Explaining Natural Language Inference with Factual and Template Memory Networks

by

Zi Xuan Zhang

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science
University of Alberta

Abstract

In the era of artificial intelligence, neural models have emerged as a powerful tool for tackling a wide range of tasks. However, these models are commonly regarded as black-box systems, making it difficult to understand their internal workings. The natural language explanation task seeks to elucidate the decisions of a black-box system by generating human-understandable explanations. The task is important for natural language understanding systems in many domains such as in the medical and legal domains. While numerous existing studies are capable of performing the task, they rely on training in an end-to-end fashion, which still limits them to being black-box machinery.

In this work, we focus on the natural language explanation task for natural language inference. The task aims to explain the relationship between two sentences with text, namely in the tone of entailment, contradiction, or neutral. We propose a memory network that utilizes factual knowledge given by weakly supervised reasoning and template knowledge extracted by rules and heuristics. Experiments show that our approach achieves state-of-the-art performance on the e-SNLI dataset. Our analyses further verify the roles of both factual and template memories.

Preface

Part of this thesis has been published as Zijun Wu, Zi Xuan Zhang, Atharva Naik, Zhijian Mei, Mauajama Firdaus, and Lili Mou, "Weakly Supervised Explainable Phrasal Reasoning with Neural Fuzzy Logic," in Proceedings of the International Conference on Learning Representations, 2023.

My contribution includes the application of predicted phrasal logical relationships as factual knowledge to enhance the generation of textual explanations and designing part of the algorithm for phrase alignment.

"I don't have dreams, I have goals. Now on to the next one."

- Suits

Acknowledgements

I would like to express my profound gratitude to Dr. Lili Mou, whose unwavering support, guidance, and encouragement have been invaluable throughout my journey. Over the past two and a half years, Dr. Mou's contribution to my growth and development has been immeasurable. He has been a constant pillar of strength, always there to steer me in the right direction whenever I felt lost.

Under Dr. Mou's exceptional mentorship, I have sensed a significant improvement in various aspects of my academic and personal life. My scientific thinking, logical reasoning, writing, reading, presentation, and overall analytical skills have all seen remarkable advancements during this time. His dedication to nurturing my abilities has had a profound impact on shaping my academic prowess and broader understanding of the world around me.

I also want to appreciate the accompany of Zijun Wu and Dongheng Li during my academic career. Throughout the years in my undergraduate and graduate life, we spent countless hours discussing topics on ideas, philosophies, and works around research. They provided valuable insights into the field of NLP, meanwhile fueling my passion for my work.

I wish to express my gratefulness towards my labmates Chenyang Huang, Puyuan Liu, Yongchang Hao, Yuqiao Wen, and Guoqing Luo. I find their captivating presentations truly enjoyable. Their willingness to openly share their work has greatly contributed to my better understanding of the research community as a whole.

I could have not done this work without Zijun Wu, Zhiqian Mei, Mauzama Firdaus, and Dr. Lili Mou. I am thankful for their contribution. Zijun Wu brainstormed

many of my small idea proposals with me, Zhiqian helped with running different experimental settings that helped stabilize our experiments, Mauzama supported the writing of the paper, and Lili educated me throughout the project with experimental settings, paper writings, and new ideas. I would like to extend my recognition and gratitude to Dr. Shailza Jolly, who guided me through a mature project that she authored. Without the experience gained from that project, I wouldn't have been able to accomplish this one.

My deepest appreciation goes to my parent. I am unable to express the amount of gratitude for their unconditional love and support with language. They are always there unreservedly supporting me, giving me the most comfort and reassurance to propel me in my desired ways.

The research is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant No. RGPIN2020-04465, the Amii Fellow Program, the Canada CIFAR AI Chair Program, a UAHJIC project, a donation from DeepMind, and the Digital Research Alliance of Canada (alliancecan.ca).

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The advent of the internet has led to an unprecedented overgrowth of data accessibility for the mass public. The abundance of data has opened many eyes to developing artificial intelligence (AI) systems that can meaningfully utilize the data for tasks such as image classification, image generation, text classification, and speech recognition.

Natural language processing (NLP) is one of the most prominent branches of AI. It combines computer science and linguistics to enable computers to communicate with human beings. Common NLP tasks include machine translation [9, 33], dialogue generation [28, 39], paraphrase generation [27, 45], and question-answering [59, 79]. Natural language processing has two primary fields: natural language understanding and natural language generation. The former focuses on understanding languages and the latter on generating texts.

Deep learning is a machine learning technique that has been used for building powerful models that have the ability to learn from mass data. NLP systems that utilize deep learning have been consistently improving on various NLP benchmarks [29, 72] over the years. While achieving high performance, deep neural models are generally regarded as black-box machinery, that is, the model outputs its decision without explaining or showing insights into its decision-making process. This is troublesome for

SNLI Input (highlights are additionally annotated rationales in e-SNLI
Premise *A woman is running a marathon in a park.* Hypothesis *The woman is running in her backyard.*
SNLI Label
○ Entailment ○ Neutral ● Contradiction
e-SNLI Explanation Reference
A woman cannot run in her backyard and at the park simultaneously.

Figure 1.1: The explanation-augmented Stanford Natural Language Inference (e-SNLI) task example.

certain applications that require explicit textual explanations in decision-making, for example, in medical and legal domains [7, 24].

In this thesis, we focus on the task of explanation-augmented Stanford Natural Language Inference (e-SNLI, Camburu *et al.*, 2018). SNLI aims to determine the logical relationship—namely, `Entailment`, `Contradiction`, and `Neutral`, abbreviated as `E`, `C`, and `N`—between two pieces of text [10]. The e-SNLI dataset extends SNLI with a textual explanation for each sample where the task is to output an explanation for the logical relationship of two pieces of text. In Figure 1.1, the example shows a premise and hypothesis that has the SNLI label of `Contradiction` followed with an e-SNLI explanation reference of "A woman cannot run in her backyard and at the park simultaneously". Additionally, e-SNLI annotators highlight *park* in the premise and *backyard* in the hypothesis to support the rationale of the explanation reference. The highlight information may seem helpful, but is extraneous for the task, and hurts the explanation generation performance. Evidence can be found in Table 4.1, where LiREx [81] tries to utilize this information but achieves worse performance when compared to the NILE approach, which uses the same base model.

1.2 Explanability in Explanation Generation

Neural models are commonly regarded as black-box systems, making it difficult to understand their decision-making process. In previous work, pretrained language models are finetuned for explanation generation in addition to SNLI classification by multi-task learning. Such approaches are trained in an end-to-end fashion, and unfortunately, are still black-box machinery. Zhao and Vydiswaran [81] propose to incorporate human-annotated rationales as highlighted in Figure 1.1 for explanation generation, but this requires extensive efforts of human labeling. Narang *et al.* [51] trains a multi-task model by providing the groundtruth NLI label.

In light of these limitations, we believe it is necessary to ensure explicit explainability for the model's output, albeit it is performing the explanation generation task. Previous studies have developed methods to interpret a model's output [2, 62] in a post-hoc manner. While these methods provide insights into the latent reasoning process for a model's response, they heavily rely on rules and human-derived scoring functions, which in turn sacrifices the robustness. For instance in Figure 1.1, the explanation *"A woman cannot run in her backyard and at the park simultaneously."* can be generated by models that are fine-tuned on e-SNLI. However, previous work cannot show the model's decision-making process that led to this output. Our intuition is that the phrase "$\cdots$ *cannot* $\cdots$ *simultaneously*" would be a template knowledge that has appeared in the training data, while "*A woman* $\cdots$ *run in her backyard and at the park* $\cdots$" represents the factual knowledge extracted from the sample's premise and hypothesis. Therefore, we incorporate these types of knowledge into the model, enabling it to utilize this latent reasoning information. Additionally, we can interpret the model's output by examining what information it has utilized. In this way, we are able to provide explicit explanations of the output for the explanation generation task.

1.3 Thesis Contribution

In this thesis, we propose to address the explainability issue in explanation generation. Our objective is to utilize the attention mechanism to enhance a system's explainability, enabling humans to comprehend the underlying reason behind the model's output by examining its decision-making process in a direct manner.

Specifically, we explicitly model the rationales for explanation generation. We adopt a fuzzy logic reasoning model [76] that yields a set of phrases and their logical relations in a weakly supervised manner to obtain factual knowledge in the form of tuples. In addition, we devise a simple yet effective rule-based approach to extract template knowledge. We propose memory networks to feed these knowledge into our explanation generator during training and inference. In details, we treat each of the factual tuples as memory slots, and perform attention on them. Likewise, the templates are also treated as a separate memory pad, where we perform another attention, and fuse the information with the model. We design a decoder to that integrates the memories for the generation process by introducing intermediate layers to match the different distributions. In this way, such explicit modeling of knowledge helps our approach to be more explainable. In contrast to Zhao and Vydiswaran [81] and Narang *et al.* [51], we do not require additional human annotations or groundtruth labels.

To evaluate our approach, we compare it with previous methods on e-SNLI. Since the evaluation metric was not consistently used in previous work, we unify them by using two settings on two metrics. Camburu *et al.* [11] reported inconsistency with the two settings. However, our experimental results show we outperform previous state-of-the-art models in terms of both BLEU and SacreBLEU scores. Furthermore, the results show our approach is also more explainable, aligning with our claim.

In summary, our thesis contributions include:

- We propose a method to explicitly model the rationales for explanation generation that does not require additional human annotation.
- We design a memory network to feed such knowledge to a explanation generator.
- We evaluate our approach on the e-SNLI [11] dataset. Our approach outperforms previous state-of-the-art models on all the metrics used in previous work.

1.4 Thesis Organization

In this chapter, we introduced the background of explanation generation for natural language inference, and stated our motivation and contribution.

Chapter 2 delves into the extensive body of related previous work in the area of explanation generation. We explore the literature on reasoning in NLP, examining different approaches and techniques used to enhance the interpretability and explainability of natural language models. Additionally, we go into the domain of memory networks, which serve as a key component in our proposed approach. By presenting a comprehensive review of prior work, we establish the context for our research and highlight its novel aspects.

Chapter 3 introduces our proposed approach for explanation generation. We describe each component of our method in detail, specifically the design of our memory network components. We present the adaptation of our design into an existing architecture and our training method and inference method.

Chapter 4 is dedicated to the comprehensive discussion of our evaluation and analyses. We start by describing our experimental setup, including the datasets used and evaluation metrics employed. Following up, we present our experimental results, highlighting the performance and effectiveness of our proposed approach. To provide in-depth insights, we further conduct additional quantitative experiments and present case studies that further verify our approach.

Chapter 5 concludes the thesis by summarizing the key findings and contributions of our research. We examine the limitations and discuss future work.

Chapter 2

Background & Related Work

2.1 Natural Language Generation

Natural language generation (NLG) refers to tasks generating natural language text from different forms of input information ranging from structured tables [37, 54] to plain texts [27, 39]. The goals of NLG systems include generating coherent, context-specific, grammatically correct, and any other quality aspects of natural language. For example, style transfer [36, 47, 70] is an NLG task with the objective of changing the tone or characteristics of the given text while preserving its context. An instance of this task is to transfer a piece of text to the Shakespearean form: from "You are a star in the night sky" to "Thou art a star in the night sky".

There are various forms of input for NLG. The task of speech recognition aims to convert spoken language into written text where the input information is usually a recording file composed of digital sound data [6, 26]. The image captioning task involves generating a textual description for an image, with the image as the input. The table-to-text task focuses on translating the data within a table into plain text, taking table information as the input [3, 30]. Even though inputs may vary in format, NLG models can effectively transform them into a representation suitable for its internal processing [37, 54].

Currently, encoder-decoder models are the prevailing approach that has been commonly applied to NLG tasks, outperforming many other model architectures [4, 8,

69]. The encoder translates the input into a vector representation of a predefined dimension, capturing the contextual information of the given sequence. Subsequently, the decoder takes the contextual representations and decodes them by generating words one after another.

