{t}}_\iota^n \cdot \vec{\pi}_n^k)} \tag{5.7}$$

Where $\Pi_n = \{\vec{\pi}_n^1, \vec{\pi}_n^2, ..., \vec{\pi}_n^m\}$ denotes a set containing neighbor MP candidates of the n-th node in a KG path and the *end of path* embedding MP_{eop}. $\vec{\pi}_n^k \in \mathbb{R}^{d_{mp}}$ denotes the k-th MP candidate of Π_n. We compute Cross Entropy loss of length and KG path prediction to optimize HiTKG:

$$\mathcal{L}_{sup} = \gamma \mathcal{L}_{CE}(\mathbf{y}_l, \hat{\mathbf{y}}_l) + \lambda \mathcal{L}_{CE}(\mathbf{y}_p, \hat{\mathbf{y}}_p) + \epsilon \sum_w w^2 \tag{5.8}$$

Where $\hat{\mathbf{y}}_l$ and $\hat{\mathbf{y}}_p$ denote the ground truth path length and KG path, respectively. γ and λ are weight coefficients. $\epsilon \sum_w w^2$ is the weight decay term.

5.3.4 Global-level Goal Learning

We propose a reinforcement KG walker HiTKG-RL, which has the same architecture as HiTKG, to walk on the $\mathbf{G}_{KG}$ under the guidance of a global goal. This learning stage can be viewed as the combination of a pre-train stage (generally the same as stage 1) and a fine-tuning stage where we apply a reinforcement framework to teach the pre-trained KG walker how to approach the global goal without losing naturalness.

User Simulation We train a user simulator to generate user responses as dialogue history when interacting. The user simulator has the same architecture as the KG walker, which takes the dialogue history and KG path history as input sources. The decoder input sequence $S^{usr} = [\vec{\mathbf{v}}_c; \mathbf{t}]$ is different from that of the KG walker, where $\vec{\mathbf{v}}_c$ denotes the KG vertex of current turn. Instead of predicting the KG paths, the decoder output is modified to predict the probability distribution over a fixed vocabulary set to generate human responses. The dialogue history is solely composed of the simulated responses. The omission of the second speaker's response barely influences the overall performance of KG path reasoning, which is indicated by an ablation experiment.

Distance Embedding To measure how close the current node is to the global goal entity, a distance metric is required. We directly use the graph distance[3] as the distance metric since the dot product of TransE embeddings does not provide

[3]The minimum length of the paths connecting two vertices.

good estimations of distances when the vertices are far away from each other. We traverse the graph to obtain a distance matrix $\mathbf{D}$ between all vertex pairs and then perform matrix factorization to get two low-dimensional matrices. Given a vertex, we retrieve the vector at the corresponding position as its distance embedding $\vec{\mathbf{e}}_d$.

Policy The policy model HiTKG-RL has the same architecture as HiTKG, while the MetaPath is modified: MP_{rl}=$[\vec{\mathbf{e}}_e; \vec{\mathbf{e}}_v; \vec{\mathbf{e}}_d]$, in order to incorporate the distance information. To maintain the same MetaPath structure between stage 1 and 2, we conduct the turn-level goal learning at stage 2 with MP_{rl} instead of MP (HiTKG performs best with MP at stage 1). The encoder and decoder inputs constitute the observable states. HiTKG-RL predicts a one-hop KG path at each step and tries to approach the global goal. We employ A2C[4] [234] to optimize the model.

Reward At the t-th turn, we directly obtain the distance of two vertices $d_t(\mathbf{v}_1, \mathbf{v}_2)$ from $\mathbf{D}$ and estimate the reward based on this. If $d_t(\mathbf{v}_{eop}, \mathbf{g}) < d_{t-1}(\mathbf{v}_{eop}, \mathbf{g})$, then the reward is set to 1, otherwise the reward is 0. $\mathbf{v}_{eop}$ denotes the ending entity of the path predicted. Currently, due to the lack of automatic KG path evaluation metrics, we use distance as the only criterion. The future work will introduce more evaluation criteria of KG paths such as naturalness.

5.4 Experiments and Results

5.4.1 Dataset

We conduct our evaluation on OpenDialKG [22]. It is a dialogue - KG dataset where each utterance of a dialogue is annotated with a KG path, which enables learning graph walkers to reason over the KG based on the conversations. It consists of 15K dialogues and 91K turns. Each dialogue is produced by two crowd workers and grounds on a given topic. We follow the baselines and split it into the train (70%), dev (15%), and test set (15%).

[4] A2C replaces the Q value of Actor-critic's gradient with the expected advantage and the learning process is more stable compared with policy gradient methods.

5.4.2 Experimental Settings

Baselines To evaluate stage 1 learning, we compare our results with six baseline models. However, we do not benchmark against previous work to evaluate stage 2 learning, since we can hardly find any similar work, or related codes are not available.

Generally, the six baseline models can be divided into breadth-centric and depth-centric models. Tri-LSTM [43] is a breadth-centric model that augments its dialogue inputs with wide-ranging shallow KG facts to retrieve short KG paths. The other five baselines and HiTKG are depth-centric models which focus on a small set of KG entity-relation connections and perform deep reasoning over the KG. Among them, Seq2Seq and DialKG Walker were proposed in [22], while Seq2Path, AttnFlow, and AttnIO were proposed in [21].

Implementation Details The MetaPath is the basic component of KG path representations, while we perform moderate modifications under different situations. When encoding an n-hop KG path history wherein the one-hop KG path components are in series connections, a starting MetaPath $\text{MP}_{bop}=[\vec{\mathbf{e}}_{bop};\vec{\mathbf{e}}_{1st}]$ is added to the beginning to indicate the starting point in KG.

In contrast, at the graph decoding stage, when predicting probability distribution over the one-hop neighbors of the current entity, MP_{bop} is not required, since all of the paths start from the current entity and they are in parallel relationships. In addition, stage 1 and 2 use MetaPaths of different structures, as stated in Section 5.3.4. We use Pytorch [235] to implement our model, which is trained on two RTX 8000 GPUs. We tune the hyperparameters by grid searching the hyperparameter space and choose the following settings that perform best: number of encoder/decoder layers: 2/6; dimension of the KG walker: 768; dimension of the KG embedding: 384 (stage 1), 256 (stage 2); loss coefficients γ/λ: 0.1/0.9; number of attention heads: 12; learning rate: 10^{-3}; dropout rate: 0.1; L2 regularization parameter ϵ: 10^{-5}; batch size: 10. We use learning rate scheduler to tune the learning rate manually and patient & early stopping to avoid overfitting. In addition, we use gradient clipping to avoid gradient explosions.

TABLE 5.1: Path-level (*path*@*k*) and target-level (*tgt*@*k*) performance of supervised KG path reasoning at stage 1 (metric: recall@k). HiTKG is benchmarked against several state-of-the-art baselines and ablation models on the OpenDialKG dataset.

Model	Recall@k									
	path@1	path@3	path@5	path@10	path@25	tgt@1	tgt@3	tgt@5	tgt@10	tgt@25
Tri-LSTM	3.2	14.2	22.6	36.3	56.2	-	-	-	-	-
Seq2Seq	3.1	18.3	29.7	44.1	60.2	-	-	-	-	-
DialKG Walker	13.2	26.1	35.3	47.9	62.2	-	-	-	-	-
Seq2Path	14.92	24.95	31.1	38.68	48.15	15.65	27.04	33.86	42.52	53.28
AttnFlow	17.37	24.84	30.68	39.48	51.4	18.97	36.23	45.48	58.84	71.35
AttnIO	23.72	37.53	43.57	52.17	62.86	24.98	43.78	53.49	65.48	78.79
HiTKG - 2EMP	24.16	37.01	46.78	58.12	67.83	30.01	43.22	55.19	66.73	85.36
HiTKG - DK	24.55	38.01	48.02	58.17	70.39	30.89	43.92	53.68	71.01	85.26
HiTKG - W2	25.82	38.56	48.63	57.93	71.25	30.99	46.29	**55.91**	70.88	85.15
HiTKG - WLH	24.49	37.31	49.10	58.22	70.12	30.13	43.29	54.76	70.16	85.52
HiTKG - SP	23.98	35.21	39.29	53.25	66.81	25.02	45.15	50.52	63.77	80.31
HiTKG-RL	25.12	36.55	45.67	56.37	69.18	30.99	**46.67**	54.16	67.20	85.19
HiTKG	**25.99**	**38.67**	**49.18**	**59.32**	**71.27**	**31.11**	46.29	55.59	**71.61**	**86.09**

5.4.3 Evaluation

Results The turn-level goal planning performance of baseline models and HiTKGs are presented in Table 5.1. Following the baselines, we use recall@k as the evaluation metric of path-level (*path*@*k*) and target entity-level (*tgt*@*k*) correctness. HiTKG outperforms all baselines we benchmark against in both *path*@*k* and *tgt*@*k*, with two metrics worse than the ablation models. The performance gain is significant, especially in recalls with larger k: there is a 13% relative improvement in *path*@25 and 9% in *tgt*@25. As illustrated in Section 5.3.4, at the second learning stage, we use a different MetaPath structure to represent the KG paths and KG neighbors for both supervised and reinforcement learning. Thus, we also report the performance of turn-level goal planning at stage 2. HiTKG-RL is designed for reinforcement learning, while it shows comparable performance with HiTKG when trained in a supervised fashion, even outperforming in *tgt*@3. This result indicates that the introduced distance embedding does not significantly influence the performance of turn-level goal planning at stage 2.

