Social Commonsense Reasoning with Structured Knowledge in Text

Debjit Paul

Department of Computational Linguistics
Heidelberg University

This dissertation is submitted for the degree of
Doctor of Philosophy

Supervisor: Prof. Dr. Anette Frank
Second supervisor: Prof. Dr. Michael Strube

Submission date: 02.03.2022.

Acknowledgements

I am fortunate to have worked under the guidance of my "Doktormutter", Anette Frank who has helped me throughout my PhD with her valuable advises, feedback and constant support. Not only did her professional ideas percolate in my work, but I will also be forever influenced by her leadership qualities. I am grateful to her for giving me the opportunity to work with her.

I am also indebted to other professors who have offered me guidance and advice throughout my research career. In particular, I want to thank Dietrich Klakow, Michael Strube and Graeme Hirst.

I would like to thank Angel Daza, Russa Biswas, Sreyasi Nag Chowdhury and Letiția Pârcălăbescu for going out of their way to help me in improving the manuscript.

I had often underestimated the importance of friendships, but looking back at my PhD journey I realise how my friends were a balm to my soul. I am thankful for the extended weekend getaways with my closest friends in Europe, the late night musings during stay-overs. I would like to thank my current and former lab members and friends : Ana Marasovic, Maria Becker, Esther van den Berg, Bhushan Kotnis, Bich-Ngoc Do, Juri Opitz, and Letiția Pârcălăbescu. I really appreciate the intellectual discussions, the coffee-table conversations and beyond.

I would like to thank Todor Mihaylov, who has been so helpful throughout my PhD, answering questions and making suggestions about my research, internship and much more. I owe a huge thanks to Éva Mújdricza-Maydt, who has helped me with everything, from accessing clusters to setting up annotations for my first experiment. Also, I would like to thank Angel Daza, for his patience to listen and his wise suggestions related to my research. Thank you for making my PhD experience run so smoothly!

And lastly, a special thanks to my wife, Manjima Bardhan for her companionship, support and endless conversations that made my life complete. I cannot thank enough those who I tend to take for granted. . . to those who have tamed me the most. . . my constants – my parents, my wife, all the family back home and my childhood friends – for contributing in ways known and unknown towards the person I have become.

Abstract

Understanding a social situation requires the ability to reason about the underlying emotions and behaviour of others. For example, when we read a *personal story*, we use our prior commonsense knowledge and social intelligence to infer the emotions, motives, and anticipate the actions of the characters in a story. For machines to understand text related to *personal stories and social conversations*, they must be able to make commonsense inferences. While most people can reason deeply about the social implications of the text, it is challenging for natural language processing systems as these implications are often subtle and implicit.

This dissertation argues that NLP systems must learn to reason more explicitly about the underlying social knowledge in text to perform social commonsense reasoning. We divide the above argument into two sub-problems: (i) understanding the underlying social knowledge and (ii) explicitly reasoning about such knowledge for social commonsense reasoning. To address these problems, we propose building NLP systems that integrate neural network-based learning with structured knowledge representations.

In the first part of this dissertation, we study the role of structured commonsense knowledge in understanding the social dynamics of characters and their actions in stories. Our motivation behind enriching the model with structured commonsense knowledge is to bridge the gap between the surface meanings of texts and the underlying social implications of each event in the stories. We develop a novel model that incorporates commonsense knowledge into neural models and showcases the importance of commonsense knowledge in understanding the social dynamics of story characters. Further, we investigate the role of temporal dynamics of story events in understanding social situations. We develop a model that can explicitly learn about *what social event follows another event* from personal narrative stories. We demonstrate that *implicitly* leveraging such temporal knowledge about story events can support social commonsense reasoning tasks.

In the second part of this dissertation, we investigate methods to explicitly reason about the knowledge related to social dynamics of characters (*behaviour, mental states*) and the cause/effect of social events. We propose a novel model named as *multi-head knowledge attention* that incorporates such social knowledge into state-of-the-art neural NLP models to address two complex commonsense inference tasks. We demonstrate that our method

of incorporating knowledge can improve – (i) the robustness and the interpretability of the model and (ii) the overall performance of the model compared to other knowledge integration methods. We also aim to investigate social commonsense reasoning as a natural language generation task. We design a story completion task that requires natural language generation models to perform both forward and backward reasoning. We study the role of contextualized commonsense knowledge in natural language generation tasks. We propose a model that jointly learns to generate contextualized inference rules as well as narrative stories. We demonstrate that our model can outperform state-of-the-art non-contextualized commonsense knowledge-based generation models.

We hope that the research presented in this dissertation will open up interesting scopes for future research involving social commonsense reasoning and other related topics.

Table of Contents

Chapter 1

Introduction

"Reasoning is here only to wag the dog, to create post-hoc justification that cover the tracks of the intuitions and emotions secretly running the show."
– Jonathan Haidt

1.1 Motivation

Humans can effortlessly understand natural language text about everyday situations by relying on commonsense knowledge and making inferences. For example, given a straightforward narrative, "My friend is upset with me. I will have to go to a gift shop." consists of two sentences that are supposedly unrelated to one another while we can understand the underlying conditions that make them related. As a first step, we identify the implicit commonsense knowledge relevant in the context, for example, "my friend might like gifts", "gifts will make my friend happy". Then this commonsense knowledge can be utilized to reason about the above context, such as:

(1) ((my friend is upset $\Rightarrow$ want to make my friend happy) $\wedge$ (buying gifts for friend $\Rightarrow$ will make my friend happy) $\wedge$ my friend might like gifts) $\Rightarrow$ have to go to a gift shop

Research on commonsense reasoning has primarily focused on physical commonsense reasoning and hence, focused on building resources such as knowledge graphs that cover physical or taxonomic knowledge (Lenat and Guha, 1989; Miller, 1995; Tandon et al., 2017), and related datasets (Winograd, 1972; Weston et al., 2016). In this dissertation, we will focus instead on *Social Commonsense Reasoning (SCR)*, that requires the ability to infer mental states (*human needs, motives*, *emotional reactions*), and deeper social implications (*causes*

and effects of social events). There are two key challenges in making social commonsense inferences: (i) usually they are subtly implied in the text, and (ii) the inferences are stochastic in nature and can be defeasible[1] with additional context. For example, when we add to the context that *"I lost all contacts with my friend"* the above conclusion (deduction) does not hold.

In recent years, natural language processing (NLP) systems have been widely used in society for different real-world applications such as interactive virtual assistants (e.g., Amazon Alexa), writing assistants (e.g., GMail's or Grammarly's writing assistant), etc. Their effectiveness relies on their ability to understand and reason about the social dynamics[2] and social commonsense (Pereira et al., 2016; Gunning, 2018). For instance, if a human user asks a virtual assistant, "Alex's friend is upset. What should Alex do next?", we would like a typical assistant to understand the user's *mental state* and suggest Alex to *call his friend or buy him a gift*. Similarly, if a user says something a bit more complex such as "Alex spilt his friend's food all over the floor, making a huge mess. How will his friend feel? What will Alex do next?", we would like systems be able to perform the required multi-hop reasoning steps and suggest the user to *apologize* and *mop up* (Sap et al., 2019b). Recently, there have been AI assistants that are designed as therapeutic counselling systems to assist people with cognitive disabilities (e.g., Woebot, Youper, Wysa, Cocobot), which also require social commonsense reasoning abilities in order to operate more effectively (Pollack, 2005; Graham et al., 2019).

The ability to reason about what others think or believe (also known as *"theory of mind"*) is an important component for reasoning about mental states (human needs, motives, emotions) (Premack and Woodruff, 1978; Moore, 2013; Gordon, 2019) and for daily communication (Apperly, 2010). Humans are good at such reasoning tasks as we rely on our prior knowledge and learned experiences through interaction with the world. On the other hand, machines do not possess such prior knowledge and interaction-based experiences. Instead, they must be provided with this knowledge directly or find alternative ways of learning it. This dissertation presents novel work that integrates social commonsense knowledge into current NLP systems to improve their social commonsense reasoning capabilities. Achieving human-level proficiency at commonsense reasoning tasks is considered "AI-complete" (i.e., solving commonsense reasoning requires human intelligence). This thesis makes an ambitious step towards filling the gap between current NLP systems and human reasoning capabilities centred around social commonsense reasoning.

[1] The reasoning is defeasible when the inference is rationally compelling but not deductively valid (Koons, 2021).

[2] Social dynamics is a form of reasoning that involves interactions between individuals, their mental states, and how this impacts their actions and behaviour (Rashkin, 2020)

Previous approaches for endowing natural language processing systems with the ability to perform commonsense reasoning have mostly focused on creating knowledge graphs (KGs) (Speer et al., 2017; Tandon et al., 2017; Cambria et al., 2020) or building neural models that can learn relevant knowledge implicitly from training on large-scale data (Goldberg, 2016). Constructing commonsense knowledge graphs is vital for commonsense reasoning; however, manual construction is expensive and time-consuming. Further, Valiant (2008) argued that a purely symbolic approach would be insufficient for successful reasoning with commonsense knowledge. More recently, large-scale pretrained language models (Peters et al., 2018; Radford et al., 2018, 2019; Devlin et al., 2019; Liu et al., 2019) have shown to capture implicit knowledge from a large text corpus (Petroni et al., 2019) and also outstanding performance on several NLU tasks. However, although these models adapt to various NLP tasks, their behaviour and reasoning capabilities remain opaque.

Neuro-symbolic AI (Besold et al., 2017; Mao et al., 2019; Garcez and Lamb, 2020), on the other hand, uses both deep neural network architectures and symbolic reasoning techniques. It aims to leverage the strengths of each approach: the structure, interpretability and readability of symbolic representations and the expressivity and connectionism of neural networks (Besold et al., 2017). In this thesis, we view knowledge graphs (KGs) as discrete symbolic representations of entities and their relations (Xiao et al., 2016; Ji et al., 2021; Hwang et al., 2021). We will focus on developing methods that combine structured knowledge and neural representations to address social commonsense reasoning tasks. We study the performance of our models on understanding social dynamics and social commonsense reasoning tasks[3] :(i) mental states prediction (Rashkin et al., 2018a), (ii) abductive commonsense reasoning (Bhagavatula et al., 2020), (iii) counterfactual invariance prediction task (Paul and Frank, 2020), and (iv) narrative story completion task (Paul and Frank, 2021a).

1.2 Research Questions

* **The role of commonsense knowledge in understanding social dynamics:** Various studies have recently shown the importance of external commonsense knowledge on different Natural Language Understanding (NLU) tasks (Bian et al., 2021). While the role of structured and unstructured external knowledge is well-established for popular NLU tasks like Question Answering, Reading Comprehension, etc., their usefulness in identifying social commonsense knowledge such as people's mental states in a social

[3]Details about these social dynamics and social commonsense reasoning tasks are in section 4.1, 2.1, 5.1 and 6.1.1.

context is still unclear. We hypothesize that *implicit knowledge is crucial for a better understanding of social dynamics*. This leads us to the following research questions:

RQ1(a) How can we develop a method to automatically predict people's mental states and how do these mental states alter with changes of social situations?

RQ1(b) What is the role of external structured commonsense knowledge in identifying mental states?

While a large knowledge graph is important for neural-symbolic AI research, Garcez and Lamb (2020) argue that a large knowledge graph is not more explainable than a large neural network. Moreover, large commonsense knowledge graphs contain unnecessary information; hence, extracting relevant knowledge[4] is an essential part of neural-symbolic AI and a significant ingredient for explaining black-box AI systems. Meanwhile, in the last few decades, several graph-based algorithms have been developed to estimate the importance of a node in a particular graph (Page et al., 1999; Borgatti, 2005). However, no prior research explores graph connectivity or graph structure to identify relevant knowledge from large KGs for understanding social dynamics. This inspires the following research question:

RQ1(c) How can we use graph-based algorithms to extract relevant knowledge (grounded in social situations) from large static commonsense knowledge graphs?

* **The role of temporal knowledge in understanding social dynamics**: Recently there are various works which focus on learning temporal knowledge[5] to order sequence of events (Chambers and Jurafsky, 2008a; McDowell et al., 2017; Madaan and Yang, 2021; Lin et al., 2021; Ghosal et al., 2021). Several prior studies have focused on temporal relation extraction, which orders pairs of events in text (Mani et al., 2006; Chambers et al., 2007; Han et al., 2019). There are also work which focused on building schema learning systems to automatically learn about related eventsm a temporal ordering of events (Chambers and Jurafsky, 2008b, 2010). However, the importance of learning about what events could follow other events for social commonsense reasoning is an understudied problem. Abductive reasoning is a backward reasoning task, which typically requires models to reason about past events. We hypothesize that *learning about temporal order of social events can support abductive reasoning*. This stimulates the following research questions:

[4]Here by *relevant knowledge*, we mean a machine readable knowledge that is correct, complete and can be used to explain a context.

[5]Here by temporal knowledge we mean knowledge about what event precedence another event

RQ2(a) How can we design a method that can learn what event could follow another event in a social context?

RQ2(b) Can we make use of temporal knowledge when performing abductive reasoning?

* **Integrating dynamic social commonsense knowledge into neural networks**: Once we have established methods for extracting relevant social commonsense knowledge from static knowledge graphs or narrative stories, we can explore how much such knowledge impacts downstream social commonsense reasoning tasks. However, one crucial bottleneck is that we need methods to integrate such knowledge into neural models. This leads us to the following research questions:

  RQ3(a) Can we find suitable methods for integrating different kinds of extracted knowledge into state-of-the-art machine learning models, in order to enhance interpretability and robustness?

  RQ3(b) To what extent does the integration of knowledge affect the performance of the model in the targeted downstream social commonsense reasoning tasks : *abductive reasoning and counterfactual invariance prediction*)?

* **Dual learning:** Grounding inferential knowledge is essential for interpreting and applying the knowledge in context. Moreover, by contextualizing the knowledge we can address the problem of disambiguation in commonsense inference. Previous works considered non-contextualized knowledge for their task settings (Guan et al., 2020; Ji et al., 2020; Yu et al., 2022). Additionally, they designed the extraction of relevant knowledge and its integration in a neural network as two separate steps. Our final set of research questions challenges this practice by asking are as follows:

  RQ4(a) How can we design a model framework that jointly learns to generate contextualized relevant knowledge and uses it in order to guide the generation of task-specific textual outputs e.g., when perform narrative story completion tasks?

  RQ4(b) Does jointly learning to generate relevant knowledge and perform downstream task helps the model to be more transparent?

1.3 Thesis Outline & Contributions

In the remainder of this thesis, we will first provide the background on several fundamental concepts and techniques that we need to achieve our proposed contributions. In **Chapter**

2 we start by describing some of the important work on commonsense reasoning in NLP (Section 2.1) and commonsense knowledge acquisition (Section 2.2). We describe some basic concepts and techniques of neural network models that are used in natural language processing and that will be relevant for this thesis (Section 2.4). **Chapter 3** reviews the literature related to the above research questions. We survey some of the most prominent knowledge extraction and integration approaches (Sections 3.1 and 3.2).

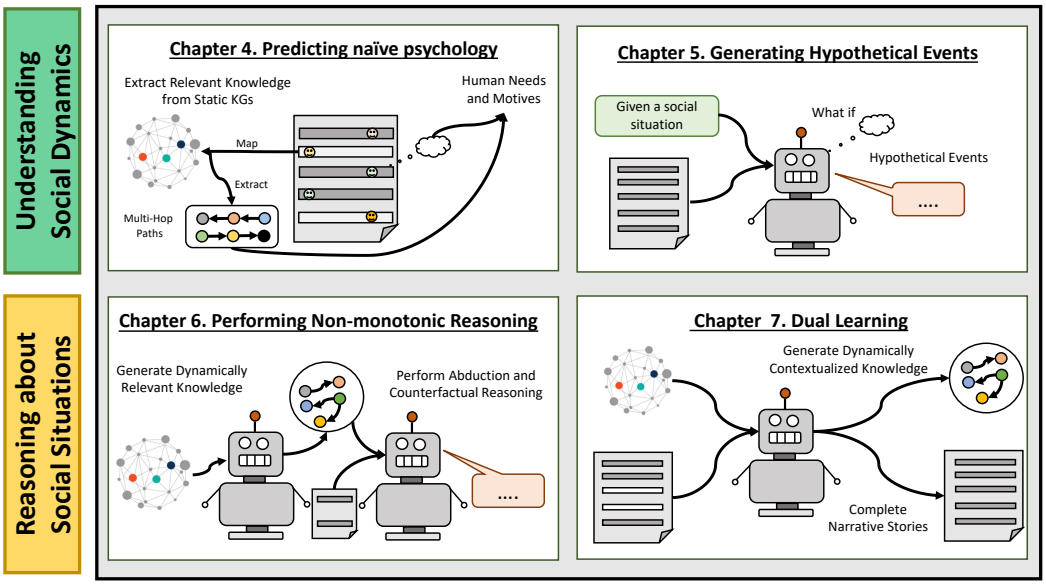


Fig. 1.1 Overview of our approach: This thesis covers four tasks centered around social commonsense reasoning.

We address the above research questions in a multi-faceted approach for studying social commonsense reasoning in text by investigating four tasks described in Figure 1.1. These tasks can be grouped into two themes: (i) understanding social dynamics and (ii) reasoning about social situation. In the first, we develop methods that can make inferences about the social dynamics of story characters (mental states prediction) and story events (temporal knowledge about events). In the second, we investigate how such knowledge related to mental states and cause/effect of events can contribute to reasoning about social situations. To this end, the contributions of this thesis can be summarized as follows:

- **Understanding Social Dynamics:**
  - In **Chapter 4**, we address the task of automatically predict mental states of story characters given a story context. We first address research question RQ1(a) by

presenting an end-to-end LSTM-based model enhanced with attention and a gated knowledge integration component to predict mental states in a given context. This model provides interpretability in two ways by selecting relevant words from the input text and choosing relevant knowledge paths from the imported knowledge. In both cases, the degree of relevance is indicated via an attention map in both cases. Next, to address RQ1(c), we propose a novel approach to extract and rank multi-hop relation paths from a commonsense knowledge resource using graph-based features and algorithms. Finally, we address RQ1(b) by investigating how well commonsense knowledge obtained from a specific commonsense resource can help NLP models in understanding the mental state of characters based on story events. To this end, we also conduct a small-scale human evaluation to study the relevance of the commonsense knowledge paths in classifying mental states.