Sequence-to-sequence (Seq2Seq) models are a specific type of the encoder-decoder framework. It is designed to handle input and output sequences of variable lengths. The pioneering work [68] utilizes recurrent neural networks (RNNs) to model the discrete nature of text. Based on this model, later work [21] develop the attention mechanisms to model the soft dependencies between contextual encodings [15, 21].

The transformer model [71] revolutionizes the use of attention mechanisms in neural models. Unlike traditional Seq2Seq designs, the transformer solely relies on a combination of attention mechanisms, eliminating the need for recurrence. The model's capability of capturing dependencies and handling varying sequence lengths advances the field of NLG, achieving state-of-the-art results in many tasks.

In the following sections, we will discuss the advantages and disadvantages of different neural models for NLP, starting with the foundational model: RNN.

2.1.1 Recurrent Neural Network

The recurrent neural network (RNN) is derived from the traditional feed-forward neural network (FFNN) where it introduces the concept of hidden states, allowing the architecture to better model discrete and sequential data [31, 63]. The calculation of RNN proceeds in terms of time steps.

Let $\{\boldsymbol{x}_t\}_{t=1}^{T}$ be all the inputs from time step 1 to T where $\boldsymbol{x}_t \in \mathbb{R}^d$ is the vector representation of some sample at time step t. The hidden state of each time step $\boldsymbol{h}_t$ of a vanilla RNN is passed on from the previous time step

$$\begin{aligned} \boldsymbol{h}_t &= \mathrm{RNN}(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}) \\ &= f(\mathbf{W}_x \boldsymbol{x}_t + \mathbf{W}_h \boldsymbol{h}_{t-1} + \boldsymbol{b}_h) \end{aligned} \tag{2.1}$$

where f is an activation function, $\mathbf{W}_x \in \mathbb{R}^{h \times d}$ is the weight matrix for the input, $\mathbf{W}_h \in \mathbb{R}^{h \times h}$ is the weight matrix for the hidden state, and $\boldsymbol{b}_h$ is the bias.

The RNN architecture effectively models the dependencies of sequences. However, it lacks the finer-grained design of aligning words between sequences. This leads to the birth of RNN-based Seq2Seq with attention model [21], where it features an attention mechanism to model the alignment of words between sequences. Given a sequence $\{\boldsymbol{x}_t\}_{t=1}^{T}$ and its target $\{\boldsymbol{y}_{t'}\}_{t'=1}^{T'}$, the goal of the Seq2Seq model is to estimate the conditional probability $p(y_1, \cdots, y_{T'} | x_1, \cdots, x_T)$. The model features an encoder and a decoder. The encoder encodes all the input in $\{\boldsymbol{x}_t\}_{t=1}^{T}$ to obtain a contextual state of $\boldsymbol{h}_T$ with Equation 2.1. Subsequently, the decoder decodes the context by

$$\boldsymbol{y}'_{t'} = \text{softmax}(\boldsymbol{y}'_{t'-1}, \boldsymbol{h}_{t-1}; \mathbf{W}_{dec}) \tag{2.2}$$

where $\boldsymbol{y}'_{t'}$ is the current decoded probability distribution over the vocabulary, $\boldsymbol{y}'_{t-1}$ is the vector representation of the previous decoded output, $\boldsymbol{h}_{t'-1}$ is the previous time step hidden state, and $\mathbf{W}_{dec}$ is the set of weight matrices of the decoder. For the first time step $t' = 1$, the hidden state $\boldsymbol{h}_{t'-1}$ is set to $\boldsymbol{h}_T$ and $\boldsymbol{y}'_0$ is set to be the representation of a special end of string token e.g. `[EOS]` [68].

It is intuitive that as the time step advances, the hidden state $\boldsymbol{h}_t$ gradually gets overwritten by more recent information, leading to the loss of earlier input context. Consequently, the decoder's initial step suffers from a limited understanding of the input due to its reliance on the final context $\boldsymbol{h}_T$. This is known as the Seq2Seq's information bottleneck problem. As RNN models sequential data by training with backpropagation through time [66], it also suffers from the vanishing and exploding gradient problem [18, 55]. When calculating the gradient using the chain rule at each time step, the gradient can either amplify significantly or diminish to nearly zero as it propagates. This characteristic hinders an RNN's ability to effectively learn long sequences. The introduction of Long Short-Term Memory (LSTM) [31] was designed to solve this problem. However, the inability lies within the RNN's way of sequential

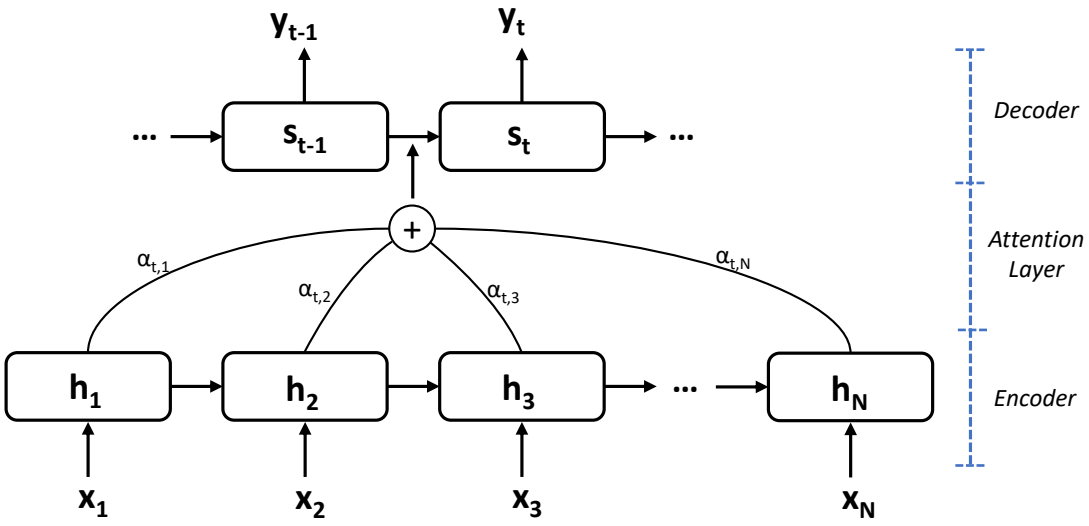

Figure 2.1: RNN with an attention mechanism.

modeling itself. In other words, naively passing the information from the previous time step to the next is insufficient for solving the task of long sequence modeling [74].

2.1.2 Sequence-to-sequence with Attention

In an effort to alleviate the sequence length constraint, Dzmitry *et al.* [21] propose a novel approach of jointly learning the alignment between input and output sequences, as illustrated in Figure 2.1. Their inspiration derives from the machine translation task, where the target language exhibits a direct correspondence with a specific section of the input language. Consequently, the need arises to effectively model the alignment between the sections of a given sample and its corresponding target. While this intuition may appear straightforward, conventional Seq2Seq approaches had previously only accounted for coarser alignments, where one full sequence corresponded to another full sequence. It is with the integration of the attention mechanism that the modeling reaches a more refined and granular level.

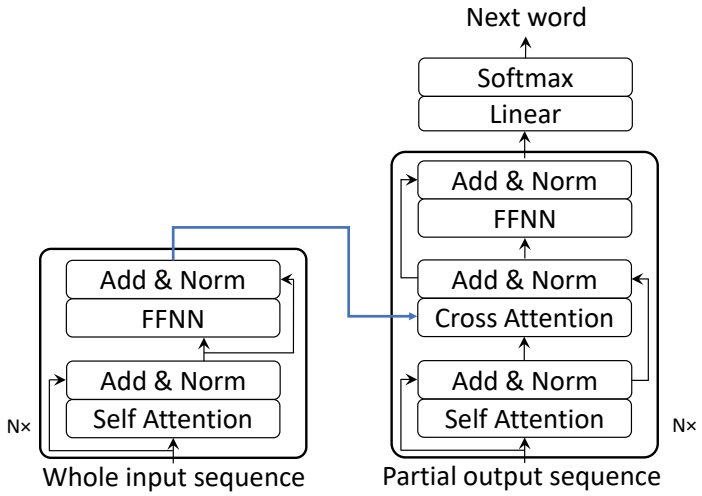

Figure 2.2: The Transformer architecture.

Formally, attention calculates a context vector $\boldsymbol{c}_t$ at each time step t, which is dependent on all the hidden states of the input sentence $(\boldsymbol{h}_1, \cdots, \boldsymbol{h}_K)$ in the encoder

$$\boldsymbol{c}_t = \sum_{k=1}^{K} \alpha_{tk} \boldsymbol{h}_k \tag{2.3}$$

$$\alpha_{tk} = \frac{\exp(e_{tk})}{\sum_{j=1}^{J} \exp(e_{tj})} \tag{2.4}$$

$$e_{tk} = a(\boldsymbol{s}_{t-1}, \boldsymbol{h}_k; \mathbf{W}_a) \tag{2.5}$$

where α_{tk} is the attention weight of each hidden state $\boldsymbol{h}_k$, $\boldsymbol{s}_{t-1}$ a hidden state of the decoder computed with an activation function q as $\boldsymbol{s}_t = q(\boldsymbol{s}_{t-1}, \boldsymbol{y}_{t-1}, \boldsymbol{c}_t)$, and a is an alignment function parameterized by weight matrix $\mathbf{W}_a$.

In this way, the decoder has an improved understanding of dependencies since the output of a time step is dependent on the previous time step and a weighting of the input. After the model converges, it is able to softly align the output to the input sequence's representation by the learned weightings α_{tk} of every time step.

RNN with attention greatly improves the performance on NLG tasks [73, 77]. However, since the encoder needs a forward pass at every time step, the attention mechanism increases the training time by multiples.

2.1.3 Transformer

Shown in Figure 2.2, the transformer architecture [71] is a groundbreaking milestone in natural language processing, revolutionizing the way of modeling dependencies by removing the recurrence that was used in previous language modeling. Through the employment of a self-attention mechanism, the transformer encodes the entire input sequence in parallel, enabling more efficient processing compared with the traditional recurrent architecture. Self-attention [15, 43, 53] refers to the process of calculating attention weights between vector representations within the same input sequence, facilitating the capture of important dependencies between tokens. By contrast, RNNs with attention solely compute what is commonly referred to as cross-attention, representing the attention interaction between the encoder and decoder, as expressed in Equation 2.5. The transformer has emerged as the most successful architecture in the field, with its state-of-the-art performances dominating the field of NLG [16, 42, 58].

Depicted in Figure 2.3, the attention mechanism in the transformer is known as "Scaled Dot-Product Attention Multi-Head Attention". Let $\mathbf{H}_q = [\boldsymbol{h}_1; \cdots; \boldsymbol{h}_Q]$, $\mathbf{H}_k = [\boldsymbol{h}_1; \cdots; \boldsymbol{h}_K]$, and $\mathbf{H}_v = [\boldsymbol{h}_1; \cdots; \boldsymbol{h}_V]$ be the hidden state matrices where $\mathbf{H}_q \in \mathbb{R}^{d_{\text{hid}} \times d_q}$, $\mathbf{H}_k \in \mathbb{R}^{d_{\text{hid}} \times d_k}$, and $\mathbf{H}_v \in \mathbb{R}^{d_{\text{hid}} \times d_v}$. d_{hid} is the model's hidden dimension, while d_q, d_k, and d_v are sequence lengths. Since the sequence might be the input sample or the target tokens, there exists different sequence lengths. The hidden state matrices are initially projected by different linear layers as

$$\mathbf{Q} = \mathbf{W}^Q \mathbf{H}_q \tag{2.6}$$

$$\mathbf{K} = \mathbf{W}^K \mathbf{H}_k \tag{2.7}$$

$$\mathbf{V} = \mathbf{W}^V \mathbf{H}_v \tag{2.8}$$

where $\mathbf{Q}$ is the query matrix, $\mathbf{K}$ is the key matrix, and $\mathbf{V}$ is the value matrix. $\mathbf{W}^Q$, $\mathbf{W}^K$ and $\mathbf{W}^V$ are weight matrices. Subsequently, the attention is computed between

these three matrices as

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_s}}\right)\mathbf{V} \tag{2.9}$$

with $\sqrt{d_s}$ being a scaling factor. Notably, the dot product employed during the attention calculation is more straightforward, as it does not necessitate an additional intermediate training step, unlike the feed-forward neural network (FFNN) in Equation 2.5. This simplicity in the attention calculation contributes to computational efficiency and streamlines the learning process.