Tri-LSTM, Seq2Seq, Seq2Path, and DialKG Walker are recurrent graph walkers, which deliver history information with a fixed-length vector. The use of a fixed-length vector creates a performance bottleneck in KG reasoning and these recurrent

baseline models show at least 42.59% lower performance than HiTKG in *path*@1. In addition, recurrent units suffer from short memories, restricting the performance in long KG path predictions. The trajectory is a good form to represent dynamic information and we leverage dialogue history and KG path history as two trajectory sources for goal planning. Most of the baselines omit the KG path history and only learn utterance patterns for KG walking. However, KG path history records the KG trajectory up to the previous turn and is an important guide to a KG walker, e.g., the model knows which paths have been walked in previous turns, which avoids or reduces repeated attempts. DialKG Walker and AttnIO are two state-of-the-art KG walkers. The recurrent architecture of DialKG Walker limits itself in feature representation, which causes its comparatively low performance, especially in recalls with small k.

Besides, when computing the context vector at the decoding stage, it needs to compute attention scores over the whole relation space, which can be computationally expensive and may affect the quality of the resulting context vector. AttnIO computes an incoming attention flow to represent entities and an outgoing attention flow to select KG paths. This design ensures an optimum path at the decoding stage because it reserves all potential paths at each step. However, to predict a T-hop path where each node has N neighbors on average, the computation of outgoing attention flow has a complexity of $O(N^T)$, which is not scalable to long KG path predictions. In addition, all baseline models separately deal with the entity and relation paths, which breaks the semantic structure of KG paths.

Ablation Study We conduct five ablation studies as reported in Table 5.1. **(1)** First, we experiment with the 2-entity MetaPaths (2EMP) where MP=$[\vec{\mathbf{e}}_{v_1}; \vec{\mathbf{e}}_e; \vec{\mathbf{e}}_{v_2}]$. The performance degradation suggests that the redundancy of entity information harms the training. **(2)** Next, the encoder-decoder attention layers α_{kg} and α_{dial} are swapped (DK). Placing the layer α_{kg} in front of α_{dial} outperforms the reversed condition, which implies that it is more reasonable to select low-level information (dialogue history) with a higher one (path history), demonstrating a better way to compute hierarchical attention. **(3)** We test the performance of supervised path learning without the utterances from speaker 2 (W2). We find that, although performance is slightly degraded, results are still comparable, even higher than the standard setting in *tgt*@5. We infer that this is because given speaker 1 (user) and speaker 2 (agent), speaker 2 will pay more attention to the utterances from

speaker 1 instead of his own for goal planning. In addition, the KG path history contains most of the essential information in the utterances from speaker 2. **(4)** To investigate the contribution of OLH, we train the KG walker without it (WLH) and this causes performance to drop 1-2%. **(5)** The fifth ablation model separately predicts entity and relation paths (SP), using both distributions for one-hop KG path reranking at each decoding step. A drop in performance suggests the contribution of MetaPath, which concurrently considers entity and relation information.

TABLE 5.2: The success rate of reaching the global goal entity.

Model	Distance / Success Rate (%)				
	3	5	7	9	11
HiTKG	2	0	0	0	0
HiTKG-RL	66	27	11	2	0

Success Rate Whether the agent can reach the global goal entity is a natural way to evaluate whether stage 2 works. For each case, we randomly select a beginning node $\mathbf{v}_{1st}$ and a target global goal $\mathbf{g}$ which has a graph distance of 3/5/7/9/11 from $\mathbf{v}_{1st}$. We report and compare the success rate of 100 independent attempts by HiTKG and HiTKG-RL, respectively, as shown in Table 5.2. The HiTKG is only trained at stage 1 while HiTKG-RL undergoes both stages. It is indicated that without global goal guided training, the HiTKG can barely succeed (only 2 cases succeeded by chance). Whereas HiTKG-RL has a 66% success rate at distance 3 and declines as the distance rises. The decline is partially ascribed to the trade-off between naturalness and success.

TABLE 5.3: Ranking results of the semantic closeness between ending node and global goal, and the path naturalness. The results are presented as the number of instances a certain model is ranked as a certain ranking (averaged).

Model	Naturalness				Semantic Closeness			
	1st	2nd	3rd	4th	1st	2nd	3rd	4th
HT	9	5.8	18.2	67	41	33.8	10.2	15
GT	39.2	27.8	24	9	12.6	17.2	39.8	30.4
Hi	32.6	43.2	18	6.2	7	11.8	41.2	40
Hi-RL	19.2	23.2	39.8	17.8	39.4	37.2	8.8	14.6

Human Evaluation We aim to plan natural turn-level goals at stage 1, while at stage 2 we aim to approach the target without losing naturalness. We conduct human evaluation to further evaluate the path naturalness of both stages and the approaching effectiveness of stage 2. We sample 100 2-hop KG paths from Shortest Path (SP), Ground Truth (GT), HiTKG (Hi), and HiTKG-RL (Hi-RL), respectively, and combine them into 100 four-tuples where each path tuple corresponds to a certain starting entity $\mathbf{v}_{1st}$ in OpenDialKG and global goal $\mathbf{g}$ which is 3 units away from $\mathbf{v}_{1st}$. Each path starts from $\mathbf{v}_{1st}$ and does not reach $\mathbf{g}$. SP denotes the shortest paths from $\mathbf{v}_{1st}$ to $\mathbf{g}$. Hi is only trained at stage 1 while Hi-RL undergoes both stages. Five human evaluators independently rank the four paths in each tuple regarding the path naturalness and the semantic closeness between the ending node and global goal. As shown in Table 5.3, GT and Hi paths show dominance in top rankings of naturalness, which indicates the effectiveness of stage 1; Hi-RL paths have significantly more instances than SP in top naturalness rankings, indicating that Hi-RL reserves naturalness when learning to approach global goal (comparing with Hi, it sacrifices some naturalness to approach the target). Hi-RL paths rank first in 39.4% closeness rankings, only second to the SP. It indicates that Hi-RL effectively approaches the target in semantic space when reasoning over the KG.

TABLE 5.4: Comparison of KG paths generated from models trained at stage 1 under a context (including ground truth).

Conversation	P1: Fiona Stafford wrote Emma. It's a romance novel. Are you into that genre? P2: Any other books that might fall under *comedy*? I'm in the mood for something light. P1: [response]
GT	([BOP], Comedy) → (genre of, One Crazy Summer)
AttnFlow	([BOP], Comedy) → (parent genre, Slapstick)
AttnIO	([BOP], Comedy) → (subject of, The War of the Worlds) → (written by, Arthur. C. Clarke)
HiTKG (WLH)	([BOP], Comedy) → (genre of, Slacker) → (release year, 1991)
HiTKG	([BOP], Comedy) → (genre of, One Crazy Summer)

Case Study Table 5.4 presents the example KG path predictions from models trained at stage 1 under the same conversational context in [21]. With the starting entity *Comedy*, the proposed HiTKG predicts precisely in both content and length. In contrast, the baseline models tend to select paths that are different from the ground truth, which lacks naturalness and may not fit the context. In addition, we also showcase the output of an ablation model which predicts without the OLH. The generated path is not consistent in length compared with the ground truth.

TABLE 5.5: Comparison of KG path selections among neighbor candidates under global goal guidance and a given context.

Conversation	P1: Do you have books by Yann Martel? P2: He wrote Life of Pi and *Beatrice and Virgil*. Have you read either of those? P1: [response]
Candidates	(in language, English) [d=1]; (release year, 2010) [d=3]; (written by, Yann Martel) [d=3]
Global Goal	*An American in Hollywood*
HiTKG	([BOP], Beatrice and Virgil) → (release year, 2010)
HiTKG-RL	([BOP], Beatrice and Virgil) → (in language, English)

We compare how KG walkers choose one-hop paths given a global goal, as shown in Table 5.5. At the node *Beatrice and Virgil*, there are three candidate paths to select, which ends with entities that are 1/3/3 units away from the global goal *An American in Hollywood*. In general, the walker trained with both stages tends to choose the entity that is the closest to the target, in both topological and semantic spaces.