- In **Chapter 5**, we address (*generating next events*) RQ2(a), where we investigate different fine-tuning strategies to learn what events could follow other events in a social situation. Next to address RQ2(b), we present a novel method for addressing the abductive reasoning task by explicitly learning temporal knowledge. We conduct human evaluation and show that learning about temporal knowledge can support abductive reasoning in both an unsupervised and a supervised setting.

- **Reasoning about Social Situations:**

  - In **Chapter 6**, to address (*social commonsense reasoning*) RQ3(a) we explore ways to integrate commonsense knowledge into state-of-the-art (SOTA) transformer-based models. We propose a multi-head knowledge attention model that encodes semi-structured commonsense knowledge rules and learns to incorporate them into transformer-based models. Next to address RQ3(b), we introduce a novel counterfactual invariance prediction (CIP) task and show a correlation between abduction and counterfactual reasoning in a narrative context. We analyze the reasoning capabilities of our model and perform model inspection using manually validated knowledge rules. Finally, we show that our knowledge enhanced model is more robust than other SOTA models by perturbing and adding noise to the knowledge.
  - In **Chapter 7**, to address (*generate contextualized inferences*) RQ4(a) we investigate how NLP models can learn to *generate commonsense inference* knowledge grounded in a context and to perform downstream reasoning, using the generated inferences as a guide. We focused on the narrative story completion (Natural

Language Generation (NLG)) task. We propose a model named COINS that recursively performs an inference step (*generating inferential knowledge*) and a generation step (*generating next story sentence*). Further to address RQ4(b), we demonstrate that the recursive nature of our model and the individuation of the inference prediction and sentence generation tasks make the process more interpretable. The generated inference rules can be viewed as intermediate representations and can serve as explanations of how the dynamically produced inferences influence the quality of generated story sentences.

Lastly, in **Chapter 8** we summarize our findings, discuss limitations of our work and propose potential future directions of research.

1.4 Published Work

The following published papers are included in the text of this dissertation, listed in the order of their appearance:

- Paul, D. and Frank, A. (2019). Ranking and Selecting Multi-Hop Knowledge Paths to Better Predict Human Needs. *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT).*
- Paul, D., Opitz, J., Becker, M., Kobbe, J., Hirst, G., and Frank, A. (2020). Argumentative Relation Classification with Background Knowledge. *Proceedings of the 8th International Conference on Computational Models of Argument (COMMA 2020), vol. 326 of Frontiers in Artificial Intelligence and Applications.*
- Paul, D. and Frank, A. (2020). Social Commonsense Reasoning with Multi-Head Knowledge Attention. *In Findings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).*
- Paul, D. and Frank, A. (2021). COINS: Dynamically Generating COntextualized Inference Rules for Narrative Story Completion. *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021).*
- Paul, D. and Frank, A. (2021). Generating Hypothetical Events for Abductive Inference. *Proceedings of The Tenth Joint Conference on Lexical and Computational Semantics (*SEM 2021).*

Chapters 4 and 5 have been published as (Paul and Frank, 2019; Paul et al., 2020) and (Paul and Frank, 2021b) respectively. Chapters 6 and 7 have been published as (Paul and Frank, 2020) and (Paul and Frank, 2021a).

Chapter 2

Background

"If I have seen further than others, it is by standing upon the shoulders of giants."

–Issac Newton

The concepts and techniques that are presented in this chapter will provide the basis for understanding the work in subsequent chapters of this dissertation. We start by introducing various research on commonsense reasoning and commonsense knowledge in Natural Language Processing (sections 2.1, 2.2). We review some popular symbolic approaches (section 2.3) and describe some fundamental concepts and methods related to Deep Learning in NLP (section 2.4).

2.1 Commonsense Reasoning

Commonsense Reasoning or Commonsense thought is the underlying reasoning process (ability) that allows humans to connect pieces of implicit knowledge to reach a new conclusion (Minsky, 2000; Davis and Marcus, 2015). For example, consider the following sentences: *"Peter was thirsty, so he went and shook his water bottle. He was disappointed when it made no sound."* From these sentences, we can easily infer that *there was no water in the bottle*, and because of that, *Peter was sad*. We use our knowledge about the world and connect them to reach a conclusion that is not explicitly stated. John McCarthy, one of the founding fathers of AI, was amongst the first to realize the importance of commonsense reasoning. McCarthy (1960) proposed commonsense reasoning through a hypothetical program named as ADVICE TAKER for solving problems by manipulating sentences in formal languages. Although McCarthy (1960) only suggested key specifications for commonsense programs, later, with the boom of AI, its importance became more evident.

2.1.1 Human Cognitive System

Daniel Kahneman, in his Nobel Prize lecture "Maps of Bounded Rationality" (Kahneman, 2003) presented a human cognitive system. Figure 2.1 depicts three cognitive operations: *perception, intuition,* and *reasoning*. *Perception* requires us to detect the surface level patterns and helps us in observing the world and in generating impressions. Examples of *perception* operations are object detection, image recognition, machine translation, and automatic speech recognition. The operations of *intuition* require intuitive inferences that

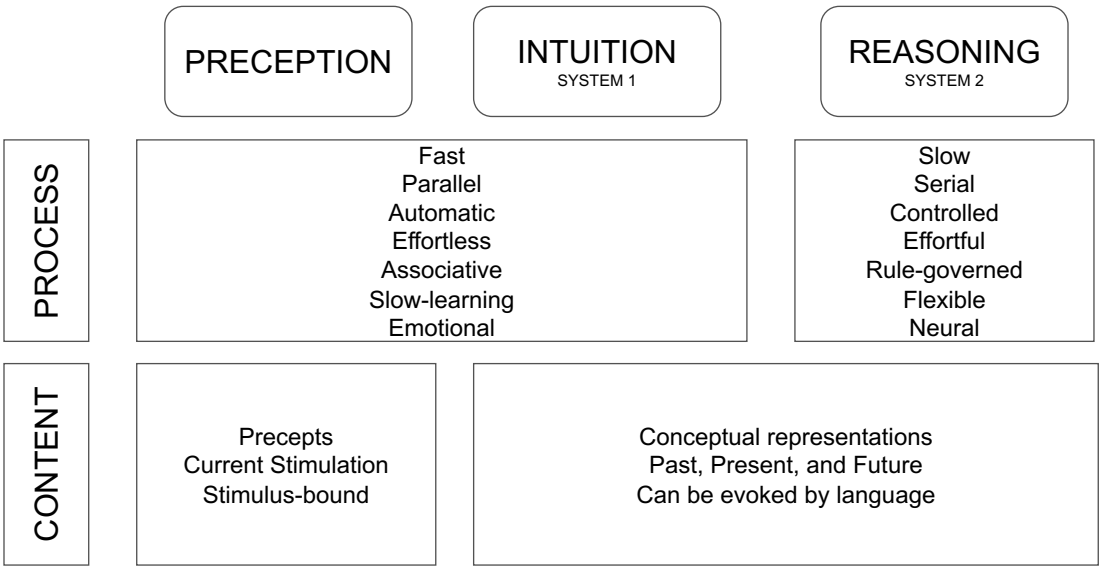


Fig. 2.1 Daniel Kahneman's three cognitive functions (Kahneman, 2003)

are fast and effortless, based on impressions. Humans perform these operations every moment of their life. Examples of such operations are understanding other person's emotions, predicting what happens before or after certain events, and reasoning about motivations and intentions. Finally, *Reasoning* involves deliberate thought processes and judgements that are slow and require effort and only happen when humans intentionally make them (deliberately controlled). Examples of such operations are writing PhD theses, solving puzzles, and writing paper reviews. The key difference between *intuition* and *reasoning* is the *accessibility* – the ease with which any particular mental contents come to mind (Kahneman, 2003; Higgins, 1996). For example, the multiplication problem 19 times 53 often takes a few moments to perform the lengthy act of computation before we arrive at its answer, 1007. On the contrary, when we read a sentence *"I broke my leg"*, we immediately infer that *"it must hurt"*, it

seems the inference is accessible, almost as an instantaneous extension of our thinking. Most intuitive inferences are commonsense inferences. In this thesis, we focus on intuitive commonsense reasoning over natural language text.

2.1.2 Commonsense Reasoning in Natural Language Processing

The research in Natural Language Processing has made much progress in lexical and syntactic tasks. However, state-of-the-art NLP models struggle to tackle pragmatic level tasks such as reference resolution, question answering, text generation, and dialogue. Moreover, these tasks often require a deeper understanding of natural languages and systems to reason over commonsense knowledge (Mihaylov and Frank, 2018a). Endowing machines with commonsense reasoning capabilities has remained an elusive goal of natural language research for decades (Bar-Hillel, 1960a). One of the primary reasons is that commonsense knowledge is implicit. From the above example in section 2.1 we could see *there was no water in the bottle* is implicit in the text.

In NLP, there has been a surge in the variety of benchmarks to evaluate and analyze the performance of models on commonsense reasoning. Some of the notable tasks and benchmarks are described as follows:

- **Reference Resolution** : The task of reference resolution is to identify a referent, typically a linguistic expression, that a particular entity (e.g., a pronoun or phrase) refers to in a span of text. Reference resolution can be significantly complicated due to ambiguities which arise when multiple entities are present in a sentence or discourse, originating a need for external knowledge, e.g., commonsense knowledge. Winograd Schema Challenge (Winograd, 1972; Levesque et al., 2012) is one such example, which was proposed in the spirit of the Turing test, to evaluate commonsense reasoning capabilities by resolving co-references for ambiguous pronouns. Interestingly later Trichelair et al. (2019) found biases in the original Winograd Schema Challenge. To reduce human bias (Sakaguchi et al., 2020) introduced a more challenging and large-scale version called Winogrande. The dataset provides an adversarial version of the problem and focuses on intuitive physics and psychology.

- **Question Answering (QA)**: Unlike reference resolution, QA focuses more on a comprehensive mix of language processing and reasoning skills. Recently, there has been a focus on creating QA benchmark datasets that require external knowledge and commonsense reasoning. OpenBookQA (Mihaylov et al., 2018) is one such dataset that focuses on the challenge of combining a corpus of provided science facts (open book) with external broad common knowledge. Talmor et al. (2019) presented a dataset

Task	Context	Alternatives
COPA (Roemmele et al., 2011)	Premise: The man broke his toe. What was the CAUSE of this?	1. He got a hole in his sock. **2. He dropped a hammer on his foot.**
α*NLI* (Bhagavatula et al., 2020)	O_1 : It was a very hot summer day. O_2 : He felt much better!	H_1: He decided to run in the heat. H_2: **He drank a glass of ice cold water.**
SocialIQA (Sap et al., 2020a)	Alex spilled the food she just prepared all over the floor and it made a huge mess. What will Alex want to do next?	1. taste the food **2. mop up** 3. run around in the mess
SC (Rashkin et al., 2018a)	Cindy really likes apples. She wanted to try something new with them.	Motivation (Reiss): **Curiosity** Emotion (Plutchik): **Joy, Anticipation**

Table 2.1 Examples from benchmarks requiring plausible inference and intuitive psychology. The correct choice in each example is given in bold text.

named CommonsenseQA in which each question requires commonsense knowledge to disambiguate a target concept from three related concepts in ConceptNet (Speer et al., 2017). There are also research work which focuses on specific aspects of commonsense reasoning such *Contextual* commonsense reasoning (Cosmos QA) (Huang et al., 2019), *Temporal* commonsense understanding (MC-TACO) (Zhou et al., 2019), *Physical* commonsense reasoning (PIQA) (Bisk et al., 2020), *Prototypical* commonsense reasoning (ProtoQA) (Boratko et al., 2020), and *Social Intelligence* commonsense reasoning (SocialIQA) (Sap et al., 2019b).

- **Plausible Inference**: One of the challenging aspects of commonsense reasoning is that it involves plausible reasoning. Therefore, it requires models to arrive at a reasonable conclusion given what is already known (Peirce, 1883; Hobbs et al., 1993a). Performing plausible reasoning requires reasoning over linguistic context and external knowledge (Davis and Marcus, 2015). Roemmele et al. (2011) proposed a dataset named Choice of Plausible Alternatives (COPA) to evaluate the model's performance on the causal reasoning between events, such that it requires commonsense knowledge about what usually takes place in the world. Each example provides a premise and either ask for the correct cause or effect from two choices, thus testing backwards or forward causal reasoning. Table 2.1 shows one example from Roemmele et al. (2011). Story Cloze Test was proposed by (Mostafazadeh et al., 2016) to test the model's capabilities in selecting the correct ending to a four-sentence story. The dataset addresses the challenge of understanding causal and correlational relationships between events.

Intuitive psychology requires inference of emotions, intentions and other observable psychological states through human's behaviour. A significant domain in plausible inference tasks is intuitive psychology (Gordon, 2016a). Rashkin et al. (2018a) annotated 15k stories from ROC Stories dataset (Mostafazadeh et al., 2016) with motivations and emotions of characters in each story to enable more concrete reasoning in this area. This dataset known as Story Commonsense (SC), consists of three classification tasks for inferring: the *basic human needs* theorized by Maslow (1943), *human motives* theorized by Reiss (2004), and *human emotions* theorized by Plutchik (1980). Sap et al. (2019b) presented Social Intelligence QA (SocialIQA) that tests a model's ability to perform intuitive psychology and commonsense knowledge of social interactions. In Table 2.1, we can see one such example where intuitive social and psychological commonsense knowledge is require to answer the plausible reactions (*want*) of *"Alex"* after *"he made a huge mess"*. Recently, Bhagavatula et al. (2020) proposed a dataset named α*NLI*, which addresses the abductive reasoning task where given two observations as an incomplete context, the model needs to predict which of two hypothesized events is more plausible to have happened between the observations. Table 2.1 depicts one example from αNLI dataset.

In this dissertation, we focus on several plausible inference tasks, which are as follows:

1. In **Chapter 3**, we address the task of classifying human need categories of characters in narrative stories from two inventories: Maslow's (Maslow, 1943) (with five coarse-grained) and Reiss's (Reiss, 2004) (with 19 fine-grained) categories.

2. In **Chapter 4, 5**, we investigate the impact of intuitive social and psychological commonsense knowledge on downstream plausible inference tasks like abductive commonsense reasoning (Bhagavatula et al., 2020), and counterfactual invariance prediction task (CIP) (Paul and Frank, 2020). In (Paul and Frank, 2020), we introduce a new task of counterfactual invariance prediction [1].

3. In **Chapter 6** unlike most of the above mentioned existing benchmark datasets, which treat commonsense reasoning as a deterministic task, we explore commonsense reasoning as a natural language generation task. Finally, we present a new task setting named Narrative Story Completion (NSC) (Paul and Frank, 2021a) to test the model's ability to generate intuitive social commonsense knowledge and perform story completion.

In the human cognitive system, there is a need for an information system to maintain knowledge representations (Kahneman, 2003). Humans learn knowledge and apply it in

[1] More details about counterfactual invariance prediction task (CIP) is in section 6.1

intuition and reasoning to make better judgments. Hence, commonsense knowledge is essential for performing intuitive reasoning and deliberate reasoning. In the next section, we will introduce existing knowledge resources and several recent efforts in building such resources to facilitate commonsense reasoning.

2.2 Commonsense Knowledge

Commonsense Knowledge (CSK) is routine knowledge that humans typically possess that helps them make sense of daily situations (Ilievski et al., 2021). Although this knowledge is assumed to be possessed by most humans, according to Gricean maxims, it is usually omitted in (written or oral) communication (Grandy and Warner, 2020). Similarly, (Gordon and Van Durme, 2013) showed that words like *"murdered", "laughed"* are observed five times more than *"inhaled", "breathed"* in a large text corpus (Google Web 1T n-gram data) even though breath or inhale are more predominant actions. Since it is not explicitly stated, automatically learning commonsense knowledge from text presents a challenge for natural language processing systems.

Commonsense knowledge can be broadly categorize into two categories : *"naive physics"* and *"intuitive psychology"*. Commonsense knowledge related to *"naive physics"* involves inference of how physical objects interact with each other. For example, if a glass falls onto the floor, one can infer that the glass most likely break. On the other hand, commonsense related to *"intuitive psychology"* involves inference about people's behaviors, intents, or emotions. For example, if a person breaks their leg, one can infer that their will be sad. Due to its prominence and implicit nature, there has been a lot of work on constructing commonsense knowledge resources in a machine-readable form that includes ConceptNet OpenCyc (Lenat and Guha, 1989), WebChild (Tandon et al., 2017), (Speer et al., 2017), Event2Mind (Rashkin et al., 2018c), ATOMIC (Sap et al., 2019a), GLUCOSE (Mostafazadeh et al., 2020a), ASER (Zhang et al., 2020b), SenticNet (Cambria et al., 2020), etc. In this thesis we mainly focused on knowledge related to the intuitive psychology of human beings (e.g., emotion states, intents, possible behaviors), that emphasizes on the social intelligence found in daily human-human interactions. Some of the most prominent commonsense knowledge resources that contain knowledge about intuitive psychology are summarized in Table 2.2 and there descriptions are as follows:

- **OpenCyc (1984-2012) (Lenat and Guha, 1989)** It is an artificial intelligence project toward integrating ontologies and commonsense knowledge from different domains into one knowledge base. The objects in Cyc are called as *constants* and categorized into entities, collections, functions, and truth functions. Cyc includes a powerful

Knowledge Graphs	Size	Relation	Annotation
OpenCyc (Lenat and Guha, 1989)	239,000 concepts, 2,039,000 facts	–	Manual
ConceptNet (Speer et al., 2017)	8 million nodes, 21 million edges	34	Crowd-sourcing
WebChild (Tandon et al., 2017)	2 million concepts, 18 million assertions	4 (groups)	Automatic
Event2Mind (Rashkin et al., 2018c)	24,716 events, 57,097 edges	3	Crowd-sourcing
ATOMIC (Sap et al., 2019a)	877,108 triples	9	Crowd-sourcing
ASER (Zhang et al., 2020a)	194,000,677 nodes, 64,351,959 edges	15	Automatic
GLUCOSE(Mostafazadeh et al., 2020a)	670,000 pairs of rules	10	Crowd-sourcing

Table 2.2 Overview of some commonsense knowledge bases. Source: Ilievski et al. (2021)

inference engine, which is capable of performing general logical deduction. The latest release (OpenCyc 4.0) contains 239,000 concepts and 2,039,000 facts.