In the encoder-decoder transformer architecture, there are three types of attention mechanisms: self-attention in the encoder, self-attention in the decoder, and cross-attention in the decoder. Each demonstrates distinct behaviors based on their specific objectives.

The self-attention mechanism in the encoder operates on the same input hidden state matrix as

$$\text{SelfAttn}(\mathbf{H}) = \text{Attention}(\mathbf{W}^Q\mathbf{H}, \mathbf{W}^K\mathbf{H}, \mathbf{W}^V\mathbf{H}) \tag{2.10}$$

This novel feature of parallel processing, as opposed to sequential processing in models like RNNs, eliminates the issue of missing the previous context from distant time steps. Consequently, the transformer's encoder possesses an effective infinite context window [1, 20, 35], allowing it to capture long-range dependencies in the input sequence. Although RNNs also possesses an infinite context window, the bottleneck design of RNNs impedes their ability to model long-term relationships adequately.

On the other hand, the self-attention mechanism in the decoder takes in the hidden state matrix of the label during training. It is equipped with a masking mechanism to prevent the attention calculation of previous tokens to access information of the future tokens. The mask is applied as

$$\text{MaskedAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\text{AttentionMask}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\right)\mathbf{V} \tag{2.11}$$

where AttentionMask assigns $-\infty$ to the attention scores of future tokens for each token. In this way, the scores will become 0 through the calculation of $\exp\{\cdot\}$ in softmax.

The cross-attention computation in the decoder differs from the self-attention calculation due to the interaction between the encoder and the decoder. Specifically, the hidden state matrix from the encoder is utilized as the input to calculate the key and value matrices, and the query matrix is computed using the decoder's representation matrix as

$$\text{CrossAttn}(\mathbf{H}_{\text{dec}}, \mathbf{H}_{\text{enc}}) = \text{Attention}(\mathbf{W}^{Q}\mathbf{H}_{\text{dec}}, \mathbf{W}^{K}\mathbf{H}_{\text{enc}}, \mathbf{W}^{V}\mathbf{H}_{\text{enc}}) \tag{2.12}$$

where $\mathbf{H}_{\text{dec}}$ and $\mathbf{H}_{\text{enc}}$ are the matrix representations from the decoder and encoder, respectively. This distinction in computation enables the model to effectively capture and leverage internal dependencies between the encoder and decoder representations, facilitating a more comprehensive understanding of the input sequence and generating contextually relevant outputs. This approach intuitively emulates the behavior of a database performing a similarity search with a query. Specifically, the query and key matrices calculate the attention weights by measuring the importance of each encoded input token representation through a dot product. Subsequently, the mechanism performs a weighted sum, utilizing the attention weights and value matrix, to obtain an aggregated vector that captures the contextual information of the input. This process effectively retrieves information that the decoder deems significant from the encoder.

Originally, the transformer model is an encoder-decoder model. However, it can also be an encoder-only model [19] or an decoder-only model [57]. Since the decoder and the encoder utilize different attention mechanisms, their applications become distinctively different.

The encoder-only transformer model is mainly applied to classification tasks. Primary, its self-attention layer allows each token to calculate attention with all the

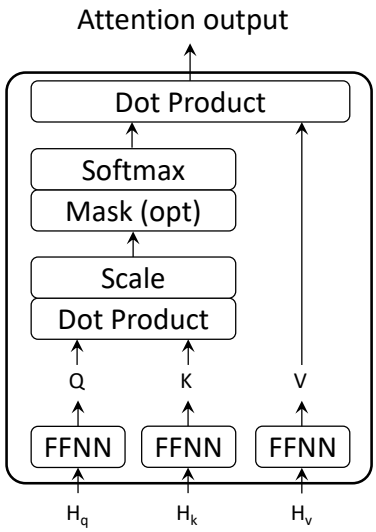

Figure 2.3: The scaled dot product attention in the Transformer architecture.

input tokens. Subsequently, the output from self-attention passes through an multi-layer perceptron (MLP). Finally, the encoder outputs all the hidden states. In this way, the whole computation process is done in parallel, which enables faster training in the encoder transformer. Unlike RNNs, where sequential computations necessitate waiting for the output from preceding steps, the encoder's parallelism significantly lowers the training time.

Conversely, the decoder-only transformer is popular for generation tasks as its objective is to predict the next word given a piece of text. While this architecture shares similar design with the decoder in an encoder-decoder architecture, it differs in key ways. Specifically, each decoder block of the decoder-only architecture only has a masked self-attention layer and an MLP layer. It omits the cross-attention layer that is presented in the decoder of the encoder-decoder architecture, as there is no encoder to interact with . To generate an output, the model first process the input through a masked self-attention layer, and then passes through an MLP layer. In this way, the model generates output sequentially, performing text generation in a step-by-step fashion.

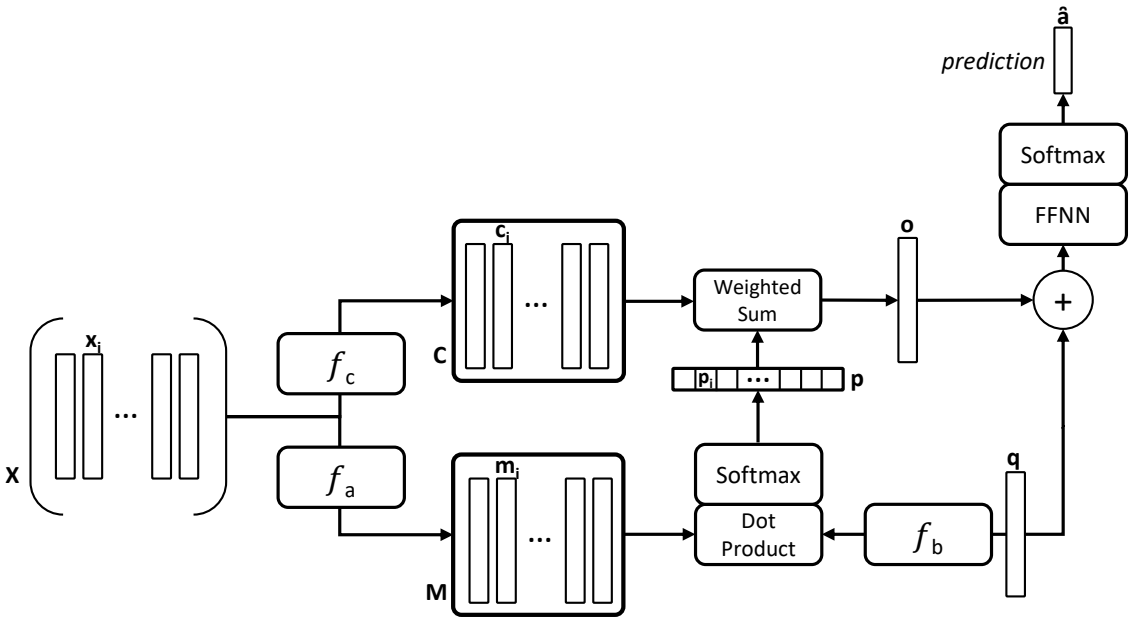

Figure 2.4: The structure of the End-to-end Memory Network.

2.1.4 Memory Network

The end-to-end memory network [67] was proposed to tackle a classification problem: the question-answering task. Its architecture is derived from reference [75] but is able to describe long-term dependencies by incorporating a memory mechanism and an attention mechanism. Given each sample of the question-answering task comprises a set of sentences and a corresponding question, the utilization of memory to store the information of the sentence set and the application of attention mechanisms to locate the correct answer introduces a non-trivial and sophisticated aspect to the model. This combination of memory and attention mechanisms allows the end-to-end memory network to effectively handle complex question-answering scenarios, where understanding long-range dependencies is key to identifying the answer.

Let $\mathbf{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ be a set of sentences and $\boldsymbol{q}$ be the question. The sentences are first embedded with two embedding functions f_C and f_A parameterized by $\mathbf{W}_a \in \mathbb{R}^{d_x \times d_{\text{hid}}}$ and $\mathbf{W}_c \in \mathbb{R}^{d_x \times d_{\text{hid}}}$, respectively. This process yields two memory matrices $\mathbf{M}, \mathbf{C} \in$

$\mathbb{R}^{d_{\text{hid}} \times N}$, given by

$$\mathbf{M} = f_a(\mathbf{X}; \mathbf{W}_a) \tag{2.13}$$

$$\mathbf{C} = f_c(\mathbf{X}; \mathbf{W}_c) \tag{2.14}$$

The memory matrix $\mathbf{M}$ is then queried by the embedding of the question $\boldsymbol{u} = f_b(\boldsymbol{q}; \mathbf{W}_b)$ to an attention probability for each embedding vector $\boldsymbol{m}_i$

$$p_i = \text{softmax}(\boldsymbol{u}^T \boldsymbol{m}_i) \tag{2.15}$$

The output of the memory network is computed as

$$\boldsymbol{o} = \sum_i p_i \boldsymbol{c}_i \tag{2.16}$$

$$\hat{a} = \text{softmax}(\mathbf{W}(o + u)) \tag{2.17}$$

In this way, the system can be trained end-to-end with added dependencies from its attention mechanisms. Such an approach is also extended to sentiment classification [32], dialogue systems [49], etc.

In addition to the end-to-end memory network, other types of memory networks have been proposed to address different problems. For instance, Asghar *et al.* [5] introduce a real-valued memory bank to effectively store distributed knowledge for domain adaptation. Meanwhile, Graves *et al.* [25] present an indirectly parameterized, read-and-write memory, designed to facilitate recurrent information processing. Closely related to these studies is the concept of episodic memory, where previously encountered samples are stored for experience replay [13, 46, 60].

Building on the inspiration derived from the studies mentioned above, our approach focuses on designing a memory network tailored to store factual and template knowledge. This knowledge is then accessed during the decoding process, further enhancing the functionality of our model.

2.2 Reasoning Mechanisms in NLP

For the past few years, reasoning in NLP has been steadily gaining popularity [41, 44]. Natural language reasoning is the process of making inferences based on existing knowledge from the given textual information. There are a few common types of reasoning: inductive, deductive, and abductive. Inductive reasoning is a bottom-up approach to forming a general conclusion from specific premises [34]; deductive reasoning is a top-down approach using forming a specific conclusion from general premises [65]; and abductive reasoning is the process of where the most sensible hypotheses are inferred from ambiguous premises [12]. However, the types of reasoning in many NLP studies may not have a clear boundary due to the approximations in vector spaces. In this section, we investigate the realm of reasoning in NLI due to its close relevance to our work.

A pivotal contribution by MacCartney and Manning [48] expands the repertoire of relations beyond the commonly studied `Entailment`, `Neutral`, and `Contradiction` categories. By proposing seven well-defined natural logic relations, their work offers a finer granularity of understanding for the NLI task, yet challenging models to capture the intricate logic presented in natural language.

Feng *et al.* [22] build on this foundation by devising a neural natural logic model equipped with an attention mechanism. They aim to predict the seven-category natural logic relationships at the word level. However, later work reports that this model falls short in achieving satisfactory reasoning performance [76].

Mahabadi *et al.* [50] propose a parameter-free interactive layer that utilizes fuzzy logic to model the three relations. Within their framework, three scoring functions are employed to compute the relationship scores between the premise and hypothesis. Although the interactive layer performs worse compared with an MLP, it successfully reduces the number of model parameters from millions to eighteen. Notably, the study applies fuzzy logic formulas solely to the real-vectored representations, resulting in

the provision of vague and implicit reasoning. Thus, this approach still lacks explicit interpretation, falling short of the expressive reasoning desired.