5.5 Summary

We propose HiTKG, a hierarchical Transformer based KG walker that leverages multiscale inputs for graph reasoning in dialogues. HiTKG first learns to plan natural turn-level goals and then learns to approach a global goal. Both automatic and human evaluation illustrate the effectiveness of our method. In the future, we will investigate how to improve the embedding, learning framework, and evaluation criteria of stage 2 to further extend this topic.

ChatGPT [236] is a recent sub-symbolic dialogue system trained with cutting-edge supervised and reinforcement learning techniques. There is no comparable result to benchmark our HiTKG against it. However, HiTKG's (neural-symbolic) advantage is that it incorporates KG as the external knowledge, thus possibly saving a large amount of data and achieving more interpretable and robust reasoning. On the other hand, the highly logical responses of ChatGPT indicate that, with correct training techniques and enough data, the sub-symbolic approaches are also capable of learning strong reasoning abilities.

Chapter 6

Conclusion and Future Work

This section summarizes the thesis and demonstrates some potential future works that deserve exploring. In this thesis, we propose state-of-the-art models that show superior performances in language transduction (MT, LM, and AS), understanding (ID, SF, and ER), and reasoning (DCR). We focus on optimizing the attention mechanism for better sub-symbolic and neural-symbolic architectures, which sheds light on future architectural designs for NLP tasks.

6.1 Conclusion

This thesis studies optimizing attention mechanisms for sub-symbolic-based and neural-symbolic-based NLP tasks. The attention-based architectures are optimized to solve the inherent issues of the attention mechanism itself such as redundancy and over-parameterization, or issues that exist in the tasks such as the granularity gap in SLU and the short-sighted problems in DCR.

We first try to solve the redundancy and over-parameterization issues of multi-head attention. We assume that only focusing on the most essential and different features may mitigate the above issues. In particular, we propose the Grouped Head Attention (GHA) trained with a self-supervised group constraint that forces MHA to divide its heads to work in several separate groups, where each group focuses on an essential but unique feature subset. We further propose a Voting-to-Stay procedure to remove redundant parameters in GHA and obtain GHA-PS,

a lighter-weight attention. The improvements on three tasks and the extensive analysis verify our hypothesis and the effectiveness of our redundancy optimization methods. Our study provides guidance on future MHA design and training to achieve higher performance.

We then ease the modality and granularity inconsistency problem when transferring knowledge for Spoken Language Understanding tasks, by refining the attention hidden states based on the attention map distribution. We propose the Prior-informed Adaptive knowledge Distillation (PAD) that leverages adaptive spans and the in-stored significance prior distribution to achieve better alignments between well-established text and speech pre-trained models. The experimental results and analysis illustrate the effectiveness of PAD. We also observe that the trade-off between global and local alignments always exists because they may be contradictory in essence.

Finally, we improve the multi-source and long-term commonsense reasoning in dialogues with a hierarchical attention-based decoding block. We present HiTKG, a hierarchical attention-based graph walker that leverages multiscale inputs to make precise and flexible reasoning on KG paths. Furthermore, we propose a two-hierarchy learning framework that employs two stages to learn both turn-level (short-term) and global-level (long-term) KG entities as conversation topics. Specifically, in the first stage, HiTKG is trained in a supervised fashion to learn how to plan turn-level topic sequences; in the second stage, HiTKG tries to naturally approach the assigned global topic via reinforcement learning. In addition, we propose MetaPath as the backbone method for KG path representation to exploit the entity and relation information concurrently. We further propose Multi-source Decoding Inputs and Output-level Length Head to improve the decoding controllability of the proposed hierarchical attention. Our experiments show that HiTKG achieves a significant improvement in the performance of turn-level topic learning compared to state-of-the-art baselines. Additionally, both automatic and human evaluation prove the effectiveness of the two-hierarchy learning framework for both short-term and long-term KG reasoning.

6.2 Future Work

6.2.1 Bias-free Attention-based Architectures

Different NLP tasks require different inductive biases. For tasks with short inputs, the main concern is to look at the whole sequence and have each head looking at different aspects [26]; whereas for tasks with long inputs, the main issue is the quadratic complexity of the attention map computation [17, 18], hence local receptive fields are preferred to reduce computation complexity and out-of-memory issues. Moreover, tasks of different fields present different model inductive biases as well. For example, NLP tasks assume that temporal dependencies exist between tokens, whereas Computer Vision (CV) tasks require more spatial dependencies.

A meaningful future study will be exploring attention-based architectures that do not rely on task-specific or input-specific inductive biases, and are thus scalable to tasks of very different forms. Current models usually depend heavily on their corresponding inductive biases and may have a large variance in their performances when learning different tasks [237]. Thus, a single attention-based architecture that does not rely heavily on specific inductive biases is more desirable, and it is more similar to human brains that are well scalable to various tasks.

6.2.2 Universal Pruning Algorithms

The current attention-based models, such as Transformers, are heavy in their parameters, which is caused by either the Multi-Head Attention (MHA) [15, 16] or Feed-Forward Network (FFN) [27, 36, 182] modules of them. In this thesis, based on the Lottery Ticket hypothesis [185], we propose a divide-and-conquer strategy that prunes unimportant attention heads. However, it is only applicable to models that have completed Group-Constrained Training (GCT). Moreover, it prunes different heads for different tasks, which means that a model is not reusable for other tasks once pruned. To this end, a universal pruning algorithm that directly prunes on untrained/pre-trained Transformers without harming the model's performance and transferability will be impactful to the whole AI community. It increases the efficiency of untrained/pre-trained models from various domains without additional training and at a low cost.

6.2.3 Interpretable Alignments between Pre-trained Models

In this thesis, we show that it is possible to achieve alignments between different pre-trained models by refining the attention representations according to the priors. In this way, the representations of the two models become similar so that knowledge learned by one of them in a certain task is transferred to the other one that solves other tasks. Currently, the alignments are end-level soft alignments that do not exactly follow the original mapping relationships between two representations [20] (e.g., in speech-text pairs, a non-blank speech frame can be translated to a text token, thus forming a mapping relationship). Such alignments lack interpretability and are not reliable despite their success on various downstream understanding tasks. As such, there remains space for researching interpretable alignments between pre-trained models of various domains such as text, speech, vision, etc.

6.2.4 Ethical Considerations and Bias Mitigation of Large Language Models

As large language models become increasingly integrated into real-world applications, addressing ethical concerns and mitigating biases within these models becomes paramount. Future research should emphasize developing frameworks and techniques to identify and rectify biases in training data and model outputs. Incorporating fairness and diversity metrics into the training process can help ensure equitable and unbiased responses from language models, making them more reliable and suitable for diverse user populations.

6.2.5 Continual Learning and Lifelong Adaptation

Large language models have traditionally been trained on static datasets, limiting their ability to adapt to new information and evolving contexts. Future research should focus on developing methods for continual learning, enabling language models to incrementally acquire knowledge and refine their understanding over time. By leveraging techniques like transfer learning and reinforcement learning, models

can continually update their knowledge base, adapt to new domains or languages, and remain up-to-date with the latest information.

Appendix A

Additional Settings for Chapter 4

A.1 Trainig Settings

Machine Translation We use Adam to optimize the MT models and set the $\beta_1 = 0.9, \beta_2 = 0.98$. We use the Inverse Square Root Schedule [23] where it first warms up for 4K steps until the learning rate reaches 5×10^{-4}, and then it exponentially decays the learning rate. We apply early stop as a terminate condition. We apply a 0.3 dropout rate for all Machine Translation models. A weight decay of 10^{-4} is used for all IWSLT 2014 models, whereas for WMT models we use a weight decay of 0. We apply a 0.1 label smoothing [238, 239] for the uniform prior distribution over the vocabulary.

Language Modeling Following Baevski and Auli [46], we use Nesterov's accelerated gradient method [240] with a momentum of 0.99. We clip the gradient norm if it exceeds 0.1[241]. The learning rate is linearly warmed up from 10^{-7} to 1 for 16K steps and then annealed using a cosine learning rate schedule[242] with multiple cycles. Each cycle doubles the number of updates than the previous cycle and we shrink the maximum and minimum learning rates by 0.75 compared to the previous cycle. The initial minimum learning rate is 10^{-4} and the maximum is 1. We apply 0.2 adaptive softmax dropout rate, 0.1 attention dropout rate, and 0.1 activation dropout rate.

TABLE A.1: The configuration of α, β, and Feature Maps (FM, including $\hat{\mathbf{V}}$, $\mathbf{A}$, and $\mathbf{O}$) for GHT and GHT-PS on different Machine Translation datasets.