- **ConceptNet(1999-2017) (Speer et al., 2017)** is a multilingual commonsense knowledge graph derived from the Open Mind Common Sense (OMCS) (Singh et al., 2002). It is a directed graph whose nodes are concepts, and the edges are relations which connect the concepts, e.g., *"is a", "used for", "motivated by goal"*, etc. The nodes are natural language phrases, e.g., noun phrases, verb phrases, or clauses and there are 34 different relation types. ConceptNet 5.5 is the latest version which contains over 8 million nodes and over 21 million edges. Figure 2.2(a) depicts an example of entities and relations extracted from ConceptNet that are related to the concept of *glass*.

- **WebChild (2014–2017) (Tandon et al., 2017)** is a large-scale commonsense knowledge base of general noun-adjective relations extracted from Web contents, using semi-supervised label propagation over graphs of noisy candidate assertions. It consists of about 78,000 distinct noun senses, 5,600 distinct adjective senses, and 4.6 million assertions between them. These assertions captured fine-grained relations among the noun and adjective senses. The knowledge base contains fine-grained relationships (like "hasShape", "hasTaste", "evokesEmotion", etc.) between nouns and adjectives. WebChild 2.0 was released in 2017 and includes over 2 million concepts and activities, and over 18 million assertions.

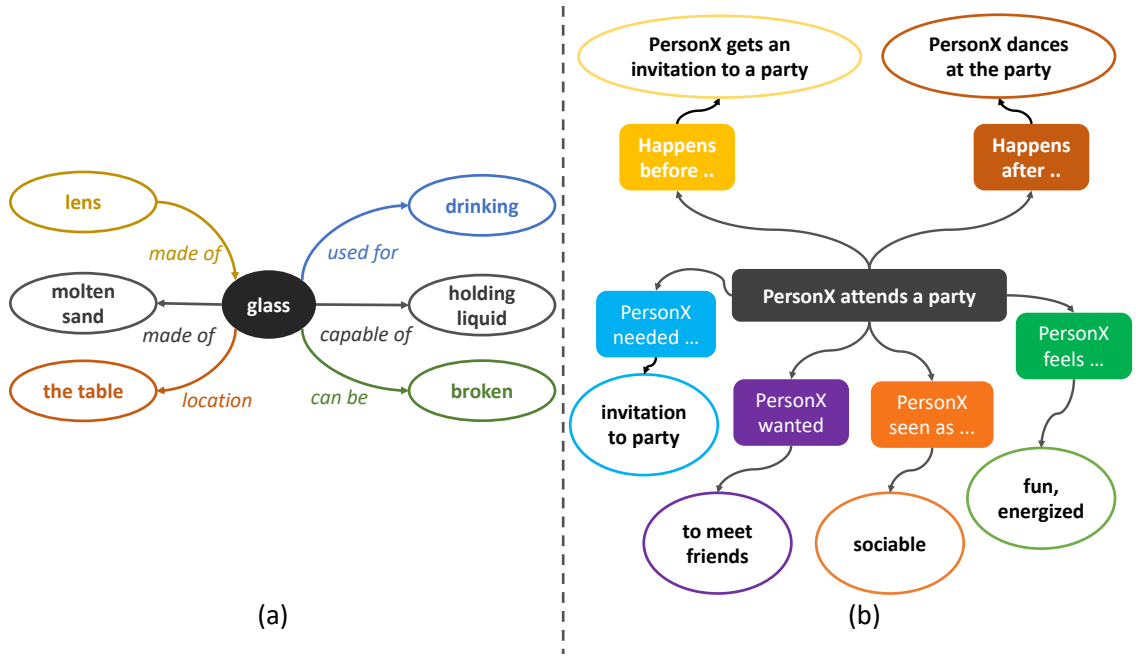


Fig. 2.2 Examples from (a) ConceptNet (Speer et al., 2017), (b) ATOMIC knowledge graph (Sap et al., 2019a)

- **Event2Mind (Rashkin et al., 2018c)** is a corpus that supports commonsense inference about people's intents and reactions, described in short free-form text over a diverse range of everyday events. The knowledge focused on stereotypical intents and reactions of people involved in the events.

- **ATOMIC (Sap et al., 2019a)** is a commonsense knowledge graph consisting of 877K textual descriptions of inferential (*if-then*) knowledge obtained from crowd-sourcing. The knowledge expresses pre- and post-states for events and their their participants in a lexical form with nine relations (e.g., *xIntent, xNeed, xReact*, etc.). These relations connect the event in question with manifold properties, emotions, as well as other states or events. Figure 2.2(b) shows an example of causes and effects of an event *"PersonX attends a party"* from ATOMIC knowledge graph.

- **ASER (Zhang et al., 2020a)** is a large-scale eventuality knowledge graph. The authors proposed to extract eventualities from a wide range of corpora from different sources based on dependency graphs. The relations were automatically extracted using a selected set of seed (unambiguous) connectives found from PDTB (Marcus et al., 1993). The knowledge graph consists of around 11-billion-token unstructured textual data. It contains 194 million unique eventualities, 15 relation types divided into five

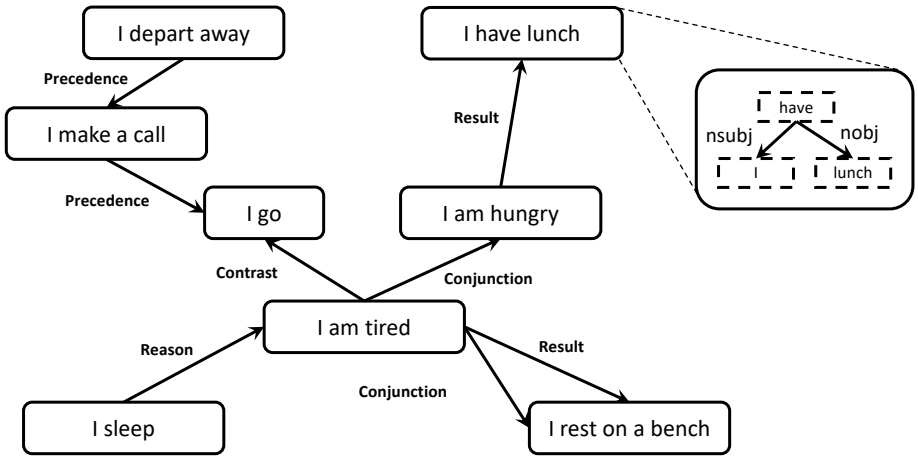


Fig. 2.3 Examples from (a) ASER. Eventualities are connected with weighted directed edges. Each eventuality is a dependency graph. Source : (Zhang et al., 2020a)

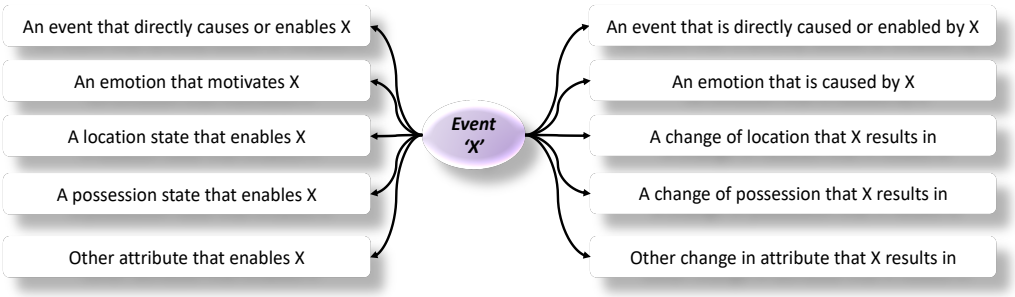


Fig. 2.4 GLUCOSE contains 10 causal relation types (Mostafazadeh et al., 2020b)

categories, and 64 million edges between them. Figure 2.3 depicts an example from ASER.

- **GLUCOSE (Mostafazadeh et al., 2020a)** is a dataset, which contains implicit commonsense knowledge in form of semi-structured general and specific inference rules. It has a unique take on explaining story events, inspired by human cognitive psychology. In GLUCOSE, give a story *S* and a selected sentence *X* from the story, the authors define ten causal dimensions to explain the sentence *X*. These ten dimensions are about states (i.e., location, possession states of an event), motivations, and emotions. GLUCOSE dataset is sourced from ROCStories (Mostafazadeh et al., 2020b) and the knowledge is crowd-sourced based on semi-automatic templates, and generalized

Dimensions	**Semi-structured Specific Statement and Inference Rule: *antecedent* connective *consequent***
1: Event that directly causes or enables X	A car turned in front of him *Causes/Enables* Gage turned his bike
	Something$_A$ turns in front of Something$_B$ (that is Someone$_A$'s vehicle) *Causes/Enables* Someone$_A$ turns Something$_B$ away from Something$_A$
2 : Emotion or basic human drive that motivates X	Gage wants safety *Causes/Enables* Gage turned his bike
	Someone$_A$ wants safety *Causes/Enables* Someone$_A$ moves away from Something$_A$ (that is dangerous)
6 : Emotion or basic human drive that motivates X	Gage turned his bike *Causes/Enables* He fell off his bike
	Someone$_A$ turns Something$_B$ (that is Someone$_A$'s vehicle) *Causes/Enables* Someone$_A$ falls off Something$_B$
8 : Emotion or basic human drive that motivates X	Gage turned his bike away from the car *Results* in Gage was further from the car
	Someone$_A$ moves away from Something$_A$ *Results* in Someone$_A$ is further from Something$_A$

Table 2.3 An example from GLUCOSE knowledge. Given a story and a sentence from the story X = *"Gage turned his bike sharply"*. White and gray rows show specific statements and general rules, respectively. Source : Mostafazadeh et al. (2020b)

from individual stories to more abstract rules. Figure 2.4 depicts the dimensions in GLUCOSE and 2.3 demonstrates an example from GLUCOSE.

In this thesis, we use different knowledge resources for each task settings for the following reasons:

1. We use ConceptNet to explain the characters' human needs in a story. ConceptNet contains *how* and *what* commonsense knowledge (for example, it contains relational knowledge like *is a*, *used for*, *has a*, etc.), which we hypothesize to be helpful to explain the mental states of characters in a story (see Chapter 4). ConceptNet contains a large amount of entities (8 million nodes see Table 2.2), therefore extracting relevant knowledge is a challenging task. To address that we propose a graph-based method to extract relevant knowledge (see Chapter 4).

2. We utilize ATOMIC to investigate the usefulness of inferential knowledge on downstream social commonsense reasoning tasks. However, given the amount of commonsense knowledge needed for real-world tasks, the size of the ATOMIC knowledge resource is small, hence incomplete. Thus, we propose to dynamically generate inference rules for downstream SCR task (see Chapter 6).

3. We use GLUCOSE for narrative story completion tasks. GLUCOSE contains contextualized knowledge, we propose learning to generate contextualized inference rules

and use the inference rules to improve the quality of generated story sentences (see Chapter 7).

The following sections will briefly overview various approaches ranging from symbolic to recent advanced deep neural approaches to address problems like knowledge representation to natural language reasoning. This section will also point out some of the limitations of existing methods.

2.3 Symbolic Approaches

Symbolic approaches make use of logical forms and perform inferences. There has been a wide range of work in logical reasoning, starting from Aristotle's theories of logic and deductive reasoning (Aristotle, 1989; Smith, 2020) to modern mathematical, logical frameworks by Selman and Levesque (1990); Hobbs et al. (1993b); de Morgan (2002). Simultaneously, Peirce (1883) proposed the process of logical abduction (non-deductive inference), i.e., the process of identifying a hypothesis from a limited set of observations. Similarly, later Davis and Marcus (2015) proposed plausible inference as a natural language problem. In this thesis, we present a method to address the task of abductive reasoning for NLP.

In Linguistics, Lakoff (2004) proposed a theory of natural logic (a logical form for natural language) in the direction of a semantic representation of language. *Natural logic* 's goal is to represent all concepts capable of being expressed in natural language, to characterize all the valid inferences that can be made in natural language. Meanwhile, Zadeh (1975) proposed fuzzy logic (a basis for approximate reasoning), which maps linguistic descriptions (words or sentences in a natural language) of numeric variables to probability distributions over the numeric variables. For example, the word *age* is a variable if its value is more linguistic than numerical, such as *young* or *old*, and someone may calculate the likelihood of these words being used to describe someone over a range of ages. It was a significant development to handle ambiguity in human language. In 1990s, symbolic approaches were pre-dominantly used for knowledge representation and semantic processing of language (Birnbaum, 1991; Menzies, 1996).

Raina et al. (2005) proposed a system for the textual inference that uses parsed sentences (a logical-formula semantic representation of text) and learned assumptions. The system aims to combine statistical machine learning and classical logical reasoning to find the robustness and scalability of learning with the preciseness and elegance of logical theorem proving. Recently, Gordon (2016b) proposed a new dataset named Triangle-COPA and presented a benchmark by creating a set of manually authored logic and commonsense rules and

performing logical reasoning. The logic-based benchmark achieved 91% accuracy on the Triangle-COPA task.

The main limitations of symbolic approaches are scalability, and due to this reason, these methods are not competitive in recent benchmarks with large data sizes. Consequently, creating manually authored logic and rules are time-consuming and expensive. Hence, we need methods to automatically capture the variation in knowledge and language and semantic phenomena. In Chapter 6 and 7, we study methods that learn to automatically generate inference rules. The next section provides some of the basic building blocks of many machine learning models introduced in later chapters.

2.4 Deep Learning in Natural Language Processing

In the NLP field, statistical approaches were predominately used to address the scalability problem from the mid-1990s to the 2010s. Most statistical approaches relied on engineered features to train various statistical models from training data and applied them to various NLP benchmarks (Manning and Schütze, 2002). Later, with the shift in computational paradigm, there was a shift from statistical to neural approaches. Neural approaches are used to identify valuable features in the data, rather than manually selecting features.

The inspiration behind Neural Networks (NNs) is the human brain. The neural network came from a very popular machine learning algorithm named perceptron (Freund and Schapire, 1999). The perceptron is a mathematical model of a biological neuron. A computational neural network architecture comprises multiple interconnected units called neurons. In practice, multiple layers of neurons are added in a network.

Word Embeddings. Most machine learning models require inputs in a vector form for computation. Therefore, in NLP, it is essential to represent discrete data (words, characters, and sentences) in vector form that a neural network can process. A naive approach in constructing such vector representations $x \in \mathbb{R}^V$ is *one-hot-encoding*, where V is the vocabulary size. Due to sparsity, the curse of dimensionality (Neal, 2007), and the incapability of capturing the polysemous nature of language, several works considered using the distributional property of language. Bengio et al. (2003) is the first to propose learnable dense representations for word embeddings. Bengio et al. (2003) initialized word embeddings as dense vectors $x \in \mathbb{R}^{VXD}$, allowing each word in the vocabulary to be represented by a unique vector with D dimensions. Later, word embeddings are usually trained using neural networks on large-scale text corpora. Among traditional word embedding models word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) are most popular. In Chapter 4, we use GloVe embeddings as initialization embeddings for our models. However, these

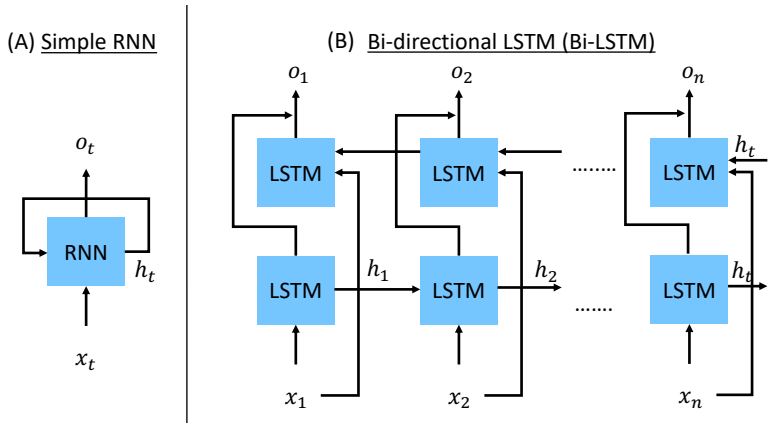


Fig. 2.5 (a) A simple Recurrent Neural Network (left-hand side). (b) An illustration of the BiLSTM architecture where each token x_t from the input sequence is mapped to a corresponding label o_t (right-hand side).

embedding vectors are context-independent. Hence, the embedding of a target word is always constant. Recently, some works focused on incorporating context into word embeddings, such as Embeddings from Language Models (ELMO) by Peters et al. (2018) and Bidirectional Encoder Representations from Transformers (BERT) by Devlin et al. (2019). These models provide different distributional representations based on the context in which they appear. These pre-trained word representations can be used as initialization embeddings or fine-tuned for downstream tasks. In Chapter 4, we use ELMO embedding as as initialization embeddings for our models and in Chapter 5, and 6 we use BERT to fine-tune on downstream tasks.