While existing research has significantly advanced our understanding of reasoning in NLI, there remains a need to explore more interpretable and expressive models to achieve comprehensive and explicit reasoning capabilities.

2.3 Natural Language Explanation

The natural language explanation (NLE) task aims to explain the decision of a black-box system by generating explanations [64]. The task is sometimes accompanied by a classification task as some previous work requires the model to explain its output [14, 64].

Shown in Figure 1.1, the e-SNLI task [11] includes good quality references for the NLE task. Camburu *et al.* [11] deploy an LSTM-based architecture [17] as a baseline model to tackle the task. The model first performs the NLI classification task and augments the result vector with the bottleneck information to perform explanation generation similar to Equation 2.2. In this way, the NLI information is added to the model, and performance multi-task learning in an end-to-end fashion. The authors also propose to first generate the explanation, and then solve the classification task by conditioning the output only on the explanation. Such a method greatly improves the NLE performance but lowers the classification accuracy compared with the previous approach. This is comprehensible since some generated explanations may not be coherent, and potentially adds noise to the model.

Kumar and Talukdar [40] introduce the NILE approach, which focuses on utilizing the explanations for the NLI task only; the intermediate NLE performance is not evaluated with automatic metric. Inspired by the baseline model [11], NILE carries out the NLE task as the initial step, employing three decoders to generate explanations for each of the `E`, `C`, and `N` labels. These explanations are then scored, and the label prediction is based on the candidate with the highest score. The model demonstrates

competitive performance in the NLI task but only shows human evaluation for the NLE generation.

Several other pertinent studies within the domain use additional data to improve on the NLE task. For instance, the LIREx approach [81] adopts supplementary annotated rationales to enhance the generation of explanations. Similar to NILE, they use the explanations to improve on the NLI task and only systematically evaluate the classification performance. However, this methodology provides some insights into how explanations can be facilitated through the incorporation of additional annotations. Furthermore, the WT5 model [51] is built upon the T5 model to perform both the sentence classification and the NLE task for multiple domains by fine-tuning a T5 with multiple explanation generation datasets. This approach leverages the power of pre-trained language models to truly address explanation generation, enriching the landscape of research in this field.

While the mentioned systems may excel in generating explanations, it is important to note that the nature of such finetuning approaches renders the explanation generator *per se* unexplainable. In contrast to the aforementioned methods, we present a novel memory network architecture that effectively integrates factual and template knowledge, acquired through a weakly supervised approach. This integration significantly enhances the overall explainability of our proposed method.

2.4 Summary

This chapter provides a comprehensive overview of natural language generation (NLG) by first introducing its concepts, and then its landscape of tasks. Subsequently, we focus on reviewing revolutionary approaches that offer novel modeling techniques for NLG. For each approach, we analyze the specific problems it addresses, which were unresolved by previous work, as well as the challenges associated with the new methods. Furthermore, relevant models that bear significance to our own work are also covered within this discussion.

The following section examines various reasoning techniques in NLP, where we provide details of the novelty of each method and highlight their contributions to the field of reasoning. Moreover, we pinpoint the disadvantages of each approach to provide a better picture of the field of reasoning.

The final section discusses our task of natural language explanation, with a focus on the e-SNLI dataset. We inspect the existing work in this field, outlining their areas of excellence as well as identifying aspects that may require improvement. Moreover, we diligently compare each work, considering various aspects such as the input information, methodology, and the evaluation metrics they used in their systems.

Chapter 3

Approach

3.1 Overview

In this chapter, we propose an approach that effectively captures the underlying rationale information from the data, eliminating the need for human annotation. We employ a phrasal logic relationship detection approach that extracts factual knowledge tuples, detailed in Section 3.2. In Section 3.3, we present a memory network that utilizes these factual tuples. Likewise, we propose a method in Section 3.4 to extract template information and incorporate them as slots into a memory pad. Finally, we show the integration of these networks into a carefully engineered decoder architecture for the explanation generation process in Section 3.5.

3.2 Phrasal Logic Relationship Detection

In previous research [76, 82], an Explainable Phrasal Reasoning (EPR) approach is developed to determine phrasal logic relationships for the natural language inference (NLI) task[1]. My thesis employs this line of work and takes its output as factual knowledge. The EPR method involves several key steps to enhance the reasoning capabilities of a model. The method starts by devising a set of rules that aim at

[1]My contribution to EPR lies in the Phrase Detection and Alignment section, where I proposed and implemented the algorithm for obtaining the phrase alignment. Specifically, a phrase pair $(\boldsymbol{p}_m, \boldsymbol{h}_n)$ is considered to be aligned if $\boldsymbol{h}_n$ is selected as the closest phrase to $\boldsymbol{p}_m$, and $\boldsymbol{p}_m$ is the closest to $\boldsymbol{h}_n$ in terms of similarity scores.

detecting and aligning phrases in both the premise and hypothesis. Subsequently, the work proposes a neural fuzzy logic model to predict the logical relationship between each pair of aligned phrases. Specifically, EPR first embeds each of these phrases individually into vector representations and feeds them to an multilayer perceptron (MLP) to produce three-dimensional score vectors to represent the probability of each relation. The scores are then converted into probabilities using the Softmax function. To determine the overall sentence label, the work adopts an inductive reasoning approach by introducing novel custom fuzzy logic formulas tailored to each of the `E`, `C`, and `N` relationships.

Consider $\{(\mathrm{p}_k, \mathrm{h}_k)\}_{k=1}^{K} \bigcup \{(\mathrm{p}_k, \mathrm{h}_k)\}_{k=K+1}^{K'}$ to be all the detected phrase pairs. The phrases are aligned for $k = 1, \cdots, K$, while $k = K+1, \cdots, K'$ are unaligned phrases. An unaligned premise phrase is paired with the special token $\mathrm{h}_{\langle \mathrm{EMPTY} \rangle}$ and an unaligned hypothesis phrase is paired with the special token $\mathrm{p}_{\langle \mathrm{EMPTY} \rangle}$. Then, the EPR defines the fuzzy logic scores for `E`, `C`, and `N` as

$$S_{\text{sentence}}(\texttt{E}|\mathrm{P}, \mathrm{H}) = \Big[\prod\nolimits_{k=1}^{K'} P_{\text{phrase}}(\texttt{E}|\mathrm{p}_k, \mathrm{h}_k) \Big]^{\frac{1}{K'}} \tag{3.1}$$

$$S_{\text{sentence}}(\texttt{C}|\mathrm{P}, \mathrm{H}) = \max\nolimits_{k=1,\cdots,K} P_{\text{phrase}}(\texttt{C}|\mathrm{p}_k, \mathrm{h}_k) \tag{3.2}$$

$$S_{\text{sentence}}(\texttt{N}|\mathrm{P}, \mathrm{H}) = \Big[\max\nolimits_{k=1,\cdots,K'} P_{\text{phrase}}(\texttt{N}|\mathrm{p}_k, \mathrm{h}_k) \Big] \cdot \Big[1 - S_{\text{sentence}}(\texttt{C}|\mathrm{P}, \mathrm{H}) \Big] \tag{3.3}$$

The entailment rule posits that the premise entails the hypothesis if all the phrase pairs demonstrate an `Entailment` relationship. Conversely, the contradiction rule states that the premise and hypothesis are classified as `Contradiction` if there exists at least one phrase pair that exhibits a contradictory relationship. Meanwhile, the neutral rule indicates that the premise is deemed `Neutral` to the hypothesis when a `Neutral` phrase pair exists, and there are no contradicting phrase pairs present. The nature of fuzzy logic permits the above rules to be a real-valued score between 0 and 1, which allows a more nuanced way of reasoning compared to the traditional binary logic. Therefore, fuzzy logic is well-suited to tolerate the ambiguity of natural

Factual Memory

Phrase in p	Phrase in h	Label
a woman	*the woman*	E
running a marathon	*running*	E
in a park	*in her backyard*	C

Figure 3.1: Example of the factual memory

language. Additionally, as fuzzy logic formulas output real-valued scores, it is able to facilitate backpropagation, which in turn enables EPR to perform end-to-end training.

Finally, the scores are normalized into probability by dividing the sum of all the scores

$$P_{\text{sentence}}(\mathtt{L}|\cdot) = \frac{S_{\text{sentence}}(\mathtt{L}|\cdot)}{S_{\text{sentence}}(\mathtt{E}|\cdot)+S_{\text{sentence}}(\mathtt{C}|\cdot)+S_{\text{sentence}}(\mathtt{N}|\cdot)} \tag{3.4}$$

where $\mathtt{L} \in \{\mathtt{E}, \mathtt{C}, \mathtt{N}\}$ is the groundtruth sentence-level label.

EPR is trained with cross-entropy loss by minimizing $-\log P_{\text{sentence}}(\mathtt{L}|\cdot)$ with the groundtruth sentence-level label. In this way, the logical reasoning component is trained end-to-end in a weakly supervised manner using backpropagation.

3.3 Factual Memory

In this thesis, we propose to utilize such predicted phrasal logical relationships as factual knowledge to enhance the generation of textual explanation. As illustrated in Figure 3.1, EPR yields a set of tuples $\{(\mathrm{p}_k, \mathrm{h}_k, \mathrm{l}_k)\}_{k=1}^{K}$, where l_k is the predicted phrasal label (E, C, or N) for the aligned phrases, p_k and h_k.

We apply Sentence-BERT (SBERT, Reimers and Gurevych, 2019) on individual phrases to obtain the phrasal embeddings as

$$\boldsymbol{p}_k = \mathrm{SBERT}(\mathrm{p}_k) \tag{3.5}$$

$$\boldsymbol{h}_k = \mathrm{SBERT}(\mathrm{h}_k) \tag{3.6}$$

Template Memory

Template	Label
⋯ *implies* ⋯	E
⋯ *same as* ⋯	E
⋯ *not* ⋯	C
⋯ *cannot* ⋯ *same time*	C
⋯ *does not imply* ⋯	N
just because ⋯ *does not mean* ⋯	N

Figure 3.2: Example of the template memory

The phrase-level NLI label is encoded as a one-hot vector by

$$\boldsymbol{l}_k = \text{onehot}(\text{l}_k) \tag{3.7}$$

We concatenate these representations to form a vector for the factual tuple $(\text{p}_k, \text{h}_k, \text{l}_k)$:

$$\boldsymbol{m}_k = [\boldsymbol{p}_k; \boldsymbol{h}_k; \boldsymbol{l}_k] \tag{3.8}$$

where $[\cdot;\cdot]$ represents column vector concatenation.

We compose the vectors as a factual memory matrix:

$$\mathbf{M}_f = [\boldsymbol{m}_1^\top; \cdots; \boldsymbol{m}_K^\top] \tag{3.9}$$

where $\mathbf{M}_f \in \mathbb{R}^{K \times d}$ and d is the dimension of $\boldsymbol{m}_k$. The knowledge of factual memory will be fed to the decoder by an attention mechanism, detailed later.

In this way, we are able to enhance an encoder-decoder model with interpretable factual knowledge of phrases and their relationships, given by weakly supervised reasoning.

3.4 Template Memory

We observe that NLI explanations often share similar expressions, which suggests there are common structures among the human-annotated explanation references.

Therefore, we design a simple yet effective approach to extract these expressions and integrate them into our model by a template memory.

To extract a template, we take the following steps and provide our rationale:

1. Mask words in the reference explanation that also occurs in either the premise or hypothesis.

   According to our observation, such overlapping words are typically content-specific. This is not a specific phenomenon that only occurs in the e-SNLI dataset. Fundamentally, templates are designed for repeated use across a range of samples in a broader sense. Consider the following case: "*Premise: A person on a horse jumps over a broken down airplane. Hypothesis: A person is training his horse for a competition. Reference: the person is not necessarily training his horse.*" We mask the overlapping words "*person*", "*is*", "*training*", "*his*", and "*horse*", and obtain "*the* $\cdots$ *not necessarily* $\cdots$".