Model	IWSLT ($\alpha/\beta/FM$)					WMT ($\alpha/\beta/FM$)	
	de-en	it-en	en-de	en-it	en-fr	en-de	en-fr
GHT	0.7/0.5/$\hat{\mathbf{V}}$	0.3/0.5/$\hat{\mathbf{V}}$	0.3/0.1/$\mathbf{A}$	0.3/0.3/$\hat{\mathbf{V}}$	0.7/0.7/$\hat{\mathbf{V}}$	0.5/0.5/$\hat{\mathbf{V}}$	0.3/0.3/$\hat{\mathbf{V}}$
GHT-PS	0.5/0.7/O	0.3/0.3/$\mathbf{A}$	0.3/0.7/$\mathbf{O}$	0.3/1/$\hat{\mathbf{V}}$	0.5/0.3/$\mathbf{A}$	0.5/0.5/$\hat{\mathbf{V}}$	0.3/0.3/$\hat{\mathbf{V}}$

TABLE A.2: The configuration of α, β, and Feature Maps (FM, including $\hat{\mathbf{V}}$, $\mathbf{A}$, and $\mathbf{O}$) for GHT and GHT-PS in Abstractive Summarization and Language Modeling.

Model	$\alpha/\beta/FM$
GHT	0.5/0.5/$\hat{\mathbf{V}}$
GHT-PS	0.5/0.5/$\hat{\mathbf{V}}$

Abstractive Summarization We use the same training set up with IWSLT 2014 models. We apply 0.1 clip norm and 0.2 attention dropout. The model is warmed up for 10K updates.

A.2 GHT model settings

Machine Translation configurations are detailed in Table A.1; Language Modeling and Abstractive Summarization are detailed in Table A.2.

Note that $\varphi(\mathbf{v}_{i,l}, \mathbf{a}_{i,l}, \mathbf{o}_{i,l}; \mathbf{z}_{i,l}) = \tau_1\varphi(\mathbf{v}_{i,l}; \mathbf{z}_{i,l}) + \tau_2\varphi(\mathbf{a}_{i,l}; \mathbf{z}_{i,l}) + \tau_3\varphi(\mathbf{o}_{i,l}; \mathbf{z}_{i,l})$, we set one of the $\{\tau_1, \tau_2, \tau_3\}$ to be 1, the other to be 0.

A.3 Datasets and Evaluation

For all three tasks, we follow the data pipeline of fairseq[1].

Machine Translation (MT). To fully evaluate the effectiveness of our methods, we evaluate seven MT datasets of IWSLT'14 and WMT 2014 benchmarks. We closely follow the setup of Vaswani et al. [23] for data preparation. WMT

[1] https://github.com/facebookresearch/fairseq/blob/main/examples

2014 English-German dataset consists of about 4.5M sentence pairs. It is encoded with byte-pair encoding [243], having a shared source-target vocabulary of about 40K tokens. Following the standard setting [23], we validate on newstest2013 and test on newstest2014 for experiments on this dataset. The WMT 2014 English-French dataset consists of 36M sentence pairs and is encoded with a joint source-target BPE of about 43K vocabularies. Following the standard split, we validate on a concatenation of newstest2012 and newstest2013 and test on newstest2014. For IWSLT'14 German to English, IWSLT'14 English to German, IWSLT'14 English to French, IWSLT'14 English to Italian and IWSLT'14 Italian to English, we encode the sentence pairs with joint source-target BPE. Following Edunov et al. [244], the validation set is randomly splited from the training set with a ratio of 1:23. The testset consists TED.tst2010, TED.tst2011, TED.tst2012 and TED.dev2010, TEDX.dev2012 for IWSLT'14 German to English, IWSLT'14 English to German, and IWSLT'14 English to French; the TEDX.dev2012 is replaced by TEDX.dev2014 for IWSLT'14 English to Italian and IWSLT'14 Italian to English.

For all Machine Translation datasets, we use detokenized BLEU. WMT 2014 English-German and WMT 2014 English-French are evaluated with a beam size 4 and length penalty 0.6; IWSLT'14 datasets are evaluated with a beam size 5 and without length penalty.

Language Modeling (LM). We evaluate the LM of GHT on WIKITEXT-103 [32] which has about 100M tokens and a 260K BPE vocabulary. Following Baevski and Auli [46], we use perplexity as an evaluation metric and a context window of 2047 at the inference stage.

Abstractive Summarization (AS). We also evaluate on CNN-DailyMail [245] for AS to test the ability of GHT in hard tasks with long inputs. The dataset comprises over 280K news articles paired with multi-sentence summaries. Following Wu et al. [36], articles are truncated to 512 tokens and encoded with 50K BPE. We use F1-Rouge [246] to evaluate the performance, including Rouge-1, Rouge-2 and Rouge-L.

Bibliography

[1] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 253–263, Taipei, Taiwan, 2017. Asian Federation of Natural Language Processing. URL `https://aclanthology.org/I17-1026`. xvii, 28

[2] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 3776–3784. AAAI Press, 2016. URL `http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11957`. xvii, 35, 36

[3] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2440–2448, 2015. URL `https://proceedings.neurips.cc/paper/2015/hash/8fb21ee7a2207526da55a679f0332de2-Abstract.html`. xvii, 37, 40

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1409.0473`. xvii, 7, 22, 39, 40

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. xvii, 3, 6, 7, 15, 18, 41, 42, 83

[6] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama,

and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2692–2700, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/29921001f2f04bd3baee84a12e98098f-Abstract.html. xvii, 43, 44

[7] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany, 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1154. URL https://aclanthology.org/P16-1154. xvii, 46

[8] Wikipedia contributors. Intelligence, 11 2022. URL https://en.wikipedia.org/wiki/Intelligence. 1

[9] Eleni Ilkou and Maria Koutraki. Symbolic vs sub-symbolic ai methods: Friends or enemies? In *CIKM (Workshops)*, 2020. 1

[10] Jinjie Ni, Tom Young, Vlad Pandelea, Fuzhao Xue, Vinay Adiga, and Erik Cambria. Recent advances in deep learning based dialogue systems: A systematic survey. *Artif Intell Rev (2022). https://doi.org/10.1007/s10462-022-10248-8*, 2022. 2, 3, 77

[11] Artur d'Avila Garcez, Tarek R Besold, Luc De Raedt, Peter Földiak, Pascal Hitzler, Thomas Icard, Kai-Uwe Kühnberger, Luis C Lamb, Risto Miikkulainen, and Daniel L Silver. Neural-symbolic learning and reasoning: contributions and challenges. In *2015 AAAI Spring Symposium Series*, 2015. 2

[12] Jing Zhang, Bo Chen, Lingxi Zhang, Xirui Ke, and Haipeng Ding. Neural, symbolic and neural-symbolic reasoning on knowledge graphs. *AI Open*, 2: 14–35, 2021. doi: 10.1016/j.aiopen.2021.03.001. URL https://doi.org/10.1016/j.aiopen.2021.03.001. 2

[13] Jinjie Ni, Vlad Pandelea, Tom Young, Haicang Zhou, and Erik Cambria. Hitkg: Towards goal-oriented conversations via multi-hierarchy learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36-10, pages 11112–11120, 2022. 2, 3, 6, 9, 15, 27, 77

[14] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *CoRR*, abs/2106.04554, 2021. URL https://arxiv.org/abs/2106.04554. 3, 4

[15] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS*

2019, December 8-14, 2019, Vancouver, BC, Canada, pages 14014–14024, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/2c601ad9d2ff9bc8b282670cdd54f69f-Abstract.html. 4, 19, 49, 50, 52, 56, 59, 97

[16] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers*, pages 5797–5808. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1580. URL https://doi.org/10.18653/v1/p19-1580. 4, 19, 49, 50, 52, 56, 58, 59, 64, 97

[17] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *CoRR*, abs/2006.04768, 2020. URL https://arxiv.org/abs/2006.04768. 4, 6, 17, 97

[18] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5156–5165. PMLR, 2020. URL http://proceedings.mlr.press/v119/katharopoulos20a.html. 4, 6, 16, 97

[19] Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. Star-transformer. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1315–1325. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1133. URL https://doi.org/10.18653/v1/n19-1133. 4

[20] Yu-An Chung et al. SPLAT: speech-language joint pre-training for spoken language understanding. In *NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1897–1907. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.152. URL https://doi.org/10.18653/v1/2021.naacl-main.152. 4, 6, 68, 71, 72, 73, 74, 98

[21] Jaehun Jung, Bokyung Son, and Sungwon Lyu. Attnio: Knowledge graph exploration with in-and-out attention flow for knowledge-grounded dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3484–3497, 2020. 4, 10, 27, 78, 80, 82, 88, 93

[22] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 845–854, 2019. 4, 10, 26, 27, 78, 80, 82, 87, 88