2.4.1 Recurrent Neural Networks

Classical deep feed-forward neural networks generally assume that data points are independent of each other. However, in Natural Language Processing (NLP), words are the input features, and the input is presented as a sequence of tokens. Hence, we need models that can deal with sequential data and are powerful enough to compress full-text sequences down to arbitrarily-sized vector representations. Recurrent neural networks (RNNs) (Elman, 1990) are expressive neural models that can encode representations for sequential inputs. The recursive nature of RRN takes previous vector x_{t-1} in the sequence and its own previous state h_{t-1} as input at each time-step t to upgrade the current state h_t. A vanilla RNN is defined using the following equation:

$$h_t = tanh(W_x * x_t + W_h * h_{t-1} + b_h) \tag{2.1}$$

where, h_t is the current hidden state of the network, h_{t-1} is the hidden state at the previous time step, and x_t is the input to the RNN at the current time step. One significant drawback of the vanilla RNN is that it suffers from the vanishing gradient problem[2]. For long input sequences the gradient are computed through multiple steps. Hence, the gradient either saturates to zero (*vanishes*) or they grow too large (*explodes*) which makes the learning process unstable. Gated recurrent neural networks such as the long short-term memory (LSTM) and gated recurrent unit (GRU) have been more commonly used models to overcome this problem.

The Long Short-term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) is an RNN-based architecture that controls gradient flow through its network using a cell state to avoid passing through a bounded activation function. An LSTM contains a cell state c_t, which encodes a hidden representation of the information collected by the LSTM up to a point in time t. The LSTM also contains multiple gates (input gate, forget gate, output gate) to control how input information is added to the cell state. The input gate (i_t) controls the information fed into the cell state. The forget gate (f_t) controls the information retained by the cell state. Finally, the output gate (o_t) estimates how much information the cell state should be extracted after computing the hidden representation. The values of these gates are computed in the following way:

$$i_t = \sigma(W_{x_i} * x_t + W_{h_i} h_{t-1} + b_i) \tag{2.2}$$

$$f_t = \sigma(W_{x_f} * x_t + W_{h_f} * h_{t-1} + b_f) \tag{2.3}$$

$$o_t = \sigma(W_{x_o} * x_t + W_{h_o} * h_{t-1} + b_o) \tag{2.4}$$

where all W are unique weights, all b are unique bias, x_t corresponds to the input vector to the LSTM at any time step, and h_{t-1} corresponds to the output of the LSTM at the previous step. Bidirectional-LSTM (Bi-LSTM) is a variant of LSTM, which computes a representation of a sequence in both the forward and backward directions.

In this thesis, for consistency purpose, from now on when we talk about recurrent neural networks, we will refer to LSTM and Bi-LSTM. We use recurrent neural networks as a model base in Chapter 4.

[2]Vanishing Gradient problem arises because of the non-linear functions used in practice (e.g. sigmoid) in the neurons squash the numeric values into a small region.

2.4.2 Attention Mechanism

The LSTM model addresses the vanishing gradient problem, and it should capture the long-range dependency better than the RNN. However, it is shown by (Cho et al., 2014) that for specific cases, it becomes forgetful. Additionally, with the vanilla LSTM model, it is difficult to learn how to give more importance to some input words compared to others. The *attention mechanism* was first introduced as an application by Bahdanau et al. (2015) to address the above limitations for neural machine translation. Bahdanau et al. (2015) propose a simple yet elegant idea where they suggested considering not only all the input words in the context representation but also relative importance for each word in the context. Given a sequence of hidden states $h_1, h_2, ..., h_T$, the attention mechanism combines all hidden states to calculate the context representation c_i:

$$c_i = \sum_{j=1}^{N} \alpha_{ij} h_j \tag{2.5}$$

where N is the number of words in the sequence. The weights α_{ij} are calculated by a softmax function given by the following equation:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{N} exp(e_{ik})} \tag{2.6}$$

where e_{ij} is the output score of a feedforward neural network.
There are the following advantages of the attention mechanism :

- It addresses the vanishing gradient problem by providing a way to consider words that are far away in the input sequence.
- The learned attention distribution can provide an alignment between inputs and outputs, which allows some understanding of their relations.

Because of the above advantages, attention mechanisms have been often applied to many NLU benchmark tasks. We incorporate attention mechanism in our neural model discussed in Chapter 4.

2.4.3 Transformers

In recurrent neural networks, the input representations h_t at each time step depends on the previous steps h_{t-1}, hence hindering the benefits offered by modern parallel computing hardware. To combat this problem, (Vaswani et al., 2017) proposed the transformer model

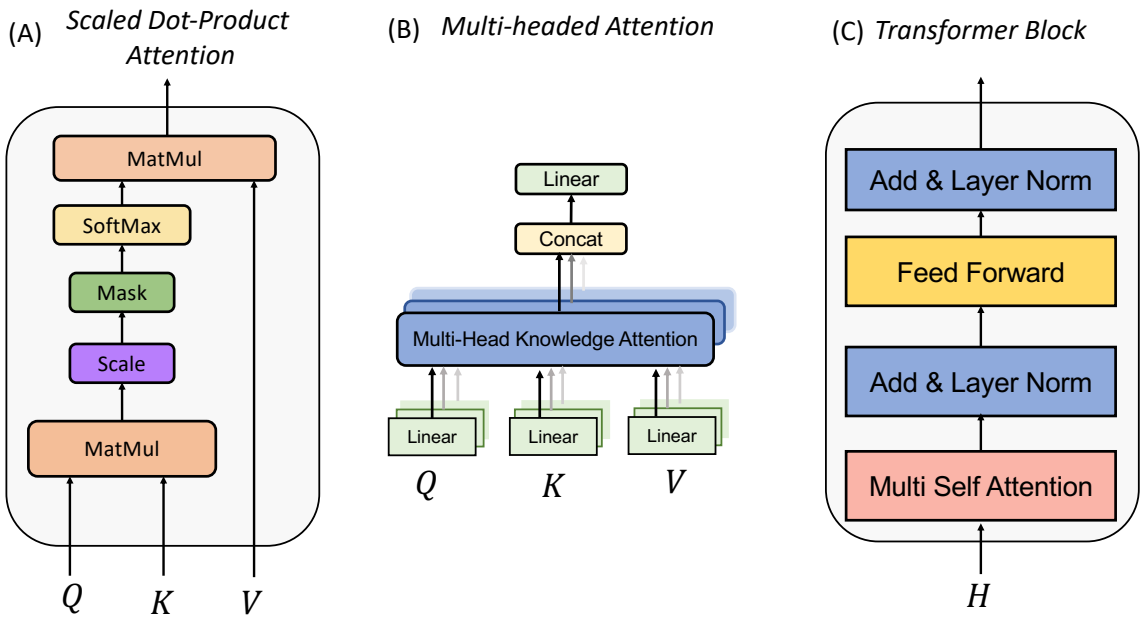


Fig. 2.6 (a) Scaled Dot-Product Attention,(b) Multi-Head Attention consists of several attention layers in parallel. (middle), (c) Transformer Block (right)

that comprises multiple self-attention layers to construct the input representations. Self-attention allows the model to build the input representation in parallel, because it re-computes a state representation from input at every step. Unlike in recurrent neural networks, the representations from the transformer do not rely on a time-dependent state. Self-attention also allows it to attend to all positions in the sequence to get a better-contextualized representation of a word. We explore the different components of a Transformer model in the following paragraphs.

Positional Embedding. The order of words and their position are crucial parts of any language. They help in defining the grammar and the semantics of a sentence. Recurrent neural networks consider the order of words as it sequentially parses a sentence. However, in Transformer architecture the recurrence mechanism is absent, therefore Vaswani et al. (2017) proposed to add a position embedding p_t along with the input sequence x_t. A position embedding can be initialized as the exact position of the word in a sentence. The input h_t^0 to the first transformer layer is following:

$$h_t^0 = x_t + p_t \tag{2.7}$$

Transformer Block. A transformer is composed of stacked layers of identical transformer blocks. Each transformer block comprises of the following transformation steps:

$$a^l = \text{MultiHead}(h^{l-1}) \tag{2.8}$$

$$f^l = \text{LN}(a^l + h^{l-1}) \tag{2.9}$$

$$\hat{h}^l = \text{FFN}(f^l) \tag{2.10}$$

$$h^l = \text{LN}(\hat{h}^l + f^l) \tag{2.11}$$

where MultiHead is a multi-headed self-attention mechanism, FFN is a fully connected feed-forward network, and LN is a layer normalization (Ba et al., 2016) operation that is applied to the output of the self-attention and the feed-forward network.

Multi-Head Attention. Instead of performing a single attention function, the transformer performs multi-head attention. It applies the attention function multiple times with different linear projections and allows the model to capture different attentions from different subspaces jointly. Vaswani et al. (2017) has defined three inputs for multi-head attention function: a query $q \in \mathbb{R}^D$, a set of keys $K \in \mathbb{R}^{N \times D}$, and a set of values $V \in \mathbb{R}^{N \times D}$. The attention is made of multiple heads where each head computes a unique scaled dot product attention distribution over V using Q and K, see Fig. 2.6:

$$\text{Attention}(q, k, V) = \text{softmax}(\frac{qK^T}{\sqrt{D}})V \tag{2.12}$$

where D is the input embedding size. The attention heads are concatenated and projected to yield the final values.

$$\text{MultiHead}(q, k, V) = \text{Concat}(head_1, ...head_h))W^o \tag{2.13}$$

where $head_i$ is Attention(q, k, V), and W^o is an output projection of the concatenated outputs of the attention heads.

The recent advancement in transfer learning methods (Ruder et al., 2019; Rothe et al., 2020), transformer model (Vaswani et al., 2017), and the resource-intense process of using large training data led the research towards training language models. With the release of these trained language models, the NLP community applies them to their NLP problems using small, if any, additional data. There are two major variants of such transformer-based language models, which are as follows :

- **Conditional Transformer Model** (Radford et al., 2018, 2019; Brown et al., 2020) is trained on a typical language modeling objective i.e., to assign a probability to a

sequence of words given some conditioning context (x):

$$p(w_n|x, w_1, .., w_{n-1}) \tag{2.14}$$

A left-to-right transformer model is trained to maximize the conditional log-likelihood of predicting the next word in a sequence given the context. We use left-to-right transformer language models as important components of our approaches in Chapters 5, 6 and 7.

- **Bidirectional Transformer Model** was introduced by Devlin et al. (2019), famously known as BERT, which uses the transformer encoder to read the entire sequence of words at once. BERT was trained with masked-language modelling (MLM) and next sentence prediction objectives, i.e., it is trained to predict words that are masked from the input. Unlike uni-directional or conditional transformer models (see Eq. 2.14), bidirectional transformers can condition on future tokens as well. Later, Liu et al. (2019) proposed RoBERTa, which outperforms BERT's performance using more training data, and the same architecture without the next-sentence prediction objective. We use bidirectional transformer language models as base model in the empirical study in Chapters 5 and 6.

2.5 Explainablity of NLP models

Despite the success of large pre-trained language models, recent studies have raised some critical points such as: high accuracy scores do not necessarily reflect understanding (Min et al., 2019), large pretrained models may exploit superficial clues and annotation artifacts (Gururangan et al., 2018; Kavumba et al., 2019). Therefore, the ability of models to generate explanations has become desirable, as this enhances interpretability. Explainations are often categories into two aspects (Guidotti et al., 2018; Danilevsky et al., 2020) : (1) *local* vs *global* (2) *self-explaining* vs *post-hoc*.

A local explanation justifies the model's prediction on a specific instance. Whereas a global explanation justifies the model's prediction process as a whole. Moreover, explanations can be further categorised into two categories – (i) when explanation generation is a part of the prediction process and (ii) when explanation generation requires post-processing after the model's prediction. A self-explaining approach generates an explanation while making a prediction, using information generated by the model. On the contrary, a post-hoc approach requires an additional operation to be performed after the model prediction.

There are a lot of approaches used to enable explainability in our literature (Guidotti et al., 2018; Moradi and Samwald, 2021), however, here we mention some notable methods.

1. *Saliency* is a neural network interpretation method that interpret a specific output y made by a model M, by assigning a distribution of importance $\phi(F)$ over the input feature set F of the original neural network model (Li et al., 2016a; Arras et al., 2016; Mudrakarta et al., 2018; Ding et al., 2019). The most widely used method to calculate importance is by the gradient (Simonyan et al., 2014). It estimates the contribution of input $x \in F$ towards output y by computing the partial derivative of y with respect to x.

2. *Attention* is a popular strategy to explain the feature importance in a neural network architecture (Luo et al., 2018; Xie et al., 2017; Li et al., 2019b). In section 2.4.2, we explain the basic idea behind attention mechanism. The key idea is that an attention layer in a NN architecture can help indicate where the model is "*focusing*". Recently, few studies have focused on answering the question that *how much explainability attention provides?* (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019; Serrano and Smith, 2019).

3. *Input Perturbations.* Recently, few studies have shown that models can be sensitive to small changes in the input that cause the model to make wrong decisions (Ribeiro et al., 2020; Moradi and Samwald, 2021). Hence, to test the robustness of a model, the input $x \in F$ is perturbed to study the change in the model's output y. One can view this as an explainability method because it explains which input features x are responsible for the correct or wrong outcome. There can be different kinds of (word-level) perturbation methods; some are as follows: *addition, deletion, replace with synonym, negation, singular verb to plural verbs, word order, verb tense*, etc.

In this thesis, we consider explainability from an end user's perspective whose goal is to understand how a model arrives at a particular output. We focus on local explanations, i.e., providing justification (using external commonsense knowledge) for the model's prediction on a specific input. In Chapter 4 we study the local self-explaining approach by investigating the learned attention distributions of the interactions between input representation and knowledge paths in order to interpret how knowledge is employed to make predictions. In Chapters 5 and 7 we study local self-explaining approach by investigating the relevance of generated knowledge. This process is similar to a human explanation as the model generates natural language text as an explanation. In Chapter 6, we study the local post-hoc method by perturbing knowledge input to study its impact on the models' output.

Chapter 3

Related Work

This chapter reviews prior related works to the research questions we address in this thesis. We start by surveying the most prominent knowledge extraction and integration approaches for LSTM-based and Transformer-based models in sections 3.1 and 3.2. We also discuss some of the limitations of these existing methods (sections 3.1.1, and 3.2.1). The purpose of this chapter is to position this thesis in the broad research spectrum.

3.1 Extracting Commonsense Knowledge

"Nothing is truth unless proven by knowledge and reason."

– Albert Williams

In recent years, the acquisition of machine-readable knowledge has become an essential sub-task for building systems that can automatically perform reasoning about daily situations (Storks et al., 2019). There has been a large number of works on (1) automatically acquiring structured knowledge from unstructured data (Mausam et al., 2012; Mausam, 2016), (2) knowledge representation learning (Bordes et al., 2013; Li et al., 2019a). However, it is difficult to acquire commonsense knowledge because of its implicit nature automatically. In section 2.2, we mentioned about some notable large social commonsense knowledge graphs. This thesis assumes that there exist commonsense knowledge graphs and instead focuses on developing methods for automatic extraction of the relevant knowledge[1] for downstream tasks.

Extract Knowledge from Static Knowledge Graphs. A standard approach to extract knowledge for a given text has two steps : (a) Linking text to Knowledge Graphs (KGs) and

[1] relevant knowledge is a knowledge that is correct, complete and can be used to explain the context.

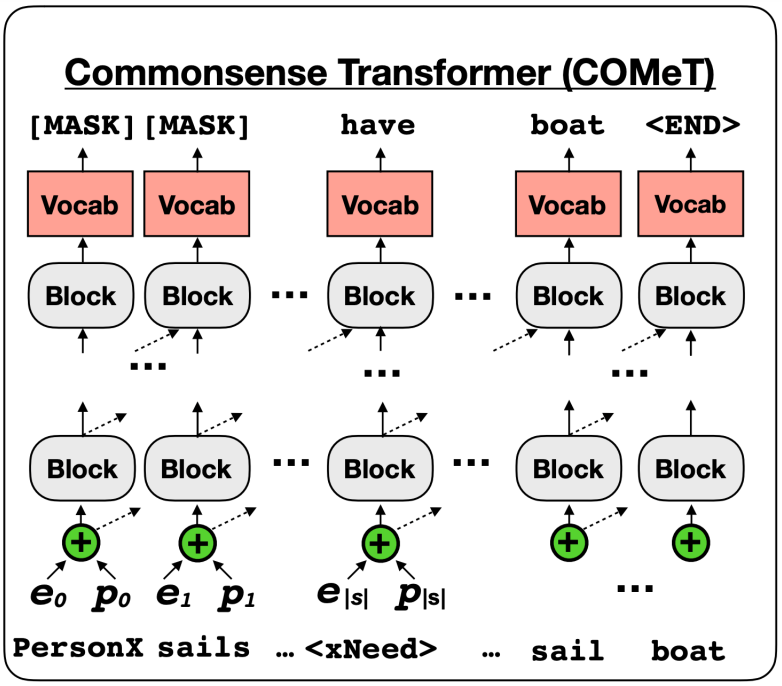


Fig. 3.1 COMET Model Architecture. Source: (Bosselut et al., 2019)

(b) Retrieve knowledge relevant to the text. For the first step, a simple approach is *string matching* i.e., to identify keywords in the input data to query large KGs (e.g., ConceptNet, WordNet, NELL, etc.) using only the lemmatized noun, verb, adjective words for each instance, with stop words excluded. Recently, Becker et al. (2021) proposed COCO-EX, a tool for extracting and linking concepts from texts to the ConceptNet knowledge graph. For the second step, Xu et al. (2017) proposed to rank the entity-attribute pairs using *tf-idf* score. Later, Mihaylov and Frank (2018a) proposed a heuristics method based on term frequency to retrieve triples (head node, relation, tail) within one hop. Concurrently, Bauer et al. (2018a) proposed a multi-hop commonsense knowledge path retrieval method for Reading Comprehension Task. They build 'prototype' paths by constructing trees rooted in concepts in the query with the aim of selecting paths with high recall. Additionally, they proposed a heuristic method to rank and filter knowledge paths with the aim to improve the precision of useful concepts in paths.