2. Remove words that belong to a predefined set of stop words.

   Words such as *a* and *the* are ubiquitous in English. For many tasks in NLP, they do not carry much meaningful context. In our case, these redundant words add noise to our templates. By removing the stop word "*the*" from the case in the previous step, the template is much cleaner in the form of "$\cdots$ *not necessarily* $\cdots$".

3. Apply procedures 1 and 2 for all the samples in the dataset.

   As we iterate through all the samples, we keep a dictionary for the count of each template and its category.

4. Keep a list of top-k most frequent templates for each of the `E`, `C`, and `N` categories.

Our goal is to extract the most representative templates, which is given by the count. To our observation, infrequent templates may be noisy. For example, the extracted template "$\cdots$ *he always* $\cdots$" under the label `E`, "$\cdots$ *cannot and simultaneously* $\cdots$" under the label `C`, or "$\cdots$ *doesn't mean he* $\cdots$" under the label `N`; these templates are infrequent templates tailored for particular input samples. Therefore, by capturing the most contextually representative template, we likely encompass the majority of the templates in e-SNLI.

We apply SBERT to embed the extracted templates (denoted by $\mathrm{T} = \{\mathrm{t}_c\}_{c=1}^{C}$). A template memory slot is $\boldsymbol{m}_c = [\mathrm{SBERT}(\mathrm{t}_c); \mathrm{onehot}(\mathrm{l}_c)]$, where $\mathrm{l}_c \in \{\mathtt{E}, \mathtt{C}, \mathtt{N}\}$ is the category of the template. The entire template memory matrix is thus $\mathbf{M}_p = [\boldsymbol{m}_1^\top; \cdots; \boldsymbol{m}_C^\top] \in \mathbb{R}^{C \times d'}$, where d' is the dimension of $\boldsymbol{m}_c$.

Different from factual knowledge, we restrict the attention to templates that have the same label as predicted by EPR [76], detailed in Section 3.5. This ensures that only relevant templates are used for generating an explanation.

3.5 Decoder Architecture

Our decoder follows a standard Transformer architecture [71] but is equipped with additional attention mechanisms to factual and template memories (Figure 3.3).

Consider the ith decoding step. We feed the factual memory to an MLP as $\tilde{\mathbf{M}}_f = \mathrm{MLP}(\mathbf{M}_f)$. We compute attention $\boldsymbol{a}_f$ over $\tilde{\mathbf{M}}_f$ with the embedding of the input $\boldsymbol{y}_{i-1}$, and aggregate factual information $\boldsymbol{c}_f$ for the rows $\boldsymbol{m}_{ft}$ in $\mathbf{M}_f$:

$$\boldsymbol{a}_f = \mathrm{softmax}(\tilde{\mathbf{M}}_f \, \boldsymbol{y}_{i-1}) \tag{3.10}$$

$$\boldsymbol{c}_f = \sum_{k=1}^{K} a_{fk} \tilde{\boldsymbol{m}}_{ft}^\top \tag{3.11}$$

where a_{fk} is the kth element of the vector $\boldsymbol{a}_f$ and $\hat{\boldsymbol{m}}_t$ is the kth row of the matrix $\tilde{\mathbf{M}}_f$.

On the other hand, another attention mechanism fetches template information in a similar way by $\tilde{\mathbf{M}}_p = \text{MLP}(\mathbf{M}_p)$ and $\boldsymbol{a}_p = \text{softmax}(\tilde{\mathbf{M}}_p \boldsymbol{y}_{i-1})$, but the attention is masked:

$$\boldsymbol{a}_p = \text{softmax}(\text{AttentionMask}(\tilde{\boldsymbol{a}}_p)) \tag{3.12}$$

where AttentionMask selects only the templates that have the same label as EPR's predicted sentence NLI label. This is accomplished by assigning $-\infty$ to some index(es) in the vector $\tilde{\boldsymbol{a}}_p$, resulting in $\exp\{\cdot\}$ being 0 for softmax computation. This attention aggregates template information, denoted by $\boldsymbol{c}_p$, where $\boldsymbol{c}_p = \sum_{k=1}^{K} a_{pk} \tilde{\boldsymbol{m}}_{pt}^{\top}$ for the rows $\boldsymbol{m}_{pt}$ in $\tilde{\mathbf{M}}_p$. It is simply added to factual information as $\boldsymbol{c} = \boldsymbol{c}_f + \boldsymbol{c}_p$, which is then fed to a subsequent layer

$$\boldsymbol{g}_i = \text{LayerNorm}(\text{MLP}([\boldsymbol{c}; \boldsymbol{y}_{i-1}]) + \boldsymbol{c}) \tag{3.13}$$

Our Transformer decoder layer starts with self-attention

$$\begin{aligned} \tilde{\boldsymbol{q}}_i &= \text{SelfAttn}(\boldsymbol{g}_i) \\ &= \text{Attention}(\mathbf{W}^Q \boldsymbol{g}_i, \mathbf{W}^K \boldsymbol{g}_i, \mathbf{W}^V \boldsymbol{g}_i) \end{aligned} \tag{3.14}$$

Then, residual connection and layer normalization are applied as

$$\boldsymbol{q}_i = \text{LayerNorm}(\tilde{\boldsymbol{q}}_i + \boldsymbol{g}_i) \tag{3.15}$$

A cross-attention mechanism obtains input information by

$$\begin{aligned} \boldsymbol{v}_i &= \text{CrossAttn}(\boldsymbol{q}_i, \mathbf{H}) \\ &= \text{Attention}(\mathbf{W}^Q \boldsymbol{q}_i, \mathbf{W}^K \mathbf{H}, \mathbf{W}^V \mathbf{H}) \end{aligned} \tag{3.16}$$

where $\mathbf{H}$ is the representation given by the encoder. $\boldsymbol{v}_i$ is fed to the Transformer's residual connection and layer normalization sub-layer.

Multiple Transformer layers, mentioned above, are stacked to form a deep architecture. The model is trained with the standard cross-entropy loss against the reference explanation as in previous work [40, 51, 81].

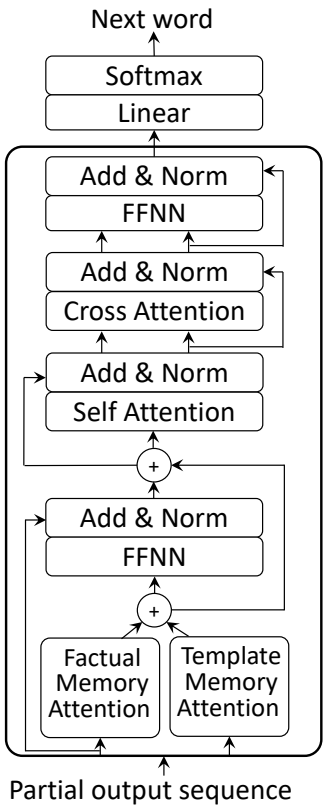

Figure 3.3: Our decoder architecture.

In this way, we enhance the model with factual information given by the EPR weakly supervised reasoning and the template information from our rule-based extraction. Experiments will show that our approach greatly improves the on the BLEU metrics score by 2 points (Section 4.5), achieving a new state of the art result. Hence, we are able to verify that the EPR indeed yields meaningful phrasal factual phrase pairs and the templates provide useful underlying information.

3.6 Training and Inference

We train our model using the cross-entropy loss, which involves minimizing the equation

$$J_\theta = -\sum_{t=1}^{T} \log \hat{p}_\theta(w_t|w_{<t}) \tag{3.17}$$

for each sample. Here, w_t is the target next word given the prefix in the ground truth, and $\hat{p}_\theta$ denotes the predicted probability parameterized by θ. The memory networks are jointly trained with the backbone model in an end-to-end fashion.

For a given sample during the inference stage, we use EPR to extract phrase pairs with their logical relationship to form the factual memory. While template memory is not changed, it is masked by EPR's predicted label (either `E`, `C`, or `N`). These memory information is fed to the decoder to generate the NLI explanation.

3.7 Summary

In this chapter, we provided a comprehensive explanation of our approach, starting with phrasal logic relationship detection. EPR supplies phrase pairs that are either aligned or unaligned. Subsequently, we utilized these phrase pairs by first converting them into vector representations, and then constructing these vectors into memory matrices to serve as factual knowledge. The memory matrix is then fed to the decoder. Regarding the template memory, we extracted templates from the dataset and converted them into vector representations. These vectors were then composed into a template memory matrix and fed to our decoder. The decoder fetches the factual and template information by applying attention mechanisms to the memory matrices. In this way, we were able to enhance the model with factual information given by the EPR's weakly supervised reasoning and the template information from our rule-based extraction.

In conclusion, we comprehensively outlined our approach in this chapter with intricate details of each and every component within our design. In this way, we are able to support a thorough understanding of our research.

Chapter 4

Experiments

4.1 Overview

In this chapter, we start by presenting the details of the e-SNLI dataset in Section 4.2. Then, we go through the specific settings of our metrics in Section 4.3. We describe the important implementation details of our approach in Section 4.4. After setting up the background, we present our main results in Section 4.5, and then follow up with our ablation study in Section 4.6. We further show analysis and case study of our model architecture in Sections 4.7 and 4.8. In Section 4.9, we analyze the different architectural designs of our decoder. Then, we follow up with a more in-depth analysis of our memory component design in Section 4.10.

4.2 Dataset

We evaluate our model on the e-SNLI dataset [11], which contains 550K training samples, 10K validation samples, and 10K test samples. All samples possess a label from the SNLI dataset. Each training sample has one reference explanation, whereas each validation or test sample contains three reference explanations. In addition, rationales are provided for each reference explanation. The rationales are used to establish the integrity of the annotated explanations. For example, a test case "Premise: An old man with a package poses in front of an advertisement. Hypothesis: A man poses in front of an ad." has three explanations with highlighted rationales:

1. Explanation: “An ad is the short form for advertisement.” Rationales: “Premise: An old man with a package poses in front of an *advertisement*. Premise: A man poses in front of an *ad*.”

2. Explanation: “A man poses in front of an ad is the same as a man poses in front of advertisement because ad is an abbreviation for advertisement.” Rationale: “Premise: An old man with a package poses in front of an *advertisement*. Hypothesis: A man poses in front of an *ad*.”

3. Explanation: “The word ‘ad’ is short for the word ‘advertisement’.” Rationales: “Premise: An old *man* with a package *poses* *in* *front* *of* an *advertisement*. Hypothesis: A *man* *poses* *in* *front* *of* an *ad*.”

Previous literature [81] uses these rationales to improve performance. However, we do not use them, and only use the references for training and evaluation.

4.3 Metrics

In previous work, evaluation methods were inconsistent: citation [11] evaluates output with BLEU on two references, citation [51] evaluates output with SacreBLEU on two references, and citations [40, 81] report human evaluations of generated output on 100 generated explanations. In this thesis, we consider the above variants of BLEU scores, including both two-reference and three-reference BLEU and SacreBLEU scores. In this way, we are able to provide a robust set of evaluation metrics.

4.3.1 BLEU

The BLEU score [52] has been widely used in various NLP tasks, ranging from machine translation [21, 68] to dialogue generation [23, 80]. This evaluation method outputs a score between 0 and 1, quantifying the similarity between a candidate text and its reference.

Let the candidate text be $w_c = (w_1, \cdots, w_L)$ and the reference text be $w_r = (w_1, \cdots, w_M)$ after tokenization where L and M are the sequence length. An uni-gram of w_c would be (w_i), a bi-gram would be (w_i, w_{i+1}), and a n-gram would be $(w_i, \cdots, w_{i+l-1})$. A set of n-gram of w_c would be $\text{n-gram}(w_c) = \{(w_i, \cdots, w_{i+l-1})\}_1^{L-l+1}$. The precision of n-grams is defined as

$$p_n = \frac{\sum_{i=1}^{L-l+1} \text{Clip}(\mathbb{1}\{\text{n-gram}(w_c)_i \in w_r\})}{\text{n-gram}(w_c)} \tag{4.1}$$

For the candidate text, the count of $\mathbb{1}\{\text{n-gram}(w_c)_i \in w_r\}$ is clipped to match the maximum number of times an n-gram appears in the reference text. For example, consider w_c = ("the", "the", "the") and w_r = ("the", "times"). In this case, the count of "the" for calculating BLEU would be 1 since there is only one occurrence of "the" in w_r. Consequently, the BLEU score of this example would be $\frac{1}{3}$.