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html`. 5, 49, 57, 58, 59, 101, 102, 103

[24] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`. 6

[25] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition. In Helen Meng, Bo Xu, and Thomas Fang Zheng, editors, *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 5036–5040. ISCA, 2020. doi: 10.21437/Interspeech.2020-3015. URL `https://doi.org/10.21437/Interspeech.2020-3015`. 6

[26] Jian Li, Zhaopeng Tu, Baosong Yang, Michael R. Lyu, and Tong Zhang. Multi-head attention with disagreement regularization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2897–2903. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-1317. URL `https://doi.org/10.18653/v1/d18-1317`. 6, 19, 50, 52, 55, 58, 59, 97

[27] Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. Lite transformer with long-short range attention. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=ByeMPlHKPH`. 6, 21, 49, 52, 58, 59, 72, 97

[28] Shuoheng Yang, Yuxin Wang, and Xiaowen Chu. A survey of deep learning techniques for neural machine translation. *CoRR*, abs/2002.07526, 2020. URL `https://arxiv.org/abs/2002.07526`. 6

[29] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL, 2002. doi: 10.3115/1073083.1073135. URL `https://aclanthology.org/P02-1040/`. 7, 59

[30] In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, 2003. 7

[31] Baevski Alexei, Edunov Sergey, Liu Yinhan, and Auli Michael. Cloze-driven pretraining of self-attention networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5360–5369, 2019. 7

[32] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=Byj72udxe`. 8, 103

[33] Ciprian Chelba, Tomás Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. In Haizhou Li, Helen M. Meng, Bin Ma, Engsiong Chng, and Lei Xie, editors, *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, pages 2635–2639. ISCA, 2014. URL `http://www.isca-speech.org/archive/interspeech_2014/i14_2635.html`. 8

[34] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *CoRR*, abs/2004.05150, 2020. URL `https://arxiv.org/abs/2004.05150`. 8, 16

[35] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1285. URL `https://aclanthology.org/P19-1285`. 8

[36] Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. In *7th International Conference on Learning Representations, ICLR 2019, New*

Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019. URL https://openreview.net/forum?id=SkVhlh09tX. 8, 21, 49, 52, 58, 59, 64, 97, 103

[37] Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In Yoav Goldberg and Stefan Riezler, editors, *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 280–290. ACL, 2016. doi: 10.18653/v1/k16-1028. URL https://doi.org/10.18653/v1/k16-1028. 8

[38] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 8, 64

[39] Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. Speech model pre-training for end-to-end spoken language understanding. In Gernot Kubin and Zdravko Kacic, editors, *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 814–818. ISCA, 2019. doi: 10.21437/Interspeech.2019-2396. URL https://doi.org/10.21437/Interspeech.2019-2396. 9

[40] Natalia Tomashenko, Antoine Caubrière, Yannick Estève, Antoine Laurent, and Emmanuel Morin. Recent advances in end-to-end spoken language understanding. In *International Conference on Statistical Language and Speech Processing*, pages 44–55. Springer, 2019. 9

[41] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*, 2018. 9

[42] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan. IEMOCAP: interactive emotional dyadic motion capture database. *Lang. Resour. Evaluation*, 42(4):335–359, 2008. doi: 10.1007/s10579-008-9076-6. URL https://doi.org/10.1007/s10579-008-9076-6. 9

[43] Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. Augmenting end-to-end dialogue systems with commonsense knowledge. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 4970–4977, 2018. 10, 26, 78, 88

[44] Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. Grounded conversation generation as guided traverses in commonsense knowledge graphs. *arXiv preprint arXiv:1911.02707*, 2019. 80

[45] Wei Li, Wei Shao, Shaoxiong Ji, and Erik Cambria. BiERU: Bidirectional emotional recurrent unit for conversational sentiment analysis. *Neurocomputing*, 467:73–82, 2022. 10, 78

[46] Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=ByxZX20qFQ`. 15, 64, 101, 103

[47] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018. 15, 27

[48] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *CoRR*, abs/1904.10509, 2019. URL `http://arxiv.org/abs/1904.10509`. 16

[49] Joshua Ainslie, Santiago Ontañón, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. ETC: encoding long and structured inputs in transformers. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 268–284. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.19. URL `https://doi.org/10.18653/v1/2020.emnlp-main.19`. 16

[50] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018. URL `http://arxiv.org/abs/1807.03748`. 16

[51] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/c8512d142a2d849725f31a9a7a361ab9-Abstract.html`. 16

[52] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Jared Davis, Tamás Sarlós, David Belanger, Lucy J. Colwell, and Adrian Weller. Masked language modeling for proteins via linearly scalable long-context transformers. *CoRR*, abs/2006.03555, 2020. URL `https://arxiv.org/abs/2006.03555`. 16

[53] Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A. Smith, and Lingpeng Kong. Random feature attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=QtTKTdVrFBB. 16

[54] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J. Colwell, and Adrian Weller. Rethinking attention with performers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=Ua6zuk0WRH. 16

[55] Qipeng Guo, Xipeng Qiu, Xiangyang Xue, and Zheng Zhang. Low-rank and locality constrained self-attention for sequence modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(12):2213–2222, 2019. 17

[56] Ziye Chen, Mingming Gong, Lingjuan Ge, and Bo Du. Compressed self-attention for deep metric learning with low-rank approximation. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 2058–2064. ijcai.org, 2020. doi: 10.24963/ijcai.2020/285. URL https://doi.org/10.24963/ijcai.2020/285. 17

[57] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14138–14148. AAAI Press, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17664. 17

[58] Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. Modeling localness for self-attention networks. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4449–4458. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-1475. URL https://doi.org/10.18653/v1/d18-1475. 17

[59] Maosheng Guo, Yu Zhang, and Ting Liu. Gaussian transformer: A lightweight approach for natural language inference. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence,*

EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pages 6489–6496. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33016489. URL https://doi.org/10.1609/aaai.v33i01.33016489. 17

[60] Yujing Wang, Yaming Yang, Jiangang Bai, Mingliang Zhang, Jing Bai, Jing Yu, Ce Zhang, and Yunhai Tong. Predictive attention transformer: Improving transformer with attention map prediction. 2020. 17

[61] Ruining He, Anirudh Ravula, Bhargav Kanagal, and Joshua Ainslie. Realformer: Transformer likes residual attention. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 929–943. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-acl.81. URL https://doi.org/10.18653/v1/2021.findings-acl.81. 18

[62] Chengxuan Ying, Guolin Ke, Di He, and Tie-Yan Liu. Lazyformer: Self attention with lazy update. *CoRR*, abs/2102.12702, 2021. URL https://arxiv.org/abs/2102.12702. 18

[63] Biao Zhang, Deyi Xiong, and Jinsong Su. Accelerating neural transformer via an average attention network. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1789–1798. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1166. URL https://aclanthology.org/P18-1166/. 18

[64] Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Synthesizer: Rethinking self-attention in transformer models. *CoRR*, abs/2005.00743, 2020. URL https://arxiv.org/abs/2005.00743. 18

[65] Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of BERT. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4364–4373. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1445. URL https://doi.org/10.18653/v1/D19-1445. 19, 72

[66] Ameet Deshpande and Karthik Narasimhan. Guiding attention for self-supervised learning with transformers. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4676–4686. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.419. URL https://doi.org/10.18653/v1/2020.findings-emnlp.419. 19

[67] Noam Shazeer, Zhenzhong Lan, Youlong Cheng, Nan Ding, and Le Hou. Talking-heads attention. *CoRR*, abs/2003.02436, 2020. URL `https://arxiv.org/abs/2003.02436`. 19, 52

[68] Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1150–1159, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1106. URL `https://aclanthology.org/P17-1106`. 19, 52

[69] Jiaoda Li, Ryan Cotterell, and Mrinmaya Sachan. Differentiable subset pruning of transformer heads. *Trans. Assoc. Comput. Linguistics*, 9:1442–1459, 2021. doi: 10.1162/tacl_a_00436. URL `https://doi.org/10.1162/tacl_a_00436`. 19, 50, 52, 58, 59

[70] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. Multi-head attention: Collaborate instead of concatenate. *CoRR*, abs/2006.16362, 2020. URL `https://arxiv.org/abs/2006.16362`. 19, 50, 52, 58, 59

[71] Mikel L Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M Tyers. Apertium: a free/open-source platform for rule-based machine translation. *Machine translation*, 25(2):127–144, 2011. 20

[72] Philipp Koehn, Franz J Och, and Daniel Marcu. Statistical phrase-based translation. Technical report, University of Southern California Marina Del Rey Information Sciences Inst, 2003. 20

[73] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014. 20, 39

[74] Ankur Bapna, Mia Xu Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. Training deeper neural machine translation models with transparent attention. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3028–3033. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-1338. URL `https://doi.org/10.18653/v1/d18-1338`. 20

[75] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. Learning deep transformer models for machine translation. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1810–1822. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1176. URL `https://doi.org/10.18653/v1/p19-1176`. 20

[76] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=HyzdRiR9Y7`. 20

[77] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 464–468. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-2074. URL `https://doi.org/10.18653/v1/n18-2074`. 21

[78] Sergey Edunov, Alexei Baevski, and Michael Auli. Pre-trained language model representations for language generation. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4052–4059. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1409. URL `https://doi.org/10.18653/v1/n19-1409`. 21

[79] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018. URL `http://arxiv.org/abs/1802.05365`. 21

[80] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL `https://doi.org/10.18653/v1/n19-1423`. 21, 22, 67

[81] Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.54. URL `https://aclanthology.org/2020.acl-main.54`. 21

[82] Stéphane Clinchant, Kweon Woo Jung, and Vassilina Nikoulina. On the use of BERT for neural machine translation. In Alexandra Birch, Andrew M.