Dynamically Generate Knowledge. A shortcoming of static KGs is that they are incomplete. To address this shortcoming, Bosselut et al. (2019) proposed a framework, named COMET, for modifying the weights of language models to learn to generate novel commonsense knowledge tuples. One key component of this framework is the conditional transformer model (see section 2.4.3), which is trained to produce the phrase object o of a knowledge tuple given the tuple's phrase subject s and relation r. Figure 3.1 depicts the

COMET architecture, where each word is an input to a first-layer transformer block along with all preceding tokens.

3.1.1 Limitations

Although the above knowledge extracting approaches have worked well for the different downstream tasks, here we mention some significant drawbacks:

1. **Single-Hop Knowledge.** Only considering a single-hop knowledge tuple will allow preserving the information of direct relations in a KG. However, it ignores indirect relations, and for a task that requires multi-hop reasoning, such single-hop knowledge will be incomplete.

2. **Knowledge Retrieval Methods.** Previous works used methods that are either bias towards frequency of concepts (Xu et al., 2017) or methods which are task-dependent (Mihaylov and Frank, 2018a; Bauer et al., 2018a). Ideally, we want a commonsense knowledge retrieval method that is unbias and can be used for various tasks.

3. **Dynamically Generating Knowledge.** Training a language model on KGs can generate novel knowledge but does not ensure that it will generate contextualized knowledge. However, addressing commonsense reasoning tasks requires context-dependent knowledge. Hence, such generated knowledge can add noise in knowledge representation and be insufficient.

In this thesis, we address the above limitations, and we investigate the importance of selecting multi-hop paths over considering only single-hop paths (see Chapter 4). To show that our proposed method is task-agnostic we apply it on the *argument classification task* (Paul et al., 2020). Further, we study the importance of automatically generating contextualized vs non-contextualized knowledge on a narrative story completion task (see Chapter 7).

3.2 Integrating Commonsense Knowledge.

> *"All possible knowledge, then, depends on the validity of reasoning. Unless human reasoning is valid no science can be true. "*
>
> – C.S.Lewis

There have been several attempts at using external knowledge to build hybrid models to address various NLP tasks. Most research on integrating knowledge into the network models

has been done based on three key reasons: (a) improving the performance of neural models on downstream tasks, (b) reduction in the amount of data needed to train the model, (c) improve the coverage of model parameters. In particular, the main reason for incorporating commonsense knowledge is to compensate for the lack of implicit knowledge in neural models. Bar-Hillel (1960b) was one of the first to recognize the importance of incorporating commonsense knowledge in NLP systems. Bar-Hillel argued that it is not feasible to build fully automatic high-quality machine translation systems without addressing the requirement of world knowledge to help machine translation to infer correct translations for ambiguous words or linguistic structures (Bar-Hillel, 1960b). Lately, with the development of new advanced neural models, the idea of integrating background knowledge has regained more attention in NLP research.

Integrating Commonsense Knowledge into LSTM-based Models. Incorporating external knowledge in a LSTM[2] model has proven beneficial for several NLP tasks: *text generation* (Kiddon et al., 2016), *language modeling* (Ahn et al., 2016), *dialogue generation* (Yang et al., 2017), *sentiment analysis* (Ma et al., 2018), *question answering* (Lan et al., 2019) and *reading comprehension* (Bauer et al., 2018b).

Although different methods are used to inject knowledge into LSTM-based models, in this section we will only review the most notable methods. Xu et al. (2017) showed that injecting loosely structured knowledge with a recall-gate mechanism is beneficial for conversation modelling. The recall-gate mechanism was designed to convert structured domain knowledge to a global memory, which incorporates with the local cell memory of LSTM to provide information to judge whether a sentence is related to another or not. Mihaylov and Frank (2018b) and Weissenborn et al. (2018) proposed the integration of commonsense knowledge for reading comprehension: the former explicitly encode selected triples from ConceptNet using attention mechanisms, the latter enriches question and context embeddings by encoding triples as mapped statements extracted from ConceptNet. Bordes et al. (2014) made use of knowledge bases to obtain longer paths connecting entities appearing in questions to answers in a QA task. They provided a richer representation of answers by building subgraphs of entities appearing in answers. Bauer et al. (2018a) proposed a heuristic method to extract multi-hop paths from ConceptNet for a reading comprehension task. They construct paths from concepts appearing in the question to concepts appearing in the context, aiming to emulate multi-hop reasoning. Tamilselvam et al. (2017) used ConceptNet relations for aspect-based sentiment analysis. Yang and Mitchell (2017) proposed to incorporate knowledge directly into the LSTM cell state to improve event and entity extraction. They used the BILINEAR (Yang et al., 2015) model to extract knowledge embeddings trained

[2]In section 2.4.1, we described the LSTM model architecture.

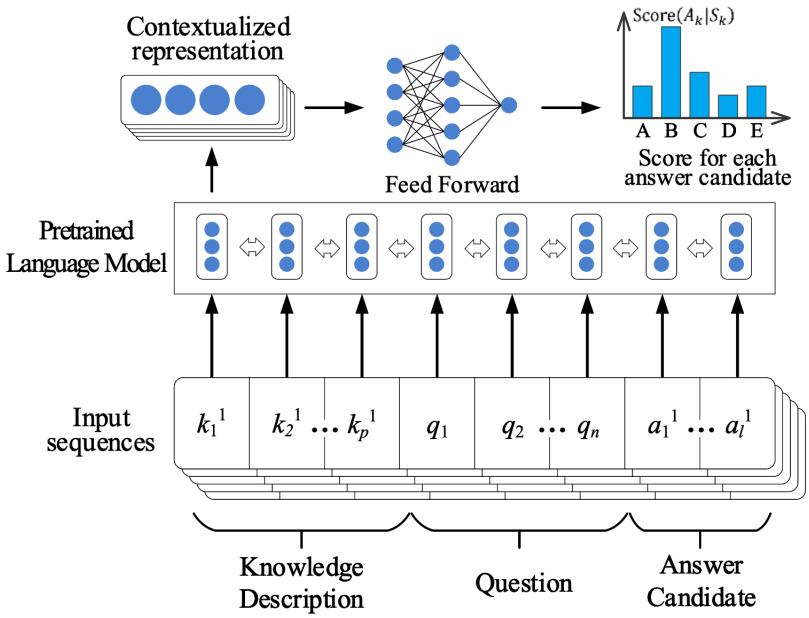


Fig. 3.2 The architecture for performing Question Answering task with external knowledge. Source: (Bian et al., 2021)

on WordNet and NELL (Miller, 1995). Kiddon et al. (2016) proposed a neural checklist model that generates globally coherent text and tracks the agenda of items. Ahn et al. (2016) proposed a language model named NKLM that combines symbolic knowledge from knowledge graphs with an LSTM-based language model. The model copies fact attributes from a topic knowledge memory. Then the model is used to predict a fact in the knowledge memory using a gating mechanism, and given this fact, the next word to be selected is copied from the fact attributes. We integrate commonsense knowledge into LSTM based model using attention mechanism to examine to what extent it can contribute to and improve the prediction of people's mental states in a narrative story setting (ref. Chapter 4).

Integrating commonsense knowledge into Transformer Models. Since 2018, there is a shift in NLP research, due to the impressive performance of transformer models. In section 2.4.3, we described the transformer architecture. There has been a few notable works which aim to incorporate external commonsense knowledge into SOTA transformer models. For *classification tasks*, a standard approach is to concatenate the extracted external knowledge as an input along with the input context (Banerjee et al., 2019; Mitra et al., 2019; Bian et al., 2021). Figure 3.2 shows one knowledge-enhanced model for Multi-Choice Question Answering task from Bian et al. (2021), where the input is a sequence of concatenated tokens from the knowledge description $K^m = (k_1^m, .. k_p^m)$, question $Q = (q_1, .. q_n)$ and the answer $A^m = (a_1^m, .. a_l^m)$ for each question and answer. It is passed through a transformer model

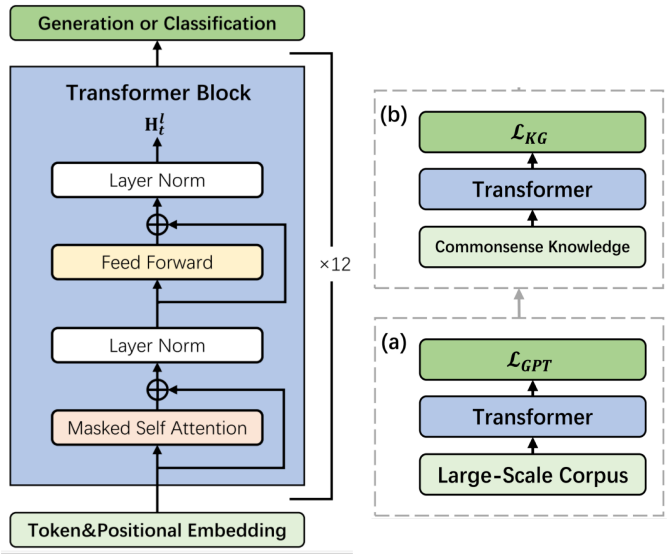


Fig. 3.3 The architecture of Knowledge Enhanced Pretrained model from (Guan et al., 2020). Transformer block architecture (left) and training framework (right). Train the language model GPT-2 (Radford et al., 2018) (a) with a large-scale corpus, (b) with commonsense knowledge from external knowledge bases. Source: (Guan et al., 2020)

to obtain a contextualized representation. After obtaining representation a feed-forward classifier is used as the outer layer to predict the answer scores.

For natural language generation task, similar to classification task, a simple approach is to concatenate the knowledge representations and word-embeddings (from pre-trained language models) and pass through a transformer-based NLG model to generate text (Bhagavatula et al., 2020). However, recent methods have explored beyond such simple concatenation (Guan et al., 2020; Liu et al., 2021). Guan et al. (2020) proposed a knowledge-enhanced pretraining model (KEP) for commonsense story generation task. Guan et al. (2020) transformed the commonsense triplets (ConceptNet, ATOMIC) into machine-readable sentences using templates. Then they fine-tuned the language model (GPT-2) on the transformed knowledge sentences to improve the long-range coherence of generated stories. Figure 3.3 illustrates an overview of KEP model. One advantage of the KEP model is that the model complexity does not change from the base GPT-2 model.

Concurrently, Ji et al. (2020) proposed a model named GRF that dynamically attends KG representations during the decoding step. First, the authors extract the subgraph consisting of inter-connected n-hop paths starting from the source concepts extracted from the input text. Then, they encode the subgraph using GNNs and the input text using pre-trained language

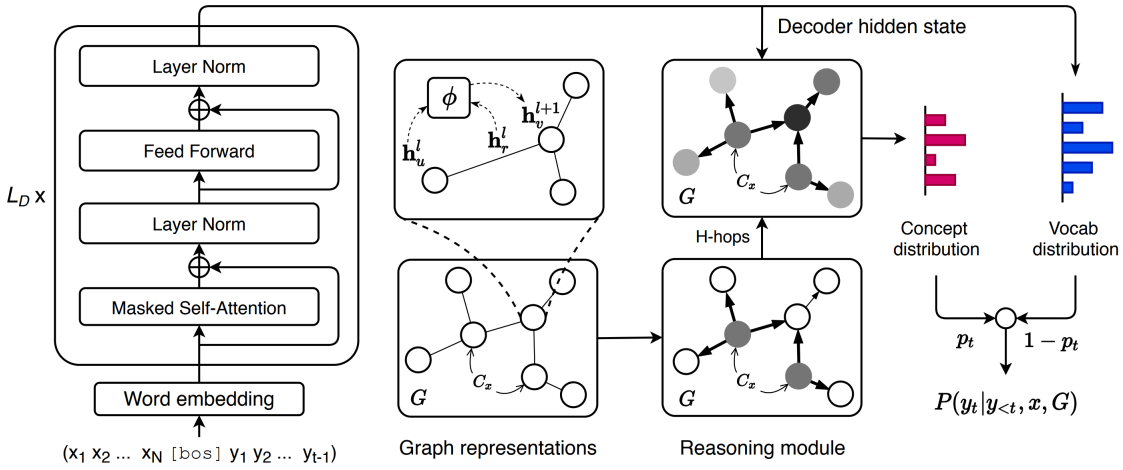


Fig. 3.4 The architecture of GRF model. It contains multiple steps: (a) context module with pre-trained transformer, (b) encoding the multi-relational graph with non-parametric operation to combine relations and concepts, (c) a multi-hop reasoning module aggregates evidence from source concepts along structural paths to all nodes where shade indicates the node score, (d) decoding module : generation distribution with gate control. Source: (Ji et al., 2020)

model (GPT-2). Figure 3.4 illustrates an overview of the GRF model. The main idea is that the sequence decoder of the model use attention mechanism to useful semantics from the subgraph representations as well as from the input text representations. Additionally, they added a relevance score that reflected the relevancy of the knowledge edge with respect to the decoding state. In short, GRF enables pre-trained models (GPT-2) with dynamic multi-hop reasoning on multi-relational paths extracted from the external ConceptNet commonsense knowledge graph.

3.2.1 Limitations

Although these knowledge integration approaches have worked well for the different downstream tasks, here we mention some significant drawbacks:

1. The method to encode both context and knowledge together can lead to a bottleneck for computation on a GPU with limited memory (Banerjee et al., 2019; Mitra et al., 2019; Bian et al., 2021).

2. The KEP model (Guan et al., 2020) has lower complexity but does not consider grounding the knowledge to the input context.

3. The GRF model (Ji et al., 2020) considered static subgraph knowledge, which might lead to low coverage of valuable concepts for generating the output. Recently Yu et al. (2022) argued that static subgraph knowledge methods like GRF do not utilize a large portion of relevant concepts in the generation process. Yu et al. (2022) found that only 21.1% of concepts in the output from Ji et al. (2020) could be found on ConceptNet, and only 5.7% of concepts in the output can be found on the retrieved 2-hop sequence-associated subgraph.

In this thesis, we address these limitations and propose a *multi-head* knowledge integration method for classification tasks that encode knowledge and context separately; for the NLG tasks, our method dynamically generates contextualized knowledge to support downstream NLG tasks (see Chapter 6 and 7 respectively).

Chapter 4

Commonsense Knowledge for Mental States Prediction in a Narrative Story

"Rationalization is a process of not perceiving reality, but of attempting to make reality fit one's emotion."

– Ayn Rand

In the first section of this chapter, we give motivation and introduce the mental states prediction problem. The second section reviews related work on mental states detection in NLP. In the third section, we present a task agnostic graph-based method to extract, rank, filter and select multi-hop relation paths from a commonsense knowledge resource to interpret the expression of sentiment in terms of their underlying human needs. The third section presents a method to integrate the acquired knowledge paths in a neural model that interfaces context representations with knowledge using a gated attention mechanism. In the fourth section, we evaluate our model in predicting appropriate categories from two theories of psychology: *Hierarchy of needs* (Maslow, 1943) and *basic motives* (Reiss, 2002) in a narrative story setting. Finally, we end the chapter with a human evaluation study to assess the relevance of the encoded knowledge. This chapter is based on work originally published in Paul and Frank (2019).

4.1 Mental States Prediction in Narrative Stories

Sentiment analysis and emotion detection are essential tasks in human-computer interaction. Due to its broad practical applications, in NLP there has been rapid growth in the field of sentiment analysis (Zhang et al., 2018). Although state-of-the-art sentiment analysis can

detect the polarity of text units (Hamilton et al., 2016; Socher et al., 2013), there has been little work towards explaining the reasons for the expression of sentiment and emotions in texts. Exploring the connection between language and people's psychology holds great potential for understanding the underlying reasons for people's emotional reactions (Li and Hovy, 2017). For example, given two syntactically similar expressions *"I broke my leg"* and *"I broke up with my girlfriend"* with same (negative) sentiment, however, the reason is very different from each other as one related to a need concerning 'health' and another related to 'social relationship'. The ability to reason about what others think or believe (mental states) is a crucial component of a system that can perform social commonsense reasoning.

The new advancement in technological capability and availability of the massive amount of textual data coming from interactions in social networks, blogs and online communities has made accessing people's emotional states remarkably easy. Hence, the research on human psychological phenomena has substantially increased, such as detecting or monitoring mental health problems. In this chapter we explore the connection between people's emotions and the naive psychology of characters in a narrative story. Figure 4.1 demonstrates an example of a narrative story annotated with human needs, basic motives and emotions. We see how the story character's (*Meg*) mental state and emotion changes over time in the narrative. For example, initially *Meg* was "sad and disgusted" but with her *Mom*'s support her emotion and needs changed, suggesting how change in the underlying need explains change in people emotion. Additionally, capturing these underlying human motives and needs of characters plays an important role in narrative understanding (Rashkin et al., 2018a). Humans are good at understanding situations described in natural language and can easily connect them to the character's psychological needs using commonsense knowledge. In this chapter, we aim to imitate humans by (i) learning to select relevant words from the text, (ii) extracting pieces of knowledge from the commonsense inventory and (iii) associating them with human motive or need categories put forth by psychological theories. We hypothesize that integrating commonsense knowledge into a neural model will be helpful in overcoming the lack of

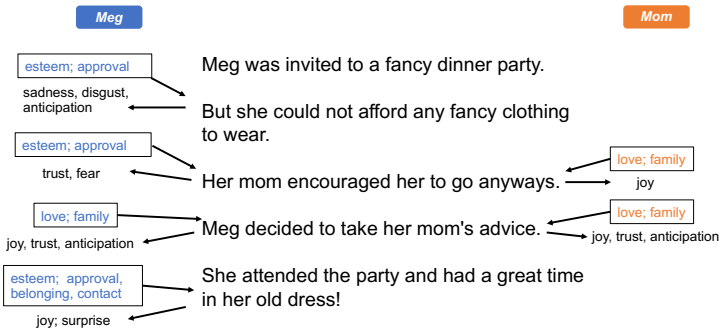


Fig. 4.1 A narrative story example with partial annotations for motivation and emotional reactions. Source : (Rashkin et al., 2018a)

textual evidence in establishing relation between story events and the inferable mental states of story characters.