Then, the BLEU score is defined as

$$BP = \begin{cases} 1 & \text{if } |w_c| \geq |w_r|, \\ \exp(1 - |w_r|/|w_c|) & \text{otherwise.} \end{cases} \tag{4.2}$$

$$\text{BLEU} = BP \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \quad \sum_{n=1}^{N} w_n = 1 \tag{4.3}$$

where BP is the brevity penalty for length, N is the n value for n-gram, and w_n is the weight hyper-parameter for each n-gram score. In other words, the BLEU score can be seen as the weighted geometric mean of all the modified n-gram precisions with the brevity penalty. For our use of the BLEU score, we follow previous papers [11, 51] and set the N to 4 and w_n to $1/N$.

4.3.2 Multi-reference BLEU

BLEU evaluates whether each token in the candidate text is present in any of the reference texts when multiple references are available. Additionally, the clipping process ensures that the maximum count of the candidate n-gram does not exceed its maximum occurrence in any of the reference texts. In this framework, only the

precision aspect of the n-gram calculation is modified. Let $w_s = \{w_{rk}\}_{k=1}^{K}$ be a set of references. The precision is calculated as

$$p_n = \frac{\sum_{i=1}^{L-l+1} \text{Clip}_{\max(\text{ws})}(\mathbb{1}\{n\text{-gram}(w_c)_i \in \cup(\{w_{rk}\}_{k=1}^{K})\})}{n\text{-gram}(w_c)} \tag{4.4}$$

Let the candidate be w_c = ("the","times","paper") and the references be w_{r1} = ("the","good","deed"), w_{r2} = ("let","times","fly"), and w_{r3} = ("reading","a","paper"). Since all the words in w_c appeared separately in all the references. The BLEU score for this example would be 1.

4.3.3 SacreBLEU

The SacreBLEU score [56] is a special case of the BLEU score that was developed for reproducibility. SacreBLEU is computed as:

$$\text{SacreBLEU} = BP \exp\left(\frac{1}{N}\sum_{n=1}^{N} \log p_n\right) \tag{4.5}$$

where p_n is defined by Equation 4.4 and BP is given by Equation 4.2. The score calculation is identical to BLEU, but it does not allow user to change the weights.

Unlike BLEU, SacreBLEU does not permit user-supplied tokenization rule. Instead, it applies a fixed text preprocessing scheme featuring a set of improved tokenization rules. Additionally, SacreBLEU aligns its parameters, such as smoothing applied to zero-count n-grams, to those defined by the Conference on Machine Translation (WMT).

In this way, SacreBLEU is able to standardize BLEU scores across different papers for reproducible results.

4.4 Implementation Details

We start the implementation process by retrieving K factual tuples $\{(\text{p}_k, \text{h}_k, \text{l}_k)\}_{k=1}^{K}$ with EPR. We then use the pre-trained model *paraphrase-MiniLM-L6-v2* from SBERT [61][1]

[1]https://www.sbert.net/docs/pretrained_models.html

Model	Info		BLEU		SacreBLEU	
	L	H	2 refs	3 refs	2 refs	3 refs
ExplainThenPredictAttention [11]§	–	–	27.58	–	–	–
NILE [40]‖	✓	–	28.57	37.73	32.51	41.78
NILE [40]†	✓	–	28.67	37.84	32.74	42.06
FinetunedWT5$_{220M}$ [51]§	✓	–	–	–	32.40	–
FinetunedWT5$_{11B}$ [51]§	✓	–	–	–	33.70	–
LIREx [81]‖	✓	✓	17.22	22.40	21.24	26.68
Finetune T5$_{60M}$	–	–	27.75	36.78	31.74	40.89
+ Factual$_{64M}$	–	–	29.14	37.81	33.23	41.96
+ Template$_{63M}$	–	–	29.22	37.87	33.05	41.96
+ Both$_{65M}$	–	–	**29.55**	**38.38**	**33.45**	**42.68**

Table 4.1: Main results. Previous work uses auxiliary information (L: the groundtruth NLI label; H: human-annotated rationales), but we use neither. §Numbers taken from previous papers. †Evaluated by checkpoints. ‖Our replication with provided code.

to embed these tuples into 384-dimensional embeddings. Similarly, we first extract C template expressions $\{t_c\}_{c=1}^{C}$ with our extraction method detailed in Section 3.4, and use SBERT to embed them as 384-dimensional embeddings. To match the T5 small's 512 model dimensions, we use an multilayer perceptron (MLP) to project the embeddings from 384 to 512. Finally, we construct the factual memory matrix by concatenating the factual embeddings as $\mathbf{M}_f \in \mathbb{R}^{K\times 512}$ and the template memory by concatenating the template embeddings as $\mathbf{M}_p \in \mathbb{R}^{C\times 512}$.

During training, we use the pretrained T5 small model with a batch size of 32. We utilize the Adam optimizer [38] with an initial learning rate of 3e-4, and set the decay rates as $\beta_1 = 0.9, \beta_2 = 0.999$. We apply learning rate warm-up for the first 2 epoch, and make the learning rate linearly decay for 10 epochs to decrease it to 3e-6. We continue training the model until the validation BLEU score does not increase for 2 consecutive epochs.

4.5 Main Results

Table 4.1 shows explanation generation performance on e-SNLI. It's important to note that evaluation metrics across previous studies were not consistent, that is, some [11] use BLEU, and others [51] use SacreBLEU. To ensure consistency and fair comparison, we replicated their approaches using either their provided code or checkpoints. For large pretrained models, we quote results from the WT5 [51]. Our model is based on T5-small with 60M parameters, due to resource limitation. Despite using a smaller model, we still achieve competent performance on the task.

As seen, most previous studies use groundtruth NLI labels and/or highlighted rationales (Fig. 1.1). This requires human annotations, which are resource-consuming to obtain and may often be unavailable when predicting the explanation. By contrast, our factual knowledge leverages a weakly supervised reasoning approach [76], and our templates are extracted with simple rules. We use no additional information but still outperform all previous work in terms of all metrics. This demonstrates the effectiveness of our approach as we are able to generate higher-quality explanations while not relying on any human-annotated knowledge.

Narang *et al.* [51] finetune a T5 model with multiple explanation tasks, namely, the explanations for sentiment analysis, question answering, reading comprehension, as well as e-SNLI. Their model is called WT5, having 220M or 11B parameters depending on the underlying T5 model. Profoundly, we achieve higher performance with 60M-parameter T5-small, which is 3.3x and 170x smaller in model size than the two WT5 variants [51]. This signifies the importance of knowledge grounding in explanation generation.

4.6 Ablation Study

In order to gain a deeper understanding of the individual contributions of various components in our proposed approach, we conduct an ablation study, as illustrated

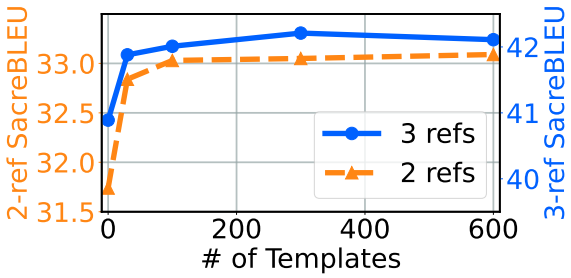

Figure 4.1: Analysis of the template memory size.

Input *Premise : Several people in blue scrubs and one in a skirt and black blouse. Hypothesis : Several people are naked.*			
Label `Contradiction` (not used during our explanation generation)			
Memory net content			
Premise phase	Hypothesis phase	EPR label	Attention score
Several people	Several people	E	30.10
in a skirt and black blouse	naked	C	54.76
in blue scrubs	[EMPTY]	E	15.14
Output explanation People cannot be naked and in scrubs at the same time.			
Reference explanations (1) If the several people here are naked, then they cannot be in scrubs or a skirt and blouse. (2) If people are in scrubs, a skirt, and a blouse they are not naked. (3) People can't wear blue scrubs and be naked simultaneously.			

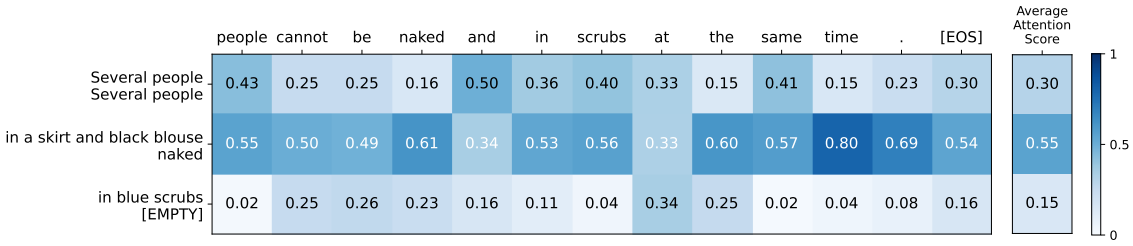

Figure 4.2: Case study of the factual memory. The heat map shows the step-by-step and average attention weights to the factual memory tuples (vertical axis).

in Table 4.1. We finetune a plain T5. Its performance is similar to ExplainThen-PredictAttention [11], which is a model that first generates an explanation and then predicts an SNLI label based solely on the generated explanation using an attention mechanism. However, it performs worse than NILE and WT5.

To explore the effect of each individual memory component, we apply factual memory and template memory separately. As seen, each memory individually improves

the performance by 1–1.5 BLEU scores, which indicates that both factual knowledge and template knowledge plays a crucial role in our approach.

Furthermore, by combining the two memory modules, we obtain a further improvement of ~ 0.5 BLEU score and outperform all previous work. This suggests the effectiveness of our approach.

4.7 Analysis of the Template Memory Size

To evaluate the template memory's impact on the model performance, we conduct quantitative experiments with various template memory sizes. We utilize the T5-small model without the factual memory so as to exclude its effect. We specifically choose the numbers of templates for evaluation from $\{0, 30, 100, 300, 600\}$, where the size of 0 indicates the T5-small baseline.

As depicted in Figure 4.1, a small template memory can already benefit the model. The results also show that the performance continues to increase with more and more templates. However, due to efficiency concerns, we choose the template size of 300. This allows us to achieve satisfactory results without excessive computational overhead.

4.8 Case Study of the Factual Memory

In Figure 4.2, we present a case study on the factual memory. The template memory is excluded to eliminate its effect since the aim of this study is to analyze how the proposed memory network performs in attending to structured factual tuples. As seen, EPR's [76] weakly supervised reasoning approach yields meaningful structured factual tuples, namely, *Several people* entailing *Several people*, *naked* contradicting *in a skirt and black blouse*, and *in blue scrubs* unaligned (matched with a special token `[EMPTY]`). Our proposed factual memory network attends to these factual tuples. A more detailed heat map illustrates that our model assigns the most attention weights

Design	BLEU		SacreBLEU	
	2 refs	3 refs	2 refs	3 refs
After cross-attention	29.42	37.96	**33.72**	42.23
Before cross-attention	29.21	38.25	33.17	42.63
Before self-attention	**29.55**	**38.38**	33.45	**42.68**

Table 4.2: Analysis of different architecture for memory network in the decoder.

averaging 0.55, to the contradicting tuple *in a skirt and black blouse* and *naked*. Consequently, the model generates the explanation "People cannot be naked and in scrubs at the same time." This confirms that our memory network learns meaningful attention to the factual knowledge, which also improves the explainability of explanation generation *per se*.