Finch, Hiroaki Hayashi, Ioannis Konstas, Thang Luong, Graham Neubig, Yusuke Oda, and Katsuhito Sudoh, editors, *Proceedings of the 3rd Workshop on Neural Generation and Translation@EMNLP-IJCNLP 2019, Hong Kong, November 4, 2019*, pages 108–117. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-5611. URL https://doi.org/10.18653/v1/D19-5611. 21

[83] Wilson L Taylor. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433, 1953. 21

[84] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. MASS: masked sequence to sequence pre-training for language generation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR, 2019. URL http://proceedings.mlr.press/v97/song19d.html. 22

[85] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html. 22, 67

[86] Yinhan Liu et al. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 22, 67

[87] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/c20bb2d9a50d5ac1f713f8b34d9aac5a-Abstract.html. 22

[88] Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/c04c19c2c2474dbf5f7ac4372c5b9af1-Abstract.html. 22

[89] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020. 22

[90] Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. Structbert: Incorporating language structures into pre-training for deep language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=BJgQ4lSFPH`. 22

[91] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112, 2014. URL `https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html`. 22

[92] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1099. URL `https://doi.org/10.18653/v1/P17-1099`. 22

[93] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL `https://openreview.net/forum?id=HkACllQgA-`. 22, 64

[94] Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. Bottom-up abstractive summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4098–4109. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-1443. URL `https://doi.org/10.18653/v1/d18-1443`. 22

[95] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/p16-1008. URL `https://doi.org/10.18653/v1/p16-1008`. 22

[96] Yen-Chun Chen and Mohit Bansal. Fast abstractive summarization with reinforce-selected sentence rewriting. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 675–686. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1063. URL `https://aclanthology.org/P18-1063/`. 22

[97] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL `https://aclanthology.org/N18-1202`. 22, 43

[98] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 22

[99] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3728–3738. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1387. URL `https://doi.org/10.18653/v1/D19-1387`. 23

[100] Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv preprint arXiv:2001.04063*, 2020.

[101] Dan Su, Yan Xu, Tiezheng Yu, Farhad Bin Siddique, Elham J. Barezi, and Pascale Fung. Caire-covid: A question answering and multi-document summarization system for COVID-19 research. *CoRR*, abs/2005.03975, 2020. URL `https://arxiv.org/abs/2005.03975`.

[102] Tiezheng Yu, Dan Su, Wenliang Dai, and Pascale Fung. Dimsum @laysumm 20: Bart-based approach for scientific document summarization. *CoRR*, abs/2010.09252, 2020. URL `https://arxiv.org/abs/2010.09252`. 23

[103] Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9332–9346. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.750. URL `https://doi.org/10.18653/v1/2020.emnlp-main.750`. 23

[104] Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. Neural text summarization: A critical evaluation. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 540–551. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1051. URL `https://doi.org/10.18653/v1/D19-1051`.

[105] Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. Faithful to the original: Fact aware neural abstractive summarization. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4784–4791. AAAI Press, 2018. URL `https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16121`.

[106] Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. Dialogue natural language inference. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3731–3741. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1363. URL `https://doi.org/10.18653/v1/p19-1363`. 23

[107] Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4715–4728. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.428. URL `https://doi.org/10.18653/v1/2020.acl-main.428`. 23

[108] Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cícero Nogueira dos Santos, Henghui Zhu, Dejiao Zhang, Kathy McKeown, and Bing Xiang. Entity-level factual consistency of abstractive text summarization. In Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 2727–2733. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.eacl-main.235. URL `https://doi.org/10.18653/v1/2021.eacl-main.235`. 23

[109] Angela Fan, David Grangier, and Michael Auli. Controllable abstractive summarization. In Alexandra Birch, Andrew M. Finch, Minh-Thang Luong,

Graham Neubig, and Yusuke Oda, editors, *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, NMT@ACL 2018, Melbourne, Australia, July 20, 2018*, pages 45–54. Association for Computational Linguistics, 2018. doi: 10.18653/v1/w18-2706. URL https://doi.org/10.18653/v1/w18-2706. 23, 64

[110] Li Deng, Gokhan Tur, Xiaodong He, and Dilek Hakkani-Tur. Use of kernel deep convex networks and end-to-end learning for spoken language understanding. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 210–215. IEEE, 2012. 23

[111] Gokhan Tur, Li Deng, Dilek Hakkani-Tür, and Xiaodong He. Towards deeper understanding: Deep convex networks for semantic utterance classification. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5045–5048. IEEE, 2012. 23

[112] D Yann, G Tur, D Hakkani-Tur, and L Heck. Zero-shot learning and clustering for semantic utterance classification using deep learning. In *International Conference on Learning Representations (cited on page 28)*, 2014. 23

[113] Ruhi Sarikaya, Geoffrey E Hinton, and Anoop Deoras. Application of deep belief networks for natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):778–784, 2014. 23

[114] Suman Ravuri and Andreas Stolcke. Recurrent neural network and lstm models for lexical utterance classification. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015. 23

[115] Suman V. Ravuri and Andreas Stolcke. A comparative study of recurrent neural network models for lexical domain classification. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, pages 6075–6079. IEEE, 2016. doi: 10.1109/ICASSP.2016.7472844. URL https://doi.org/10.1109/ICASSP.2016.7472844. 23

[116] Homa B Hashemi, Amir Asiaee, and Reiner Kraft. Query intent detection using convolutional neural networks. In *International Conference on Web Search and Data Mining, Workshop on Query Understanding*, 2016. 23

[117] Ji Young Lee and Franck Dernoncourt. Sequential short-text classification with recurrent and convolutional neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 515–520, San Diego, California, 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1062. URL https://aclanthology.org/N16-1062. 23

[118] Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. Todbert: Pre-trained natural language understanding for task-oriented dialogues. *ArXiv preprint*, abs/2004.06871, 2020. URL https://arxiv.org/abs/2004.06871. 23

[119] Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.nlp4convai-1.5. URL `https://aclanthology.org/2020.nlp4convai-1.5`. 24

[120] Waheed Ahmed Abro, Annalena Aicher, Niklas Rach, Stefan Ultes, Wolfgang Minker, and Guilin Qi. Natural language understanding for argumentative dialogue systems in the opinion building domain. *Knowledge-Based Systems*, 242:108318, 2022. 24

[121] Aleksander Obuchowski and Michal Lew. Transformer-capsule model for intent detection (student abstract). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 13885–13886. AAAI Press, 2020. URL `https://aaai.org/ojs/index.php/AAAI/article/view/7215`. 24

[122] Ruhi Sarikaya, Geoffrey E Hinton, and Bhuvana Ramabhadran. Deep belief nets for natural language call-routing. In *2011 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pages 5680–5683. IEEE, 2011. 24

[123] Anoop Deoras and Ruhi Sarikaya. Deep belief network based semantic taggers for spoken language understanding. In *Interspeech*, pages 2713–2717, 2013. 24

[124] Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. Recurrent neural networks for language understanding. In *Interspeech*, pages 2524–2528, 2013. 24

[125] Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *Interspeech*, pages 3771–3775, 2013. 24

[126] Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539, 2014. 24

[127] Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. Spoken language understanding using long short-term memory neural networks. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 189–194. IEEE, 2014. 24

[128] Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. Recursive template-based frame generation for task oriented dialog. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2059–2064, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.186. URL https://aclanthology.org/2020.acl-main.186. 24

[129] Samuel Coope, Tyler Farghly, Daniela Gerz, Ivan Vulić, and Matthew Henderson. Span-ConveRT: Few-shot span extraction for dialog with pretrained conversational representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 107–121, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.11. URL https://aclanthology.org/2020.acl-main.11. 24

[130] Seyedmahdad Mirsamadi, Emad Barsoum, and Cha Zhang. Automatic speech emotion recognition using recurrent neural networks with local attention. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, pages 2227–2231. IEEE, 2017. doi: 10.1109/ICASSP.2017.7952552. URL https://doi.org/10.1109/ICASSP.2017.7952552. 25