4.2 Related Work

Lehnert (1981) proposed formalisms for affect and mental state in story narratives that included motivations and reactions. Lehnert (1981) hypothesized that to summarize a story, it is essential to access a high-level analysis of the story that highlights its central concepts. Lehnert considered plot units (emotional reactions, affect states) conceptual structures that overlap when a narrative is cohesive. Since then, there has been a growing interest in developing methods to model aspects of human behaviour from daily life events or stories. Goyal et al. (2013) proposed fully automated system, named AESOP that generates plot unit representations for narrative texts. AESOP performs four steps: affect state recognition, character identification, affect state projection, and link creation.

Chaturvedi et al. (2016) addressed the novel task of analyzing small pieces of text containing an expression of a desire to identify if the desire was fulfilled in the given text. They used three approaches: (a) a textual entailment model to analyze small fragments of texts independently; (b) an unstructured model to analyze the complete text as a whole; (c) a structured model to understand and model the narrative structure using latent variables. In the meantime, there have been multiple works related to goals, desires, wish detection (Goldberg et al., 2009; Rahimtoroghi et al., 2017). Most recently, Ding and Riloff (2018) propose to categorize affective events into physiological needs to explain people's motivations and desires. Interestingly, Li and Hovy (2017) argued that human needs could categorize the goals of an opinion holder. Unlike previous work, in this thesis, we focus on understanding the role of commonsense knowledge in identifying the character's motivation in a narrative story. Inspired by (Lehnert, 1981), Rashkin et al. (2018b) published a dataset for tracking the emotional reactions and motivations of characters in stories. In this work, we use this dataset to develop a knowledge-enhanced system that 'explains' sentiment in terms of human needs.

4.3 Extracting Multi-Hop Commonsense Knowledge Paths

Our task is to automatically predict human needs of story characters given a story context. In this task, following the setup of Rashkin et al. (2018b), we *explain* the probable reasons for the expression of emotions by predicting appropriate categories from two theories of psychology: *Hierarchy of needs* (Maslow, 1943) and *basic motives* (Reiss, 2002). The task is defined as a multi-label classification problem with five coarse-grained (Maslow) and 19

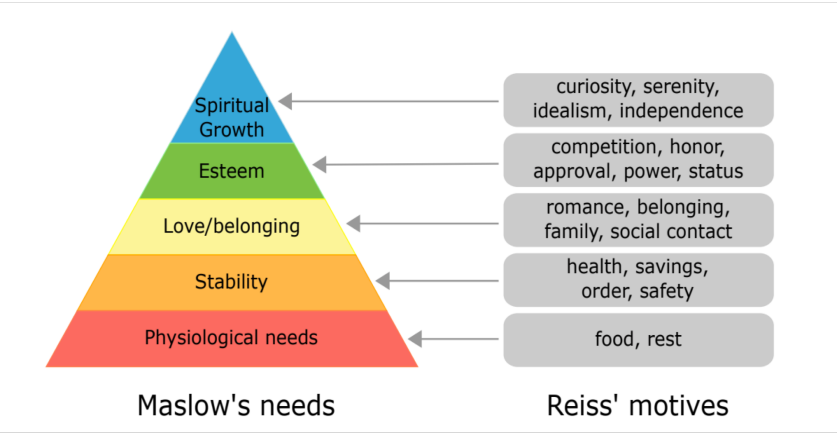


Fig. 4.2 Maslow and Reiss: Theories of Psychology as presented in Rashkin et al. (2018b).

fine-grained (Reiss) categories, respectively (see Fig. 4.2). We start with a Bi-LSTM encoder with self-attention as a baseline model, to efficiently categorize human needs. We then show how to select and rank multi-hop commonsense knowledge paths from ConceptNet that connect textual expressions with human need categories. We now describe each component in detail.

4.3.1 A Bi-LSTM Encoder with Attention to Predict Human Needs

Our Bi-LSTM encoder takes as input a sentence S consisting of a sequence of tokens, denoted as $w_1^s, w_2^s, ..., w_n^s$, or $w_{1:n}^s$ and its preceding context Cxt, denoted as $w_1^{cxt}, w_2^{cxt}, ..., w_m^{cxt}$, or $w_{1:m}^{cxt}$. As further input we read the name of a story character, which is concatenated to the input sentence. For this input the model is tasked to predict appropriate human need category labels $z \in Z$, according to a predefined inventory.

Embedding Layer: We embed each word from the sentence and the context with a contextualized word representation using character-based word representations (ELMo) (Peters et al., 2018). The embedding of each word w_i in the sentence and context is represented as e_i^s and e_i^{cxt}, respectively.

Encoding Layer: We use a single-layer Bi-LSTM (Hochreiter and Schmidhuber, 1997) to obtain sentence and context representations h^s and h^{cxt}, which we form by concatenating the final states of the forward and backward encoders.

$$h^s = BiLSTM(e_{1:n}^s); h^{cxt} = BiLSTM(e_{1:m}^{cxt}) \tag{4.1}$$

A Self-Attention Layer allows the model to dynamically control how much each token contributes to the sentence and context representation. We use a modified version of self-attention proposed by Rei and Søgaard (2018), where both input representations are passed through a feedforward layer to generate scalar values for each word in context v_i^{cxt} and sentence v_i^s (cf. (2-5)).

$$a_i^s = ReLU(W_i^s h_i^s + b_i^s), \tag{4.2}$$

$$a_i^{cxt} = ReLU(W_i^{cxt} h_i^{cxt} + b_i^{cxt}) \tag{4.3}$$

$$v_i^s = W_{v\,i}^s a_i^s + b_{vi}^s \tag{4.4}$$

$$v_i^{cxt} = W_v^{cxt}{}_i a_i^{cxt} + b_v^{cxt}{}_i \tag{4.5}$$

where, $W^s, b^s, W^{cxt}, b^{cxt}, W_v^s, W_v^{cxt}$ are trainable parameters. We calculate the soft attention weights for both sentence and context:

$$\widetilde{v_i} = \frac{1}{1 + exp(-v_i)}; \;\; \hat{v}_i = \frac{\widetilde{v_i}}{\sum_{k=1}^{N} \widetilde{v_k}} \tag{4.6}$$

where, $\widetilde{v_i}$ is the output of the sigmoid function, therefore $\widetilde{v_i}$ is in the range [0,1] and $\hat{v}_i$ is the normalized version of $\widetilde{v_i}$. Values $\hat{v}_i$ are used as attention weights to obtain the final sentence and context representations x^s and x^{cxt}, respectively:

$$x^s = \sum_{i=1}^{N} \hat{v}_i{}^s h_i^s \tag{4.7}$$

$$x^{cxt} = \sum_{i=1}^{M} \hat{v}_i{}^{cxt} h_i^{cxt} \tag{4.8}$$

with N and M the number of tokens in S and Cxt. The output of the self-attention layer is generated by concatenating x^s and x^{cxt}. We pass this representation through a FF layer of dimension Z:

$$y = ReLU(W_y[x^s; x^{cxt}] + b_y) \tag{4.9}$$

where W_y, b_y are trainable parameters and ';' denotes concatenation of two vectors. Finally, we feed the output layer y to a logistic regression layer to predict a binary label for each class $z \in Z$, where Z is the set of category labels for a particular psychological theory (Maslow/Reiss, Fig. 4.2).

4.3.2 Extracting Commonsense Knowledge

To improve the prediction capacity of our model, we aim to leverage external commonsense knowledge that connects expressions from the sentence and context to human need categories. For this purpose we extract multi-hop commonsense knowledge paths that connect words in the textual inputs with the offered human need categories, using as resource ConceptNet (Speer and Havasi, 2012), a large commonsense knowledge inventory.

Identifying contextually relevant information from such a large knowledge base is a non-trivial task. We propose an effective two-step method to extract multi-hop knowledge paths that associate concepts from the text with human need categories: (i) collect all potentially relevant knowledge relations among concepts and human needs in a subgraph for each input sentence; (ii) rank, filter and select high-quality paths using graph-based local measures and graph centrality algorithms.

4.3.2.1 Construction of Sub-graphs

ConceptNet is a graph $G = (V, E)$ whose nodes are concepts and edges are relations between concepts (e.g. CAUSES, MOTIVATEDBY). For each sentence S we induce a subgraph $G' = (V', E')$ where V' comprises all concepts $c \in V$ that appear in S and the directly preceding sentence in context Cxt. V' also includes all concepts $c \in V$ that correspond to one of the human need categories in our label set Z. Fig. 4.3 shows an example.

The sub-graph is constructed as follows:

Shortest Paths: In a first step, we find all shortest paths p' from ConceptNet that connect any concept $c_i \in V'$ to any other concept $c_j \in V'$ and to each human needs concept $z \in Z$. We further include in V' all the concepts $c \in V$ which are contained in the above shortest paths p'.

Neighbours: To better represent the meaning of the concepts in V', we further include in V' all concepts $c \in V$ that are directly connected to any $c \in V'$ that is not already included in V'.

Sub-graph: We finally construct a connected sub-graph $G' = (V', E')$ from V' by defining E' as the set of all ConceptNet edges $e \in E$ that directly connect any pair of concepts $(c_i, c_j) \in V'$.

Overall, we obtain a sub-graph that contains relations and concepts which are supposed to be useful to “explain" *why* and *how strongly* concepts c_i that appear in the sentence and context are associated with any of the human needs $z \in Z$.

4.3.2.2 Ranking and Selecting Multi-hop Paths

We could use all possible paths p contained in the sub-graph G', connecting concepts c_i from the text and human needs concepts z contained in G', as additional evidence to predict suitable human need categories.

But not all of them may be relevant. In order to select the most relevant paths, we propose a two-step method: (i) we score each vertex with a score (*Vscore*) that reflects its importance in the sub-graph and on the basis of the vertices' *Vscores* we determine a path score *Pscore*, as shown in Figure 4.3; (ii) we select the top-k paths with respect to the computed path score (*Pscore*) .

(i) Vertex Scores and Path Scores: We hypothesize that the most useful commonsense relation paths should include vertices that are *important* with respect to the entire extracted subgraph. We measure the importance of a vertex using different local graph measures: the *closeness centrality measure, page rank* or *personalized page rank*.

Closeness Centrality (CC) (Bavelas, 1950) reflects how close a vertex is to all other vertices in the given graph. It measures the average length of the shortest paths between a given vertex v_i and all other vertices in the given graph G'. In a connected graph, the closeness centrality $CC(v_i)$ of a vertex $v_i \in G'$ is computed as

$$Vscore_{CC}(v_i) = \frac{\mid V' \mid}{\sum_j d\left(v_j, v_i\right)} \tag{4.10}$$

where $\mid V' \mid$ represents the number of vertices in the graph G' and $d(v_j, v_i)$ represents the length of the shortest path between v_i and v_j. For each path we compute the normalized sum of *Vscore*$_X$ of all vertices v_j contained in the path, for any measure $X \in \{CC, PR, PPR\}$.

$$Pscore_X = \frac{\sum_j Vscore_X(v_j)}{N} \tag{4.11}$$

We rank the paths according to their *Pscore*$_{CC}$, assuming that relevant paths will contain vertices that are close to the center of the sub-graph G'.

PageRank (PR) (Brin and Page, 1998) is a graph centrality algorithm that measures the relative importance of a vertex in a graph. The PageRank score of a vertex $v_i \in G'$ is computed as:

$$Vscore_{PR}(v_i) = \alpha \sum_j u_{ji} \frac{v_j}{L_j} + \frac{1-\alpha}{n} \tag{4.12}$$

where $L_j = \sum_i u_{ji}$ is the number of neighbors of vertex j, α is a damping factor representing the probability of jumping from a given vertex v_i to another random vertex in the graph and

n represents the number of vertices in G'. We calculate $Pscore_{PR}$ using Eq. 4.11 and order the paths according to their $Pscore_{PR}$, assuming that relevant paths will contain vertices with high relevance, as reflected by a high number of incoming edges.

Personalized PageRank (PPR) (Haveliwala, 2002) is used to determine the importance of a vertex with respect to a certain topic (set of vertices). Instead of assigning equal probability for a random jump $\frac{1-\alpha}{n}$, PPR assigns stronger probability to certain vertices to prefer topical vertices. The PPR score of a vertex $v \in G'$ is computed as:

$$Vscore_{PPR}(v_i) = \alpha \sum_j u_{ji} \frac{v_j}{L_j} + (1-\alpha)\, T \tag{4.13}$$

where $T = \frac{1}{|T_j|}$ if nodes v_i belongs to topic T_j and otherwise $T = 0$. In our setting, T_j will contain concepts from the text and human needs, to assign them higher probabilities. We calculate $Pscore_{PPR}$ using Eq. 4.11 and order the paths according to their scores, assuming that relevant paths should contain vertices holding importance with respect to vertices representing concepts from the text and human needs.

(ii) Path Selection: We rank knowledge paths based on their *Pscore* using the above relevance measures, and construct ranked lists of paths of two types: (i) paths connecting a human needs concept $z \in Z$ to a concept mentioned in the text (p_{c-z}) [2] and (ii) paths connecting concepts in the text (p_{c-c}) [3]. Ranked lists of paths are constructed individually for concepts that constitute the start or endpoint of a path: a human needs concept for p_{c-z} or any concept from the text for p_{c-c}. Figure 4.3 illustrates an example where the character *Stewart* felt *joy* after winning a gold medal. The annotated human need label is *status*. We show the paths selected by our algorithm that connect concepts from the text and the human need *status*. We select the top-k paths of type p_{c-z} for each human need to capture relevant knowledge about human needs in relation to concepts in the text. Similarly, we select the top-k paths of type p_{c-c} for each c_i to capture relevant knowledge about the text (not shown in Fig. 3).

4.4 Integrating Multi-Hop Commonsense Knowledge

In this section, we extend our model with a gated knowledge integration mechanism to incorporate relevant multi-hop commonsense knowledge paths for predicting human needs. An overview of the model is given in Figure 4.4. We have seen how to obtain a ranked list of commonsense knowledge paths from a subgraph extracted from ConceptNet that

[2] p_{c-z} denotes path connecting a human needs concept $z \in Z$ and a concept c mentioned in the text.
[3] p_{c-c} denotes path connecting a concept c and another concept c mentioned in the text.

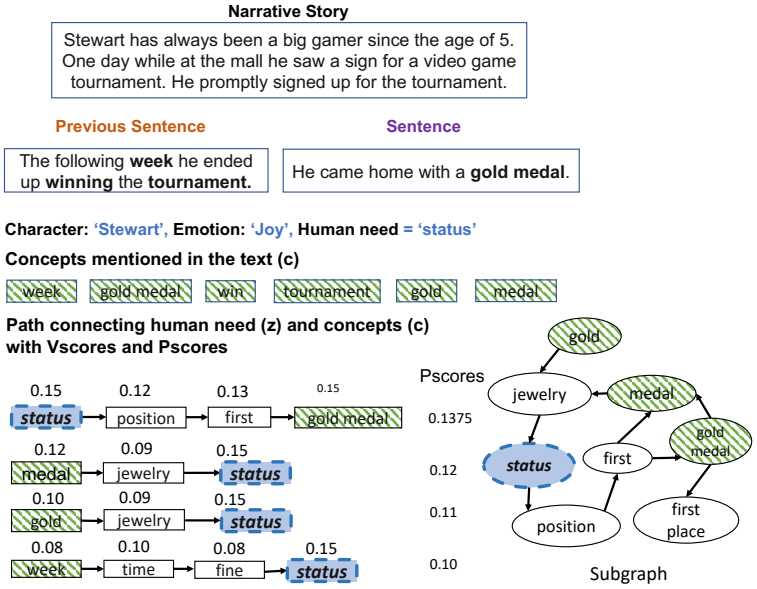


Fig. 4.3 Illustration of commonsense path selection. Top: Context and sentence, Bottom: Selected knowledge paths with *Vscores* and *Pscores* (left) and the corresponding subgraph. Concepts from the text are marked with green dashed lines; blue boxes show the human need label *status* assigned to *Stewart*.

connect concepts from the textual input and possible human needs categories that are the system's classification targets. Our intuition is that the extracted commonsense knowledge paths will provide useful evidence for our model to *link the content expressed in the text to appropriate human need categories*. Paths that are selected by the model as a relevant connection between the input text and the labeled human needs concept can thus provide *explanations* for emotions or goals expressed in the text *in view of a human needs category*. We thus integrate these knowledge paths into our model, (i) to help the model making correct predictions and (ii) to provide explanations of emotions expressed in the text in view of different human needs categories. For each input, we represent the extracted ranked list of n commonsense knowledge paths p as a list $cr^{k,1}, cr^{k,2}, ..., cr^{k,n}$, where each $cr^{k,i}_{1:l}$ represents a path consisting of concepts and relations, with l the length of the path. We embed all concepts and relations in $cr^{k,i}_{1:l}$ with pretrained GloVe (Pennington et al., 2014) embeddings.

Encoding Layer: We use a single-layer BiLSTM to obtain encodings ($h^{k,i}$) for each knowledge path

$$h^{k,i} = \text{BiLSTM}(e^{k,i}_{1:n}) \tag{4.14}$$

where h^k represents the output of the BiLSTM for the knowledge path and i its the ranking index.