4.9 Analysis of the Decoder Architectural Design

We conduct extensive experiments to evaluate the architecture of our decoder. Specifically, we place our memory networks at different positions in the decoder. We consider three places: (1) Before self-attention; (2) Before cross-attention; and (3) After cross-attention. For each architecture, only the placement of our memory network component has changed; the content of both memories remains the same. In this way, our evaluation is rigorously controlled to establish scientific conclusions. We run each experiment once with the same experimental setting outlined in Section 4.4.

As seen in Figure 4.2, placing the memory network before self-attention gives the best performance for almost all the metrics, which shows that applying our memory networks earlier in the architecture yields the best performance. In other words, the model learns the best when our factual and template knowledge is provided earlier in the architecture.

When we focus on using three references for both BLEU and SacreBLEU evaluations, we observe that as our networks are positioned later in the decoder, performance

Design	Model	BLEU		SacreBLEU		Average
		2 refs	3 refs	2 refs	3 refs	
Sequential memory	Factual → Template	28.47	37.36	32.52	41.82	35.04
	Template → Factual	27.35	36.13	31.31	40.40	33.80
Individual memory	Factual + Template	**29.55**	**38.38**	**33.45**	**42.68**	**36.02**

Table 4.3: Analysis of different designs for the memory network component.

consistently decreases. By contrast, when using only two references, performance shows fluctuations. This observation suggests that evaluations with two references may introduce more variability.

4.10 Analysis of the Memory Component Design

Given that our model incorporates two memory networks, the arrangement of these networks is crucial to harness their full capabilities. In this section, we conduct a study of our memory component design. Specifically, we consider two different methods:

1. Sequence memory uses the output of the previous memory network as the next memory's query. For example, with "Factual → Template", factual memory is utilized by Equation 3.10, but template memory is queried by the factual information as $\boldsymbol{a}_p = \text{softmax}(\tilde{\mathbf{M}}_p \, \boldsymbol{c}_f)$.

2. Individual memory, described in Section 3.5.

By exploring these configurations, we aim to identify the best-performing arrangement for our memory networks.

As illustrated in Figure 4.3, the memory pad arrangement in the Sequential memory significantly influences its performance. On average, the "Factual → Template" arrangement is 1.2 points higher than 'Template → Factual". Nevertheless, as the individual memory design has an average score of 36.02, it scores at least one point higher than any of Sequential memory variants. Consequently, we can conclude that the individual memory design is superior. as Moreover, we conduct a study on the

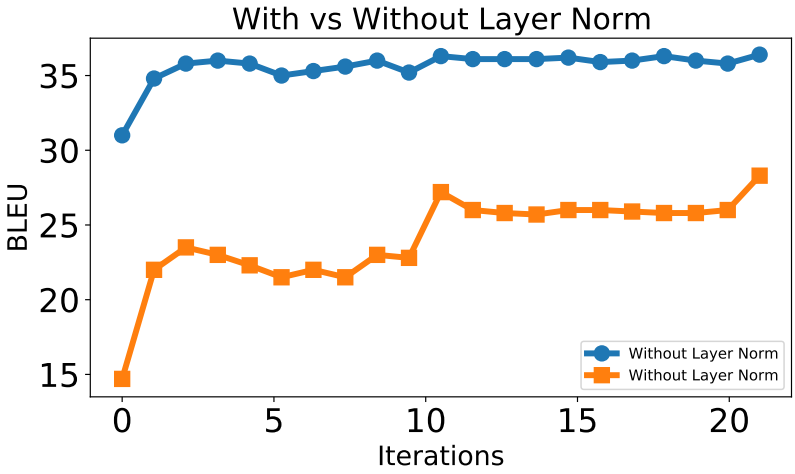

Figure 4.3: Experiments of with and without layer normalization on the validation set.

other crucial part of our memory network component—namely, layer normalization (LayerNorm). This technique is a popular method for normalizing the distributions within a neural model and smoothing the gradient to enable better model generalization [78]. Intuitively, the need for such normalization arises because we employ Sentence-BERT to obtain memory matrices and T5-small to perform explanation generation, which results in a distribution mismatch. Therefore, we need LayerNorm to harmonize the distribution between the two.

We conduct experiments on the validation set with our model, where we run each variation for 21 epochs with the same experimental setting to draw scientific conclusions. As seen in Figure 4.3, the model performs much better when the normalization layer is present, which aligns with our expectations. In conclusion, LayerNorm is essential for our method as it can assist with matching the distribution between our memory matrices and the explanation generation model.

4.11 Summary

In this chapter, we began by introducing the e-SNLI dataset, which is an extension of the SNLI dataset for the explanation generation task. The dataset contributed more than 600k human-annotated references. We followed up with the evaluation

metrics associated with e-SNLI. For each metric, we explained their tokenization and evaluation method. Furthermore, we highlighted the similarities and differences between the metrics.

In the succeeding section, we offered detailed records of our implementation. This is crucial as it encompasses the essence of reproducibility for research in NLP. We supplied the minutiae to the preprocessing steps, the models we utilized, and the hyper-parameters we used.

Subsequently, we discussed our experiments in detail. We presented our main result with a table that includes evaluations of previous work based on either replication, checkpoints, or both, along with the results from our model. The table included the multi-reference BLEU and SacreBLEU metrics, mentioned in Section 4.3. We compared previous work with our own, assessing the difference in the utilization of data and the model performance. In Section 4.6, we provided the ablation experiments, where we displayed the results of ablated models and examined their performance alongside our main results. Subsequently, we analyzed the template memory in terms of different numbers of templates with a figure and presented a case study of the factual memory.

Finally, we investigated our architectural design by positioning the memory networks at various locations in the decoder. Our results indicated that our design choice yields the highest overall performance. Additionally, we examined our memory component design. The experiments included sequentially stacking, parallel computation, and individual computation of the memory networks. The individual computation, which is our approach, consistently attained the highest scores across all metrics, suggesting that our design is well-justified.

Chapter 5

Conclusion

5.1 Thesis Summary

In this thesis, we tackle the issue of explainability explanation generation. The task aims to explain the decision of an unexplainable system by generating natural language explanations. Although previous work has succeeded in generating explanations, whether as an intermediate or end task, these efforts have generally involved merely fine-tuning existing models. As a result, the systems remain unexplainable black boxes.

To overcome this challenge, we adopt a weakly supervised approach and design a rule-based method to capture the underlying rationale information for NLE. Subsequently, we integrate these information into separate memory pads, which are then incorporated into a carefully engineered decoder architecture. We further equip the decoder with attention mechanisms to facilitate interactions between the decoder and our memory networks. In this way, we can train the whole model in an end-to-end fashion. This approach enables us to enhance a black-box model's explainability in NLE, and simultaneously improving performance on metrics adopted from previous work.

We conduct comprehensive experiments to evaluate our approach. Our model achieves state-of-the-art performance across different variants of the BLEU metric. Furthermore, our ablation study shows that each memory alone can improve our base-

line model, suggesting the significance of each module and demonstrating the efficacy of our modeling. We conduct additional analyses of various aspects of our approach. The analysis of our template memory size indicates that our memory is utilizing an optimal number of templates. Moreover, the case study of our factual memory displays that our decoder learns meaningful attention to the factual knowledge, thus validating our claim of improving the explainability of explanation generation. Our subsequent analyses of our architectural and memory component design confirm that our framework outperforms alternative design in terms of performance.

In conclusion, this thesis introduces model that significantly improves both the explainability and performance metrics in Natural Language Explanation. Through comprehensive experiments, ablation studies, and additional analyses, we demonstrate that our approach is well-designed.

5.2 Limitations and Future Work

Our method does not predict the sentence-level SNLI labels directly. This is not a limitation of our method, as we may easily perform multi-task learning to obtain the labels, which is also the common practice in previous work [11, 40, 51, 81]. Instead, our focus is on generating textual explanations for SNLI. Notably, many of the common practice views e-SNLI as an augmentation dataset for improving SNLI. However, their explanation generation performance may be sub-optimal (seen Figure 4.1), and may only marginally perform better than older approaches.

One potential limitation of our approach is that we rely on the previous study [76] for obtaining the factual tuples in a weakly supervised manner. However, we do not believe this affects the validity of our method because 1) EPR [76] achieve high accuracy in phrase detection and logical relation prediction, which are ready to use for downstream tasks; and 2) we are the first to show that such structured tuples are useful for textual explanation generation, which, along with our template extraction and the memory network, constitutes the focused contribution of our thesis.

We look to a few possible future directions: inducing factual and template knowledge end to end and performing explanation generation for other domains. Currently, we rely on preprocessing to obtain factual and template tuples, which means that our knowledge space is static. This may be restrictive to the explainability we can achieve. Instead, if the model can learn such tuples and present them in a dynamic manner, for example, using one-hot vectors, we would possibly accomplish even better explainability. Another meaningful future direction would be to generate explanations for many different domains, such as medical reports, legal documents, and programming languages. By extending to other domains, our system will become much more generalizable. However, we may need to first improve the methodology of generating knowledge as our knowledge retrieval methods are constrained to e-SNLI.

Bibliography

[1] J. Ainslie *et al.*, "ETC: Encoding long and structured inputs in transformers," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2020, pp. 268–284. [Online]. Available: https://aclanthology.org/2020.emnlp-main.19.

[2] M. Amirul Islam, M. Kalash, and N. D. B. Bruce, "Revisiting salient object detection: Simultaneous detection, ranking, and subitizing of multiple salient objects," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7142–7150. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018/CameraReady/2523.pdf.

[3] P. Anderson, S. Gould, and M. Johnson, "Partially-supervised image captioning," in *Advances in Neural Information Processing Systems*, 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/d2ed45a52bc0edfa11c2064e9edee8bf-Paper.pdf.

[4] V. Aribandi *et al.*, "ExT5: Towards extreme multi-task scaling for transfer learning," in *Proceedings of the International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=Vzh1BFUCiIX.

[5] N. Asghar, L. Mou, K. A. Selby, K. D. Pantasdo, P. Poupart, and X. Jiang, "Progressive memory banks for incremental domain adaptation," in *Proceedings of the International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=BkepbpNFwr.

[6] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems*, 2020, pp. 12 449–12 460. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf.

[7] A. Bibal, M. Lognoul, A. de Streel, and B. Frénay, "Legal requirements on explainability in machine learning," *Artificial Intelligence and Law*, pp. 149–169, 2021. [Online]. Available: https://doi.org/10.1007/s10506-020-09270-4.

[8] S. Black *et al.*, "GPT-NeoX-20B: An open-source autoregressive language model," in *Proceedings of the ACL Workshop on Challenges & Perspectives in Creating Large Language Models*, 2022, pp. 95–136. [Online]. Available: https://aclanthology.org/2022.bigscience-1.9/.

[9] O. Bojar *et al.*, "Findings of the 2016 Conference on Machine Translation," in *Proceedings of the First Conference on Machine Translation*, 2016, pp. 131–198. [Online]. Available: http://www.aclweb.org/anthology/W/W16/W16-2301.

[10] S. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 632–642. [Online]. Available: https://aclanthology.org/D15-1075.

[11] O.-M. Camburu, T. Rocktäschel, T. Lukasiewicz, and P. Blunsom, "e-SNLI: Natural language inference with natural language explanations," in *Advances in Neural Information Processing Systems*, 2018, pp. 9560–9572. [Online]. Available: http://papers.nips.cc/paper/8163-e-snli-natural-language-inference-with-natural-language-explanations.pdf.

[12] B. Chandra *et al.*, "Abductive commonsense reasoning," in *Proceedings of the International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=Byg1v1HKDB.

[13] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with A-GEM," in *Proceedings of the International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Hkf2_sC5FX.

[14] H. Chen, X. Chen, S. Shi, and Y. Zhang, "Generate natural language explanations for recommendation," *arXiv preprint arXiv:2101.03392*, 2021. [Online]. Available: https://arxiv.org/abs/2101.03392.

[15] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 551–561. [Online]. Available: https://aclanthology.org/D16-1053.

[16] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," in *Proceedings of the International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=r1xMH1BtvB.

[17] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 670–680. [Online]. Available: https://aclanthology.org/D17-1070.