[131] Yuanchao Li, Tianyu Zhao, and Tatsuya Kawahara. Improved end-to-end speech emotion recognition using self attention mechanism and multitask learning. In Gernot Kubin and Zdravko Kacic, editors, *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 2803–2807. ISCA, 2019. doi: 10.21437/Interspeech.2019-2594. URL https://doi.org/10.21437/Interspeech.2019-2594. 25

[132] Promod Yenigalla, Abhay Kumar, Suraj Tripathi, Chirag Singh, Sibsambhu Kar, and Jithendra Vepa. Speech emotion recognition using spectrogram & phoneme embedding. In B. Yegnanarayana, editor, *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, pages 3688–3692. ISCA, 2018. doi: 10.21437/Interspeech.2018-1811. URL https://doi.org/10.21437/Interspeech.2018-1811. 25

[133] Yue Xie, Ruiyu Liang, Zhenlin Liang, Chengwei Huang, Cairong Zou, and Björn W. Schuller. Speech emotion classification using attention-based LSTM. *IEEE ACM Trans. Audio Speech Lang. Process.*, 27(11):1675–1685, 2019. doi: 10.1109/TASLP.2019.2925934. URL https://doi.org/10.1109/TASLP.2019.2925934. 25

[134] Ernest Davis and Gary Marcus. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103, 2015. 26

[135] Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. Knowledge diffusion for neural dialogue generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1498, 2018. 26, 80

[136] Prasanna Parthasarathi and Joelle Pineau. Extending neural generative conversational model using external knowledge sources. *arXiv preprint arXiv:1809.05524*, 2018. 80

[137] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. Towards knowledge-based recommender dialog system. *arXiv preprint arXiv:1908.05391*, 2019. 26

[138] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629, 2018. 26

[139] Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. Conversation generation with concept flow. 2019. 26

[140] Zhibin Liu, Zheng-Yu Niu, Hua Wu, and Haifeng Wang. Knowledge aware conversation generation with explainable reasoning over augmented graphs. *arXiv preprint arXiv:1903.10245*, 2019. 26, 80

[141] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 27, 31, 32

[142] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86 (11):2278–2324, 1998. 28

[143] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012. URL `https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html`. 29

[144] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[145] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1409.1556`.

[146] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1–9. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7298594. URL `https://doi.org/10.1109/CVPR.2015.7298594`.

[147] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL `https://doi.org/10.1109/CVPR.2016.90`.

[148] Neena Aloysius and M Geetha. A review on deep convolutional neural networks. In *2017 International Conference on Communication and Signal Processing (ICCSP)*, pages 0588–0592. IEEE, 2017.

[149] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9): 2352–2449, 2017. 29

[150] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1107–1116, Valencia, Spain, 2017. Association for Computational Linguistics. URL `https://aclanthology.org/E17-1104`. 29

[151] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *ArXiv preprint*, abs/1506.00019, 2015. URL `https://arxiv.org/abs/1506.00019`. 30

[152] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13 (2):260–269, 1967. 30

[153] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982. 30

[154] MI Jordan. Serial order: a parallel distributed processing approach. technical report, june 1985-march 1986. Technical report, California Univ., San Diego, La Jolla (USA). Inst. for Cognitive Science, 1986. 30

[155] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. 30

[156] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994. 31

[157] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001. 31

[158] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989. 32

[159] Razvan Pascanu, Tomás Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 1310–1318. JMLR.org, 2013. URL `http://proceedings.mlr.press/v28/pascanu13.html`. 32

[160] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL `https://aclanthology.org/D14-1179`. 33

[161] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 33

[162] Nicole Gruber and Alfred Jockisch. Are gru cells more specific and lstm cells more sensitive in motive classification of text? *Frontiers in Artificial Intelligence*, 3(40):1–6, 2020. 33

[163] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997. 34

[164] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. 34, 36, 44

[165] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In James Bailey, Alistair Moffat, Charu C. Aggarwal, Maarten de Rijke, Ravi Kumar, Vanessa Murdock, Timos K. Sellis, and Jeffrey Xu Yu, editors, *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 553–562. ACM, 2015. doi: 10.1145/2806416.2806493. URL `https://doi.org/10.1145/2806416.2806493`. 35

[166] Alan Ritter, Colin Cherry, and William B. Dolan. Data-driven response generation in social media. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 583–593, Edinburgh, Scotland, UK., 2011. Association for Computational Linguistics. URL https://aclanthology.org/D11-1054. 35

[167] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205, Denver, Colorado, 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1020. URL https://aclanthology.org/N15-1020. 35

[168] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3295–3301. AAAI Press, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14567. 36

[169] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1410.3916. 37

[170] Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas, 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1244. URL https://aclanthology.org/D16-1244. 40

[171] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR, 2017. URL http://proceedings.mlr.press/v70/gehring17a.html. 40

[172] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf. 43, 65

[173] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`. 43

[174] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/c8512d142a2d849725f31a9a7a361ab9-Abstract.html`. 43

[175] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1285. URL `https://aclanthology.org/P19-1285`.

[176] Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. Star-transformer. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1315–1325, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1133. URL `https://aclanthology.org/N19-1133`. 43

[177] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *ArXiv preprint*, abs/2106.04554, 2021. URL `https://arxiv.org/abs/2106.04554`. 43

[178] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ArXiv preprint*, abs/2009.06732, 2020. URL `https://arxiv.org/abs/2009.06732`. 43

[179] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014. 44, 47

[180] Karol Kurach, Marcin Andrychowicz, and Ilya Sutskever. Neural random-access machines. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL `http://arxiv.org/abs/1511.06392`. 47

[181] Jinjie Ni, Rui Mao, Zonglin Yang, Han Lei, and Erik Cambria. Finding the pillars of strength for multi-head attention. *arXiv preprint arXiv:2305.14380*, 2023. 49

[182] Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Hervé Jégou, and Armand Joulin. Augmenting self-attention with persistent memory. *CoRR*, abs/1907.01470, 2019. URL `http://arxiv.org/abs/1907.01470`. 49, 52, 97

[183] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=HJlnC1rKPB`. 49

[184] Hanchuan Peng, Fuhui Long, and Chris H. Q. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1226–1238, 2005. doi: 10.1109/TPAMI.2005.159. URL `https://doi.org/10.1109/TPAMI.2005.159`. 50

[185] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=rJl-b3RcF7`. 51, 55, 97

[186] Karim Ahmed, Nitish Shirish Keskar, and Richard Socher. Weighted transformer network for machine translation. *CoRR*, abs/1711.02132, 2017. URL `http://arxiv.org/abs/1711.02132`. 52

[187] Jian Li, Baosong Yang, Zi-Yi Dou, Xing Wang, Michael R. Lyu, and Zhaopeng Tu. Information aggregation for multi-head attention with routing-by-agreement. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3566–3575. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1359. URL `https://doi.org/10.18653/v1/n19-1359`. 52

[188] Hongyi Cui, Shohei Iida, Po-Hsuan Hung, Takehito Utsuro, and Masaaki Nagata. Mixed multi-head self-attention for neural machine translation. In Alexandra Birch, Andrew M. Finch, Hiroaki Hayashi, Ioannis Konstas, Thang Luong, Graham Neubig, Yusuke Oda, and Katsuhito Sudoh, editors, *Proceedings of the 3rd Workshop on Neural Generation and Translation@EMNLP-IJCNLP 2019, Hong Kong, November 4, 2019*, pages 206–214. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-5622. URL `https://doi.org/10.18653/v1/D19-5622`. 52

[189] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987. 61

[190] James C Bezdek and Nikhil R Pal. Cluster validation with generalized dunn's indices. In *Proceedings 1995 second New Zealand international two-stream conference on artificial neural networks and expert systems*, pages 190–190. IEEE Computer Society, 1995. 61

[191] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL `https://openreview.net/forum?id=uYLFoz1vlAC`. 65

[192] Chenguang Wang, Mu Li, and Alexander J. Smola. Language models with transformers. *CoRR*, abs/1904.09408, 2019. URL `http://arxiv.org/abs/1904.09408`. 65

[193] Sriram Vajapeyam. Understanding shannon's entropy metric for information. *CoRR*, abs/1405.2061, 2014. URL `http://arxiv.org/abs/1405.2061`. 65

[194] Jinjie Ni, Yukun Ma, Wen Wang, Qian Chen, Dianwen Ng, Han Lei, Trung Hieu Nguyen, Chong Zhang, Bin Ma, and Erik Cambria. Adaptive knowledge distillation between text and speech pre-trained models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 67

[195] Zhenzhong Lan et al. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 67