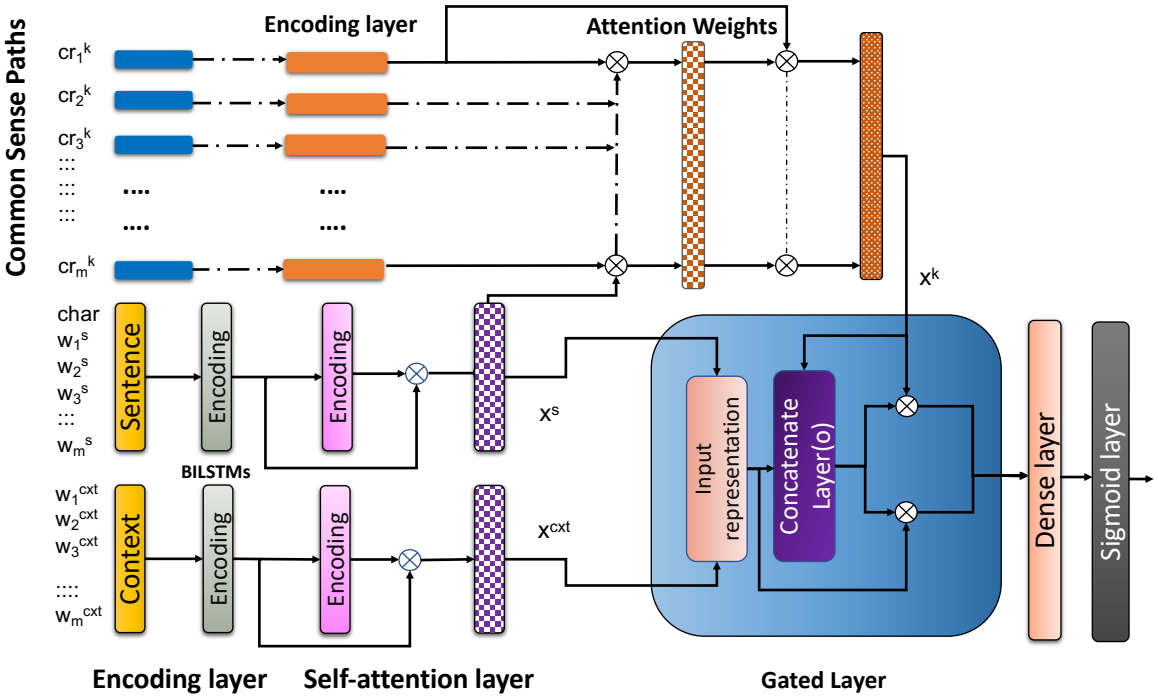


Fig. 4.4 Attention over multi-hop knowledge paths.

Attention Layer: We use an attention layer, where each encoded commonsense knowledge path interacts with the sentence representation x^s to receive attention weights ($\hat{h}^{k,i}$):

$$\widetilde{h}^{k,i} = \sigma(x^s h^{k,i}), \quad \hat{h}^{k,i} = \frac{\widetilde{h}^{k,i}}{\sum_{i=1}^{N} \widetilde{h}^{k,i}} \tag{4.15}$$

In Eq. 4.15, we use sigmoid to calculate the attention weights, similar to Eq. 4.6. However, this time we compute attention to highlight which knowledge paths are important for a given input representation (x^s being the final state hidden representation over the input sentence, Eq. 7).

To obtain the sentence-aware commonsense knowledge representation x^k, we pass the output of the attention layer through a feedforward layer. W_k, b_k are trainable parameters.

$$x^k = ReLU(W_k(\sum_{i=1}^{N} \hat{h}^{k,i} h^{k,i}) + b_k) \tag{4.16}$$

Classification	**Train**	**Dev**	**Test**
Reiss	5432	1469	5368
Reiss without *belonging* class	5431	1469	5366
Maslow	6873	1882	6821

Table 4.1 Dataset Statistics: nb. of instances (sentences with annotated characters and human need labels).

4.4.1 Distilling Knowledge into the Model

In order to incorporate the selected and weighted knowledge into the model, we concatenate the sentence x^s, context x^{cxt} and knowledge x^k representation and pass it through a FF layer.

$$o_i = ReLU(W_z[x_i^s; x_i^{cxt}; x_i^k] + b_z) \tag{4.17}$$

We employ a gating mechanism to allow the model to selectively incorporate relevant information from commonsense knowledge x^k and from the joint input representation y_i (see Eq. 4.9) *separately*. We finally pass it to a logistic regression classifier to predict a binary label for each class z in the set Z of category labels

$$z_i = \sigma(W_{\widetilde{y}_z}(o_i \odot y_i + o_i \odot x_i^k) + b_{\widetilde{y}_z}) \tag{4.18}$$

where $\odot$ represents element-wise multiplication, $b_{\widetilde{y}_z}$, $W_{\widetilde{y}_z}$ are trainable parameters.

4.5 Experimental Setup

Dataset: We evaluate our model on the *Modeling Naive Psychology of Characters in Simple Commonsense Stories (MNPCSCS)* dataset (Rashkin et al., 2018b). It contains narrative stories where each sentence is annotated with a character and a set of human need categories from two inventories: Maslow's (with five coarse-grained) and Reiss's (with 19 fine-grained) categories (Reiss's labels are considered as sub-categories of Maslow's). The data contains the original worker annotations. Following prior work we select the annotations that display the "majority label" i.e., categories voted on by ≥ 2 workers. Since no training data is available, similar to prior work we use a portion of the devset as training data, by performing a random split, using 80% of the data to train the classifier, and 20% to tune parameters. Data statistics is reported in Table 4.1.

Rashkin et al. (2018b) report that there is low annotator agreement i.a. between the *belonging* and the *approval* class. We also find high co-occurrence of the *belonging, approval*

and *social contact* classes, where *belonging* and *social contact* both pertain to the Maslow class *Love/belonging* while *approval* belongs to the Maslow class *Esteem*. This indicates that *belonging* interacts with *Love/belonging* and *Esteem* in relation to social contact. We further observed during our study that in the Reiss dataset the number of instances annotated with the *belonging* class is very low (no. of instances in training is 24, and in dev 5). The performance for this class is thus severely hampered, with 4.7 F_1 score for BiLSTM+Self-Attention and 7.1 F_1 score for BiLSTM+Self-Attention+Knowledge. After establishing benchmark results with prior work (cf. Table 4.2, including *belonging*), we perform all further experiments with a reduced Reiss dataset, by eliminating the *belonging* class from all instances. This impacts the overall number of instances only slightly: by *one* instance for training and *two* instances for test, as shown in Table 4.1.

Training: During training we minimize the weighted binary cross entropy loss,

$$L = \sum_{z=1}^{Z} w_z y_z log\widetilde{y}_z + (1 - w_z)(1 - y_z) log(1 - \widetilde{y}_z) \tag{4.19}$$

$$w_z = \frac{1}{1 - exp^{-\sqrt{P(y_z)}}} \tag{4.20}$$

where Z is the number of class labels in the classification tasks and w_z is the weight. $P(y_z)$ is the marginal class probability of a positive label for z in the training set.

Embeddings: To compare our model with prior work we experiment with pretrained GloVe (100d) embeddings (Pennington et al., 2014). Otherwise we used GloVe (300d) and pretrained ELMo embeddings (Peters et al., 2018) to train our model.

Hyperparameters for Knowledge Inclusion: We compute ranked lists of knowledge paths of two types: p_{c-z} and p_{c-c}. We use the top-3 p_{c-z} paths for each z using our best ranking strategy (Closeness Centrality + Personalized PageRank) in our best system results (Tables 4.2, 4.3, 4.5), and also considered paths p_{c-c} (top-3 per pair) when evaluating different path selection strategies (Table 4.4).

Evaluation Metrics: We predict a binary label for each class using a binary classifier so the prediction of each label is conditionally independent of the other classes given a context representation of the sentence. In all prediction tasks we report the micro-averaged Precision (P), Recall (R) and F_1 scores by counting the number of positive instances across all of the categories. All reported results are averaged over five runs. More information on the dataset, metrics and all other training details are given in the Supplement.

		Reiss			Maslow		
Model	**WE**	**P**	**R**	**F1**	**P**	**R**	**F1**
BiLSTM$^{\diamond}$	G_{100d}	18.35	27.61	22.05	31.29	33.85	32.52
CNN$^{\diamond}$	G_{100d}	18.89	31.22	23.54	27.47	41.01	32.09
REN$^{\diamond}$	G_{100d}	16.79	22.20	19.12	26.24	42.14	32.34
NPN$^{\diamond}$	G_{100d}	13.13	26.44	17.55	24.27	44.16	31.33
BM	G_{100d}	25.08	28.25	26.57	47.65	60.98	53.54
BM + K$^{\clubsuit}$	G_{100d}	**28.47**	**39.13**	**32.96**	**50.54**	**64.54**	**56.69**
BM	ELMo	29.50	**44.28**	$35.41_{\pm 0.23}$	53.86	**67.23**	$59.81_{\pm 0.23}$
BM + K$^{\clubsuit}$	ELMo	**31.74**	43.51	$\mathbf{36.70}_{\pm 0.14}$	**57.90**	66.07	$\mathbf{61.72}_{\pm 0.11}$
BM*	ELMo	31.45	44.29	37.70			
BM + K$^{\star\clubsuit}$	ELMo	36.76	42.53	39.44			

Table 4.2 Multi-label Classification Results: $^{\diamond}$: results in Rashkin et al.; *: w/o *belonging*; BM: BiLSTM+Self-Att.; +K:w/ knowledge, $^{\clubsuit}$:ranking method CC+PPR.

4.6 Results

Our experiment results are summarized in Table 4.2. We benchmark our baseline BiLSTM+Self-Attention model (BM, BM w/ knowledge) against the models proposed in Rashkin et al. (2018b): a BiLSTM and a CNN model, and models based on the recurrent entity network (REN) (Henaff et al., 2016) and neural process networks (NPN) (Bosselut et al., 2017). The latter differ from the basic encoding models (BiLSTM, CNN) and our own models by explicitly modeling entities. We find that our baseline model BM outperforms all prior work, achieving new state-of-the-art results. For Maslow we show improvement of 21.02 pp. F_1 score. For BM+K this yields a boost of 6.39 and 3.15 pp. F_1 score for Reiss and Maslow, respectively. When using ELMo with BM we see an improvement in recall. However, adding knowledge on top improves the precision by 2.24 and 4.04 pp. for Reiss and Maslow. In all cases, injecting knowledge improves the model's precision and F_1 score.

Table 4.2 (bottom) presents results for the reduced dataset, after eliminating Reiss' label *belonging*. Since *belonging* is a rare class, we observe further improvements. We see the same trend: adding knowledge improves the precision of the model.

4.6.1 Model Ablations

To obtain better insight into the contributions of individual components of our models, we perform an ablation study (Table 4.3). Here and in all later experiments we use richer (300d) GloVe embeddings and the dataset w/o *belonging*. We show results including and not including self-attention and knowledge components. We find that using self-attention over

WE	Atten	K	Gated	P	R	F1
G_{300d}	-	-	-	23.31	34.69	27.89
G_{300d}	✓	-	-	26.09	35.59	30.11
G_{300d}	✓	✓	-	27.99	37.73	32.14
G_{300d}	✓	✓	✓	**28.65**	**39.42**	**33.19**
ELMo	-	-	-	32.35	42.66	36.80
ELMo	✓	-	-	31.45	44.29	37.70
ELMo	✓	✓	-	32.65	**45.60**	38.05
ELMo	✓	✓	✓	**36.76**	42.53	**39.44**

Table 4.3 Model ablations for Reiss Classification on *MNPCSCS* dataset w/o *belonging*.

Path	Ranking	P	R	F1
S+M(P_{c-z}+ P_{c-c})	None	32.51	42.70	36.90
S+M(P_{c-z}+ P_{c-c})	Random	31.63	43.35	36.57
Single Hop(P_{c-z})	CC + PPR	33.00	44.63	37.94
S+M(P_{c-c} + P_{c-z})	CC + PPR	35.30	44.11	39.21
S+M(P_{c-z})	CC	33.45	**47.93**	39.40
S+M(P_{c-z})	PR	35.51	42.82	38.82
S+M(P_{c-z})	PPR	36.23	43.09	39.34
S+M(P_{c-z})	CC + PPR	**36.76**	42.53	**39.44**

Table 4.4 Results for different path selection strategies on *MNPCSCS* w/o *belonging*; S+M:Single+Multi hop.

sentences and contexts is highly effective, which indicates that learning how much each token contributes helps the model to improve performance. We observe that integrating knowledge improves the overall F_1 score and yields a gain in precision with ELMo. Further, integrating knowledge using the gating mechanism we see a considerable increase of 3.58 and 1.74 pp. F_1 score improvement over our baseline model for GloVe and ELMo representations respectively.

4.6.2 Commonsense Path Selection

We further examine model performance for (i) different variants of selecting commonsense knowledge, including (ii) the effectiveness of the relevance ranking strategies discussed in §4.3.2.2. In Table 4.4, rows 3-4 use our best ranking method: CC+PPR; rows 5-8 show results when using the top-3 ranked p_{c-z} paths for each human need z with different ranking measures. *None* shows results when no selection is applied to the set of extracted knowledge

paths (i.e., using all possible paths from p_{c-z} and p_{c-c}). *Random* randomly selects 3 paths for each human need from the set of paths used in *None*. This yields only a slight drop in performance. This suggests that not every path is relevant. We evaluate the performance when only considering single-hop paths (now top-3 ranked using CC+PPR) *(Single-Hop)*. We see an improvement over random paths and no selection, but not important enough. In contrast, using both single and multi-hop paths in conjunction with relevance ranking improves the performance considerably (rows 4-8). This demonstrates that multi-hop paths are informative. We also experimented with p_{c-c}+p_{c-z}. We find improvement in recall, however the overall performance decreases by 0.2 F_1 score compared to paths p_{c-z} ranked using CC + PPR. Among different ranking measures *precision* for Personalized PageRank performs best in comparison with CC and PR in isolation, and recall for CC in isolation is highest. Combining CC and PPR yields the best results among the different ranking strategies (rows 5-8).

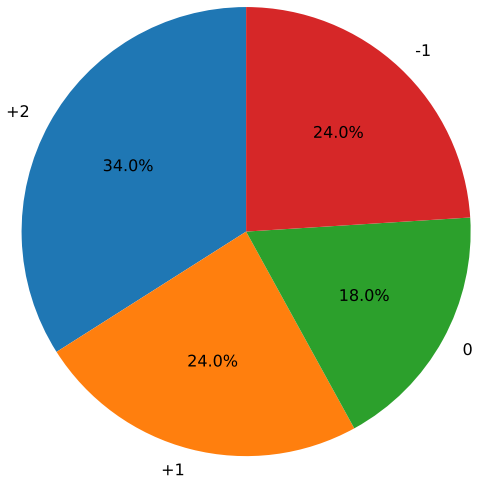


Fig. 4.5 Human evaluation: Distribution of scores.

4.7 Analysis

4.7.1 Performance per Human Need Categories

We examined the model performance on each category (cf. Figure 4.6). The model performs well for basic needs like *food*, *safety*, *health*, *romance*, etc. We note that inclusion of knowledge improves the performance for most classes (only 5 classes do not profit from knowledge compared to only using ELMo), especially for labels which are rare like *honor,*

Model	WE	P	R	F1
BM	ELMo	33.39	45.15	38.39
BM+K	ELMo	36.36	44.02	39.83

Table 4.5 Multi-label classification on *MNPCSCS* w/o *belonging class* and w/o context (1^{st} sentence only)

.

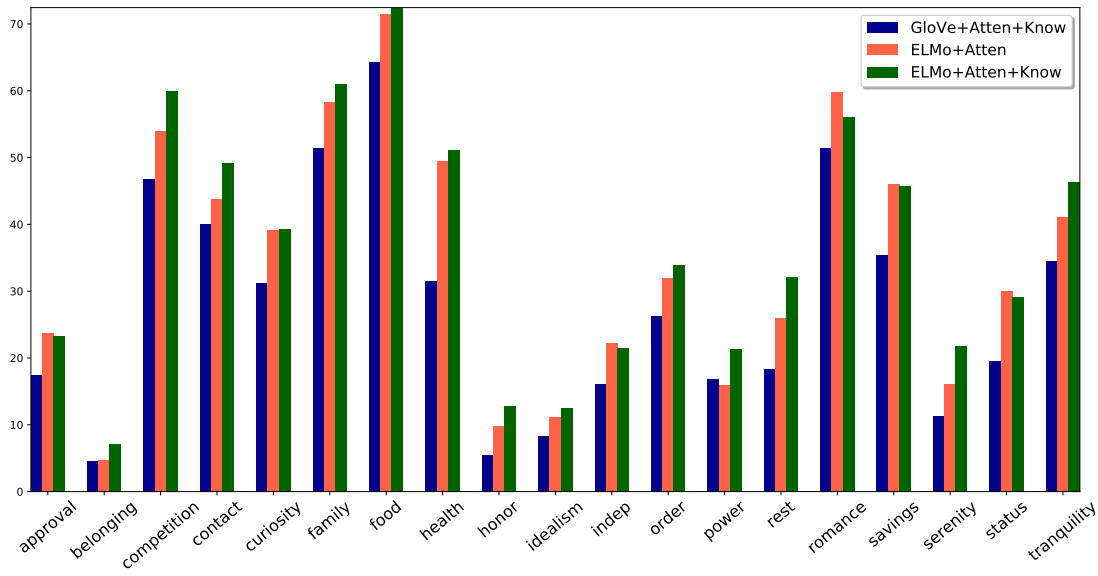
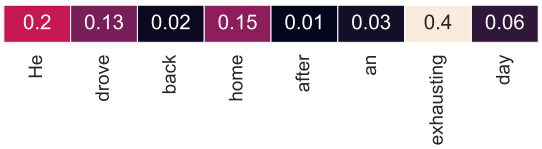


Fig. 4.6 Best model's performance per human needs (F_1 scores) for Reiss on *MNPCSCS* dataset.

idealism, power. We also found that the annotated labels can be subjective. For instance, *Tom lost his job* is annotated with *order* while our model predicts *savings*, which we consider to be correct. Similar to Rashkin et al. (2018b) we observe that preceding context helps the model to better predict the characters' needs, e.g., *Context: Erica's [..] class had a reading challenge [..]. If she was able to read 50 books [..] she won a pizza party!; Sentence: She read a book every day for the entire semester* is annotated with *competition*. Without context the predicted label is *curiosity*, however when including context, the model predicts *competition, curiosity*. We measure the model's performance when applying it only to the first sentence of each story (i.e., without the context). As shown in Table 4.5, also in this setting the inclusion of knowledge improves the performance.