[18] M. Cuéllar, M. Delgado, and M. Pegalajar, "An application of non-linear programming to train recurrent neural networks in time series prediction problems," in *Enterprise Information Systems VII*, 2006, pp. 95–102. [Online]. Available: https://link.springer.com/chapter/10.1007/978-1-4020-5347-4_11#citeas.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423.

[20] S. Ding *et al.*, "ERNIE-Doc: A retrospective long-document modeling transformer," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 2914–2927. [Online]. Available: https://aclanthology.org/2021.acl-long.227.

[21] B. Dzmitry, C. Kyunghyun, and B. Yoshua, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations*, 2015. [Online]. Available: https://arxiv.org/abs/1409.0473.

[22] Y. Feng, Z. Zheng, Q. Liu, M. Greenspan, and X. Zhu, "Exploring end-to-end differentiable natural logic modeling," in *Proceedings of the International Conference on Computational Linguistics*, 2020, pp. 1172–1185. [Online]. Available: https://aclanthology.org/2020.coling-main.101.

[23] J. Gao, W. Bi, X. Liu, J. Li, and S. Shi, "Generating multiple diverse responses for short-text conversation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4601.

[24] M. Ghassemi, T. Naumann, P. Schulam, A. L. Beam, and R. Ranganath, "Opportunities in machine learning for healthcare," *arXiv preprint arXiv:1412.6980*, 2018. [Online]. Available: http://arxiv.org/abs/1806.00388.

[25] A. Graves, G. Wayne, and I. Danihelka, "Neural Turing machines," *arXiv preprint arXiv: 1410.5401*, 2014. [Online]. Available: ttp://arxiv.org/abs/1410.5401.

[26] A. Gulati *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Proceedings of the Annual Conference of the International Speech Communication Association*, 2020, pp. 5036–5040. [Online]. Available: http://www.interspeech2020.org/index.php?m=content&c=index&a=show&catid=418&id=1331.

[27] A. Gupta, A. Agarwal, P. Singh, and P. Rai, "A deep generative framework for paraphrase generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11956.

[28] Y. Hao, Y. Liu, and L. Mou, "Teacher forcing recovers reward functions for text generation," in *Advances in Neural Information Processing Systems*, 2022, pp. 12 594–12 607. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/51ae7d9db3423ae96cd6afeb01529819-Paper-Conference.pdf.

[29] P. He, X. Liu, J. Gao, and W. Chen, "DeBERTa: Decoding-enhanced BERT with disentangled attention," in *Proceedings of the International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=XPZIaotutsD.

[30] S. Herdade, A. Kappeler, K. Boakye, and J. Soares, "Image captioning: Transforming objects into words," in *Advances in Neural Information Processing Systems*, 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/680390c55bbd9ce416d1d69a9ab4760d-Paper.pdf.

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, 1997. [Online]. Available: https://direct.mit.edu/neco/article-abstract/9/8/1735/6109/Long-Short-Term-Memory?redirectedFrom=fulltext.

[32] B. Huang and K. Carley, "Parameterized convolutional neural networks for aspect level sentiment classification," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 1091–1096. [Online]. Available: https://aclanthology.org/D18-1136.

[33] C. Huang, H. Zhou, O. R. Zaïane, L. Mou, and L. Li, "Non-autoregressive translation with layer-wise prediction and deep supervision," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, pp. 10 776–10 784. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/21323.

[34] J. Huang and K. C.-C. Chang, "Towards reasoning in large language models: A survey," in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023, pp. 1049–1065. [Online]. Available: https://aclanthology.org/2023.findings-acl.67.

[35] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 874–880. [Online]. Available: https://aclanthology.org/2021.eacl-main.74.

[36] D. Jin, Z. Jin, Z. Hu, O. Vechtomova, and R. Mihalcea, "Deep learning for text style transfer: A survey," *Computational Linguistics*, pp. 155–205, 2022. [Online]. Available: https://aclanthology.org/2022.cl-1.6.

[37] S. Jolly, Z. X. Zhang, A. Dengel, and L. Mou, "Search and learn: Improving semantic coverage for data-to-text generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, pp. 10 858–10 866. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/21332.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference for Learning Representations*, 2015. [Online]. Available: https://arxiv.org/abs/1412.6980.

[39] M. Komeili, K. Shuster, and J. Weston, "Internet-augmented dialogue generation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 8460–8478. [Online]. Available: https://aclanthology.org/2022.acl-long.579.

[40] S. Kumar and P. Talukdar, "NILE: Natural language inference with faithful natural language explanations," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8730–8742. [Online]. Available: https://aclanthology.org/2020.acl-main.771.

[41] T. Lei, R. Barzilay, and T. Jaakkola, "Rationalizing neural predictions," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 107–117. [Online]. Available: https://aclanthology.org/D16-1011.

[42] M. Lewis *et al.*, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880. [Online]. Available: https://aclanthology.org/2020.acl-main.703.

[43] Z. Lin *et al.*, "A structured self-attentive sentence embedding," in *Proceedings of the International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=BJC_jUqxe.

[44] X. Liu, L. Mou, H. Cui, Z. Lu, and S. Song, "Jumper: Learning when to make classification decision in reading," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2018, pp. 4237–4243. [Online]. Available: https://doi.org/10.24963/ijcai.2018/589.

[45] X. Liu, L. Mou, F. Meng, H. Zhou, J. Zhou, and S. Song, "Unsupervised paraphrasing by simulated annealing," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 302–312. [Online]. Available: https://aclanthology.org/2020.acl-main.28.

[46] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 6467–6476. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/f87522788a2be2d171666752f97ddebb-Paper.pdf.

[47] G. Luo, Y. T. Han, L. Mou, and M. Firdaus, "Prompt-based editing for text style transfer," *arXiv preprint arXiv:2301.11997*, 2023. [Online]. Available: https://arxiv.org/pdf/2301.11997.pdf.

[48] B. MacCartney and C. D. Manning, "An extended model of natural logic," in *Proceedings of the International Conference on Computational Semantics*, 2009, pp. 140–156. [Online]. Available: https://aclanthology.org/W09-3714.

[49] A. Madotto, C.-S. Wu, and P. Fung, "Mem2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 1468–1478. [Online]. Available: https://aclanthology.org/P18-1136.

[50] R. K. Mahabadi, F. Mai, and J. Henderson, "Learning entailment-based sentence embeddings from natural language inference," *Technical Report*, 2019. [Online]. Available: https://openreview.net/forum?id=BkxackSKvH.

[51] S. Narang, C. Raffel, K. Lee, A. Roberts, N. Fiedel, and K. Malkan, "WT5?! Training text-to-text models to explain their predictions," *arXiv preprint arXiv:2004.14546*, 2020. [Online]. Available: https://arxiv.org/abs/2004.14546.

[52] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2002, pp. 311–318. [Online]. Available: https://aclanthology.org/P02-1040.

[53] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2249–2255. [Online]. Available: https://aclanthology.org/D16-1244.

[54] A. Parikh *et al.*, "ToTTo: A controlled table-to-text generation dataset," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2020, pp. 1173–1186. [Online]. Available: https://aclanthology.org/2020.emnlp-main.89.

[55] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proceedings of the International Conference on Machine Learning*, 2013, pp. 1310–1318. [Online]. Available: https://arxiv.org/abs/1211.5063.

[56] M. Post, "A call for clarity in reporting BLEU scores," in *Proceedings of the Conference on Machine Translation: Research Papers*, 2018, pp. 186–191. [Online]. Available: https://aclanthology.org/W18-6319.

[57] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI Blog*, 2019. [Online]. Available: https://insightcivic.s3.us-east-1.amazonaws.com/language-models.pdf.

[58] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html.

[59] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2383–2392. [Online]. Available: https://aclanthology.org/D16-1264.

[60] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/papers/Rebuffi_iCaRL_Incremental_Classifier_CVPR_2017_paper.pdf.

[61] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, 2019, pp. 3982–3992. [Online]. Available: https://aclanthology.org/D19-1410.

[62] M. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 2016, pp. 97–101. [Online]. Available: https://aclanthology.org/N16-3020.

[63] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, 533–536, 1986. [Online]. Available: https://www.nature.com/articles/323533a0.

[64] F. Sammani, T. Mukherjee, and N. Deligiannis, "NLX-GPT: A model for natural language explanations in vision and vision-language tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8322–8332. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2022/papers/Sammani_NLX-GPT_A_Model_for_Natural_Language_Explanations_in_Vision_and_CVPR_2022_paper.pdf.

[65] S. Sanyal, H. Singh, and X. Ren, "FaiRR: Faithful and robust deductive reasoning over natural language," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 1075–1093. [Online]. Available: https://aclanthology.org/2022.acl-long.77.

[66] J. Sjöberg *et al.*, "Nonlinear black-box modeling in system identification: A unified overview," *Automatica*, pp. 1691–1724, 1995. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0005109895001208.

[67] S. Sukhbaatar, A. szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2431–2439. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/8fb21ee7a2207526da55a679f0332de2-Paper.pdf.

[68] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 3104–3112. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

[69] S. Takase and S. Kiyono, "Lessons on parameter sharing across layers in transformers," *arXiv preprint arXiv:2104.06022*, 2021. [Online]. Available: https://arxiv.org/pdf/2104.06022.pdf.

[70] A. Tikhonov, V. Shibaev, A. Nagaev, A. Nugmanova, and I. P. Yamshchikov, "Style transfer for texts: Retrain, report errors, compare with rewrites," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, 2019, pp. 3936–3945. [Online]. Available: https://aclanthology.org/D19-1406.

[71] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[72] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018, pp. 353–355. [Online]. Available: https://aclanthology.org/W18-5446.

[73] Q. Wang, T. Luo, D. Wang, and C. Xing, "Chinese song iambics generation with neural attention-based model," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2016, pp. 2943–2949. [Online]. Available: https://www.ijcai.org/Proceedings/16/Papers/418.pdf.

[74] Y. Wen, W. Zhang, R. Luo, and J. Wang, "Learning text representation using recurrent convolutional neural network with highway layers," *arXiv preprint arXiv:1606.06905*, 2016. [Online]. Available: https://arxiv.org/abs/1606.06905.

[75] J. Weston, S. Chopra, and A. Bordes, "Memory networks," in *Proceedings of the International Conference for Learning Representations*, 2015. [Online]. Available: https://arxiv.org/abs/1410.3916.

[76] Z. Wu, Z. X. Zhang, A. Naik, Z. Mei, M. Firdaus, and L. Mou, "Weakly supervised explainable phrasal reasoning with neural fuzzy logic," in *Proceedings of the International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=Hu4r-dedqR0.

[77] J. Wuebker, S. Green, J. DeNero, S. Hasan, and M.-T. Luong, "Models and inference for prefix-constrained machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 66–75. [Online]. Available: https://aclanthology.org/P16-1007.

[78] J. Xu, X. Sun, Z. Zhang, G. Zhao, and J. Lin, "Understanding and improving layer normalization," in *Advances in Neural Information Processing Systems*, 2019, pp. 4381–4391. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/2f4fe03d77724a7217006e5d16728874-Paper.pdf.

[79] J. Yin, X. Jiang, Z. Lu, L. Shang, H. Li, and X. Li, "Neural generative question answering," in *Proceedings of the Workshop on Human-Computer Question Answering*, 2016, pp. 36–42. [Online]. Available: https://aclanthology.org/W16-0106.

[80] T. Zhao, R. Zhao, and M. Eskenazi, "Learning discourse-level diversity for neural dialog models using conditional variational autoencoders," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 654–664. [Online]. Available: https://aclanthology.org/P17-1061.

[81] X. Zhao and V. Vydiswaran, "LIREx: Augmenting language inference with relevant explanations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 14 532–14 539. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17708.

[82] W. Zijun, "Neural fuzzy logic reasoning for natural language inference," M.S. thesis, University of Alberta, 2022. [Online]. Available: https://era.library.ualberta.ca/items/a61e440f-6f96-4f00-879d-5bf0f3da8baf.