[196] Alec et al. Radford. Language models are unsupervised multitask learners. *OpenAI blog*, 2019. 67

[197] Steffen Schneider et al. wav2vec: Unsupervised pre-training for speech recognition. In Gernot Kubin and Zdravko Kacic, editors, *Interspeech 2019, Graz, Austria, 15-19 September 2019*, pages 3465–3469. ISCA, 2019. doi: 10.21437/Interspeech.2019-1873. 67

[198] Alexei Baevski et al. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 73

[199] Wei-Ning Hsu et al. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:3451–3460, 2021. doi: 10.1109/TASLP.2021.3122291. URL `https://doi.org/10.1109/TASLP.2021.3122291`. 67

[200] Shu-Wen Yang et al. SUPERB: speech processing universal performance benchmark. In Hynek Hermansky, Honza Cernocký, Lukás Burget, Lori Lamel, Odette Scharenborg, and Petr Motlícek, editors, *Interspeech 2021, Brno, Czechia, 30 August - 3 September 2021*, pages 1194–1198. ISCA, 2021. doi: 10.21437/Interspeech.2021-1775. URL https://doi.org/10.21437/Interspeech.2021-1775. 67, 72

[201] Junyi Ao et al. Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5723–5738. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.393. URL https://doi.org/10.18653/v1/2022.acl-long.393. 68

[202] Sachidananda et al. Calm: Contrastive aligned audio-language multirate and multimodal representations. *arXiv preprint arXiv:2202.03587*, 2022. 68

[203] Ankur et al. Bapna. Slam: A unified encoder for speech and language modeling via speech-text joint pre-training. *arXiv preprint arXiv:2110.10329*, 2021. 68

[204] Ankur et al. Bapna. mslam: Massively multilingual joint pre-training for speech and text. *arXiv preprint arXiv:2202.01374*, 2022.

[205] Bhuvan et al. Agrawal. Tie your embeddings down: Cross-modal latent spaces for end-to-end spoken language understanding. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7157–7161. IEEE, 2022.

[206] Alexander et al. Liu. Cross-modal discrete representation learning. *arXiv preprint arXiv:2106.05438*, 2021. 68

[207] Pavel Denisov and Ngoc Thang Vu. Pretrained semantic speech embeddings for end-to-end spoken language understanding via cross-modal teacher-student learning. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 881–885. ISCA, 2020. doi: 10.21437/Interspeech.2020-2456. URL https://doi.org/10.21437/Interspeech.2020-2456. 68

[208] Liunian Harold Li et al. What does BERT with vision look at? In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5265–5275. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.469. URL https://doi.org/10.18653/v1/2020.acl-main.469. 72

[209] Vassil Panayotov et al. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pages 5206–5210. IEEE, 2015. doi:

10.1109/ICASSP.2015.7178964. URL `https://doi.org/10.1109/ICASSP.2015.7178964`. 72

[210] Yukun Ma, Khanh Linh Nguyen, Frank Xing, and Erik Cambria. A survey on empathetic dialogue systems. *Information Fusion*, 64:50–70, 2020. 77

[211] Tom Young, Frank Xing, Vlad Pandelea, Jinjie Ni, and Erik Cambria. Fusing task-oriented and open-domain dialogues in conversational agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11622–11629, 2022.

[212] Geng Tu, Jintao Wen, Cheng Liu, Dazhi Jiang, and Erik Cambria. Context- and sentiment-aware networks for emotion recognition in conversation. *IEEE Trans. Artif. Intell.*, 3(5):699–708, 2022. doi: 10.1109/TAI.2022.3149234. URL `https://doi.org/10.1109/TAI.2022.3149234`.

[213] Jintao Wen, Dazhi Jiang, Geng Tu, Cheng Liu, and Erik Cambria. Dynamic interactive multiview memory network for emotion recognition in conversation. *Inf. Fusion*, 91:123–133, 2023. doi: 10.1016/j.inffus.2022.10.009. URL `https://doi.org/10.1016/j.inffus.2022.10.009`.

[214] Dazhi Jiang, Runguo Wei, Jintao Wen, Geng Tu, and Erik Cambria. Automl-emo: Automatic knowledge selection using congruent effect for emotion identification in conversations. *IEEE Transactions on Affective Computing*, 2022.

[215] Wei Li, Yang Li, Vlad Pandelea, Mengshi Ge, Luyao Zhu, and Erik Cambria. Ecpec: emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 2022.

[216] Wei Li, Luyao Zhu, Rui Mao, and Erik Cambria. Skier: A symbolic knowledge integrated model for conversational emotion recognition. *arxiv*, 2023. 77

[217] Liang Qiu, Yizhou Zhao, Weiyan Shi, Yuan Liang, Feng Shi, Tao Yuan, Zhou Yu, and Song-Chun Zhu. Structured attention for unsupervised dialogue structure induction. *arXiv preprint arXiv:2009.08552*, 2020. 77

[218] Cao Liu, Shizhu He, Kang Liu, and Jun Zhao. Vocabulary pyramid network: Multi-pass encoding and decoding with multi-level vocabularies for response generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3774–3783, 2019. 77

[219] Hui Su, Xiaoyu Shen, Sanqiang Zhao, Xiao Zhou, Pengwei Hu, Randy Zhong, Cheng Niu, and Jie Zhou. Diversifying dialogue generation with non-conversational text. *arXiv preprint arXiv:2005.04346*, 2020. 77

[220] Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. Personalizing dialogue agents via meta-learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5459, 2019. 77

[221] Zhenqiao Song, Xiaoqing Zheng, Lu Liu, Mu Xu, and Xuan-Jing Huang. Generating responses with a specific emotion in dialog. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3685–3695, 2019. 77

[222] Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. Proactive human-machine conversation with explicit conversation goals. *arXiv preprint arXiv:1906.05572*, 2019. 77, 80

[223] Xiuyi Chen, Fandong Meng, Peng Li, Feilong Chen, Shuang Xu, Bo Xu, and Jie Zhou. Bridging the gap between prior and posterior knowledge selection for knowledge-grounded dialogue generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3426–3437, 2020. 77

[224] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. A survey on knowledge graphs: Representation, acquisition and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10), 2022. 78

[225] Tom Young, Vlad Pandelea, Soujanya Poria, and Erik Cambria. Dialogue systems with audio context. *Neurocomputing*, 388:102–109, 2020. 80

[226] H Xu, Haiyun Peng, H Xie, Erik Cambria, L Zhou, and W Zheng. End-to-end latent-variable task-oriented dialogue system with exact log-likelihood optimization. *World Wide Web*, 23:1989–2002, 2020. 80

[227] Jun Xu, Haifeng Wang, Zhengyu Niu, Hua Wu, and Wanxiang Che. Knowledge graph grounded goal planning for open-domain conversation generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9338–9345, 2020. 80

[228] Jianheng Tang, Tiancheng Zhao, Chenyan Xiong, Xiaodan Liang, Eric P Xing, and Zhiting Hu. Target-guided open-domain conversation. *arXiv preprint arXiv:1905.11553*, 2019. 80

[229] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013. 83

[230] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019. 83

[231] Yu-An Wang and Yun-Nung Chen. What do position embeddings learn? an empirical study of pre-trained language model positional encoding. *arXiv preprint arXiv:2010.04903*, 2020. 83

[232] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 83

[233] Jun Quan and Deyi Xiong. Modeling long context for task-oriented dialogue state generation. *arXiv preprint arXiv:2004.14080*, 2020. 83

[234] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016. 87

[235] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 88

[236] OpenAI. ChatGPT: Optimizing Language Models for Dialogue. `https://openai.com/blog/chatgpt/`, 2022. 94

[237] Haixu Wu, Jialong Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Flowformer: Linearizing transformers with conservation flows. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 24226–24242. PMLR, 2022. URL `https://proceedings.mlr.press/v162/wu22m.html`. 97

[238] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.308. URL `https://doi.org/10.1109/CVPR.2016.308`. 101

[239] Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing neural networks by penalizing confident output distributions. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=HyhbYrGYe`. 101

[240] Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 1139–1147. JMLR.org, 2013. URL `http://proceedings.mlr.press/v28/sutskever13.html`. 101

[241] Razvan Pascanu, Tomás Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 1310–1318. JMLR.org, 2013. URL http://proceedings.mlr.press/v28/pascanu13.html. 101

[242] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Skq89Scxx. 101

[243] Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc V. Le. Massive exploration of neural machine translation architectures. *CoRR*, abs/1703.03906, 2017. URL http://arxiv.org/abs/1703.03906. 103

[244] Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. Classical structured prediction losses for sequence to sequence learning. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 355–364. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-1033. URL https://doi.org/10.18653/v1/n18-1033. 103

[245] Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/afdec7005cc9f14302cd0474fd0f3c96-Abstract.html. 103

[246] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013. 103