Context: Timmy had to renew his driver's license. He went to his local DMV. He waited in line for nearly 2 hours. He took a new picture for his driver's license.
Sentence: He drove back home after an exhausting day.
True Label: *rest*
Predicted Label (BM): *status, approval, order*
Predicted Label (BM+K): *rest*



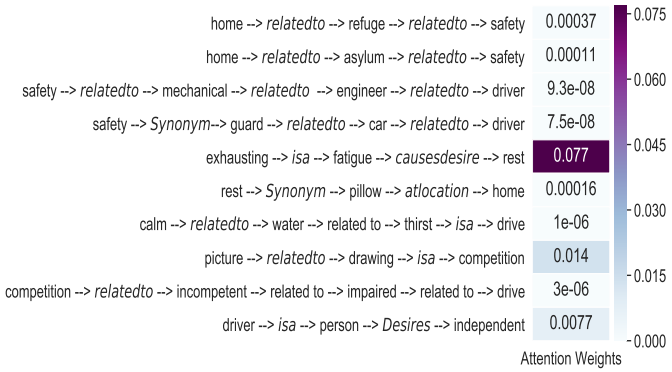


Fig. 4.7 Interpreting the attention weights on sentence representation and selected commonsense paths.

4.7.2 Human Evaluation of Extracted Paths

We conduct human evaluation to test the effectiveness and relevance of the extracted commonsense knowledge paths. We randomly selected 50 extracted knowledge paths that contain the gold label (using CC+PPR for ranking). We asked three expert evaluators to decide whether the paths are relevant to provide information about the missing links between the concepts in the sentence and the human need (gold label). We asked them to assign scores according to the following definitions:

+2: the path specifies perfectly relevant information to provide the missing link between the concepts in the sentence and the human need.

+1: the path contains a sub-path that specifies relevant information to provide the missing links between the concepts in the sentence and the human need.

0: when the path is irrelevant but the starting and the ending nodes stand in a relation that is relevant to link the sentence and the expressed human need. (In this case, either the

Case 1: Inclusion of knowledge path improves the performance when there is no context.

Context: No Context
Sentence: Tina was out for a walk in the street.
True Label: *Health*
Predicted without Knowledge: *Serenity*
Predicted with Knowledge : *Health*

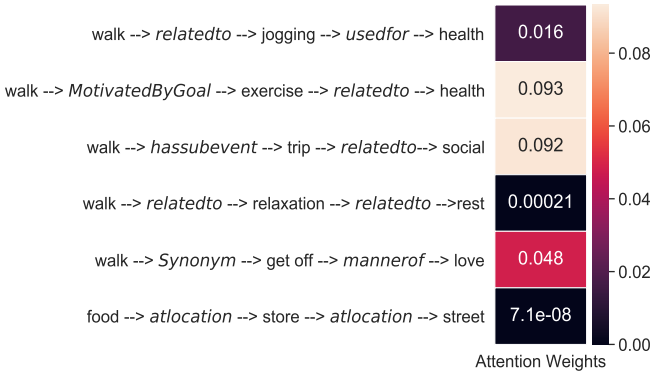


Fig. 4.8 Example 1: Visualizing the attention weights of the input sentence and of selected commonsense paths.

path selected by our algorithm is not relevant or there is no relevant path connecting the nodes given the context.)

-1: the path is completely irrelevant.

The inter-annotator agreement had a Fleiss' κ= 0.76. Figure 4.5 depicts the distribution of assigned scores (based on the majority class). The result for this evaluation shows that in 34% of the cases computed on the basis of majority agreement, our algorithm was able to select a relevant commonsense path.

We study the visualization of attention distributions produced by our model. We provide examples for different scenarios. Here we show the results found by our best model i.e., BiLSTM+Self-Attention+Gated-Knowledge with CC+PPR as path selection method.

4.7.3 Interpretabilty

Finally we study the learned attention distributions of the interactions between sentence representation and knowledge paths, in order to interpret how knowledge is employed to make predictions. Visualization of the attention maps gives evidence of the ability of the

Case 2: Inclusion of knowledge paths improves the precision of the model.

Context: No Context
Sentence: Noah wanted to play golf against Nick.
True Label: *Competition*
Predicted without Knowledge: *Competition, Curiosity*
Predicted with Knowledge: *Competition*

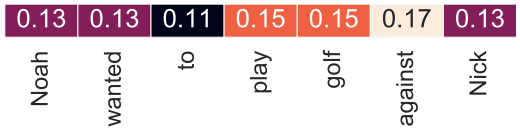
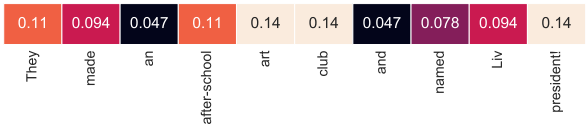


Fig. 4.9 Example 2: Visualizing the attention weights of the input sentence and of selected commonsense paths.

model to capture relevant knowledge that connects human needs to the input text. The model provides interpretability in two ways: by selecting tokens from the input text using Eq.4.6 and by choosing knowledge paths from the imported knowledge using Eq.4.15. Figure 4.7-4.10 shows some examples where including knowledge paths helped the model to predict the correct human need category. For example, in figure 4.7, the attention map depicts which exact paths are selected to make the prediction. In this example, the model correctly picks up the token *"exhausting"* from the input sentence and the knowledge path *"exhausting is a fatigue causes desire rest"*. We present more examples of extracted knowledge and its attention visualization with different scenarios like (**case1**) when no context is given, (**case2**) when the precision of the model improved due to knowledge incorporation, (**case3**) when the recall of the model improved due to knowledge incorporation. Finally, in Figure 4.11 we see an example when our model fails to attend to the relevant knowledge path. Interestingly, the graph-based ranking and selection algorithm were able to extract a relevant knowledge path, but the neural model failed to correctly pick (attend to) the correct path. One intuitive reason can be there are training data size for the class "serenity" is small compared to classes.

Case 3: Inclusion of knowledge paths improves the recall of the model

Context: Liv was a budding artist and she loved painting. She wanted to go to art classes, but her school didn't offer any!, So Liv got together with her friends and began brainstorming. They decided to form their own art group at the high school.
Sentence: They made an after-school art club and named Liv president!
True Label: *Independent, Curiosity, Contact*
Predicted without Knowledge: *Contact*
Predicted with Knowledge : *Independent, Curiosity, Contact*



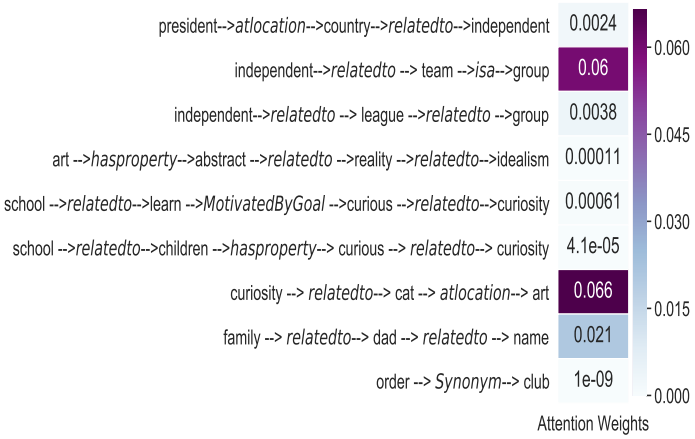


Fig. 4.10 Example 3: Visualizing the attention weights of the input sentence and of selected commonsense paths.

4.8 Summary

In this Chapter, we introduced a new method to rank and select multi-hop relation paths from a commonsense knowledge resource using graph-based algorithms. Our end-to-end model incorporates multi-hop knowledge paths to predict human needs. We show that due to the attention mechanism we can analyze the knowledge paths that the model considers in prediction. This enhances transparency and interpretability of the model. We show that implicit knowledge is crucial for a better predicting human needs and motives.

In our ablation study we observe that integrating knowledge and self-attention have a complementary impact of the model performance. Particularly, we notice that self-attention

Case 4: In this case our model fails to attend to the relevant path. Although the graph-based ranking and selection algorithm were able to extract a relevant knowledge path, the neural model fails to correctly pick (attend to) the correct path.

Context: Tom was driving his car. He wanted to take a scenic way home. He deliberately passed his exit. Tom saw many beautiful trees.
Sentence: Tom took the scenic way home.
True Label: *Serenity*
Predicted without Knowledge: *Independent, Curiosity*
Predicted with Knowledge : *Family, Independent, Curiosity, Serenity*

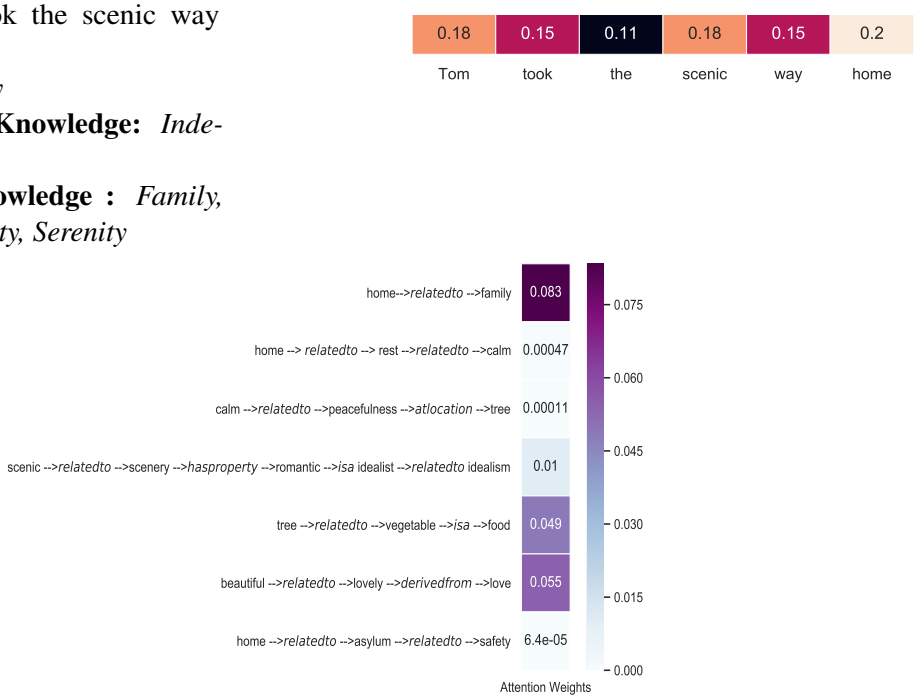


Fig. 4.11 Example 4: Visualizing the attention weights of the input sentence and of selected commonsense paths.

improves the *precision* of the model (in relative terms) and the knowledge improves the *recall* of the model. In our quantitative analysis we find that combination of closeness centrality and personalized page rank yields the best results among the different ranking strategies. To show that our ranking strategy is task-agnostic, we apply our method on *argumentation classification task*[1].

[1]More details about our experiments and results on argumentation classification task are in Appendix A

Chapter 5

Generating Hypothetical Events for Abductive Inference

"Abduction is the process of forming explanatory hypotheses. It is the only logical operation which introduces any new idea"

– Charles Sanders Peirce

In the previous chapter, we investigated the role of commonsense knowledge in inferences about the dynamics of mental states in stories. In this chapter, we investigate further the dynamics of story events. We study how learning about what social event follows another event impacts abductive commonsense reasoning. The first section of this chapter gives motivation and introduction to abductive commonsense reasoning tasks. The second section presents a method to learn about future events from a given social situation. In the third section, we present supervised and unsupervised methods for leveraging the knowledge about future events to support abductive commonsense reasoning tasks. Finally, we end the chapter with our method's automatic and human evaluation. This chapter is based on work originally published in Paul and Frank (2021b).

5.1 Abductive Inference

Abductive reasoning (AR) is inference to the best explanation. It typically starts from an incomplete set of observations about everyday situations and comes up with what can be considered the most likely possible explanation given these observations (Pople, 1973; Douven, 2017). One of the key characteristics that make abductive reasoning more chal-

lenging and distinct from other types of reasoning is its non-monotonic character (Strasser and Antonelli, 2019) i.e., even the most likely explanations are not necessarily correct. For example, in Figure 5.1, the most likely explanation for *Observation 1: "wet grass outside my house"* is that *"it has been raining"*. However, when a new piece of information (observation or evidence) becomes available, the explanation must possibly be retracted, *showing the defeasible character of abduction*. With the new observation (*"the sprinkler was switched on*") the most plausible explanation changes to *"Sprinkler caused the grass to be wet"*. Humans, in such situations, could induce or validate such abductive inferences by performing hypothetical reasoning (such as *"What would happen if the sprinkler was switched on?"*) to arrive at a plausible explanation for *"wet grass outside my house"*.

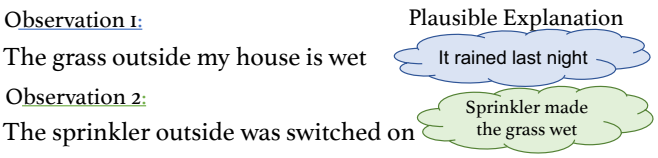


Fig. 5.1 Motivational example illustrating Abductive Reasoning and its non-monotonic character.

There has been longstanding work on theories of abductive reasoning (Peirce, 1903, 1965a,b; Kuipers, 1992, 2013). Researchers have applied various frameworks, some focused on pure logical frameworks (Pople, 1973; Kakas et al., 1992), some on probabilistic frameworks (Pearl, 1988), and others on Markov Logics (Singla and Mooney, 2011). Recently, moving away from logic-based abductive reasoning, Bhagavatula et al. (2020) proposed to study language-based abductive reasoning. They introduced two tasks: *Abductive Natural Language Inference (αNLI)* and *Generation (αNLG)*.

In this chapter, we focus on the **αNLI task** (Bhagavatula et al., 2020), where given two observations (O_1 at time t_1, O_2 at time t_2, with $t_1 < t_2$) as an incomplete context, the task is to predict which of two given hypothesized events (H_1 or H_2) is more plausible to have happened between O_1 and O_2. Figure 5.2 illustrates this with an example: given observations O_1*:"Priya decided to try a new restaurant."* and O_2*: "Priya thought her food was delicious."*, the task is to predict whether H_1 or H_2 is the more plausible explanation given observations O_1 and O_2. Both H_1 and H_2 are different plausible hypothetical situations that can evolve from the same observation (premise) O_1.

We hypothesize that learning how different hypothetical scenarios (H_1 and H_2) can result in different outcomes (e.g., $O_2^{H_j}$, Fig. 5.2) can help in performing abductive inference. In order to decide which H_i, is *more plausible* given observations, we assume each H_i to be *true* and generate a *possible next event* $O_2^{H_i}$ for each of them independently (e.g.: *What will happen if Priya's ordered food was microwaved and precooked?*). We then compare the

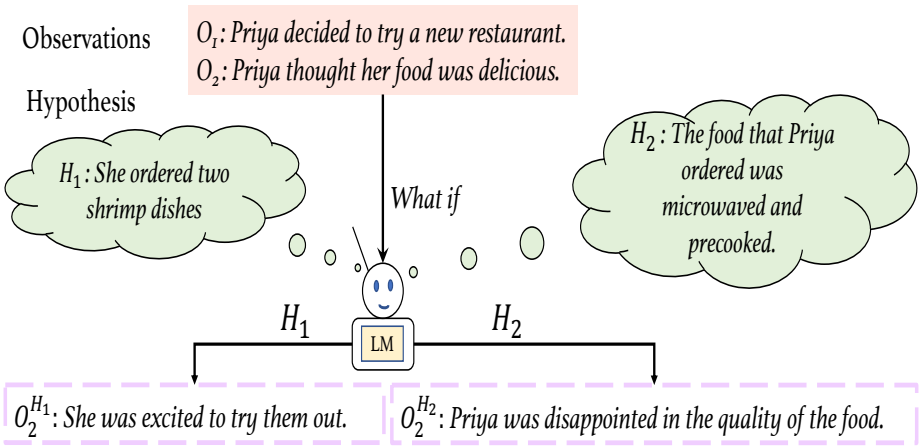


Fig. 5.2 Motivational example for αNLI : The top box (red) shows the observations and two callout clouds (green) contain the hypotheses. The implications ($O_i^{H_i}$) – generated by the LM conditioned on each hypothesis and the observations – are given in pink colored boxes.

generated sentences ($O_2^{H_1}$, $O_2^{H_2}$ in Fig. 5.2) to what has been observed (O_2) and choose as most plausible hypothesis the one whose implication is closest to observation O_2.

We design a language model ($LM_{\mathcal{I}}$) which, given observations and a hypothesis, generates a possible event that could happen next, given one hypothesis. In order to train this language model, we use the TIMETRAVEL (TT) corpus (Qin et al., 2019) (a subpart of the *ROCStories* corpus[1]). We utilize the $LM_{\mathcal{I}}$ model to generate a possible next event for each hypothesis, given the observations. We then propose a multi-task learning model $\mathcal{MTL}$ that jointly chooses from the generated possible next events ($O_2^{H_1}$ or $O_2^{H_2}$) the one most similar to the observation O_2 and predicts the most plausible hypothesis (H_1 or H_2).

5.2 Learning about Counterfactual Scenarios

The main idea is to learn to generate assumptions, in a given situation, about *"What could have happened (next) if we had done X?" or "What could happen (next) if we do X?"* (Bhatt and Flanagan, 2010). Figure 5.3(a) depicts the αNLI task framework. We hypothesize that getting to know *what will happen (next) if any of two hypotheses occurs*, will help us verifying which of them is more plausible (see Fig. 5.3(c)). Therefore, we encourage the model to learn how different hypothetical events (including counterfactual events) evolving from the same premise (s_1) can lead to different or similar outcomes (see Fig. 5.3(b)).

Accordingly, we teach a pre-trained GPT-2 (Radford et al., 2019) language model how to generate *a sequence of* possible subsequent events given different hypothetical situations

[1]We ensure that αNLI testing instances are held out.

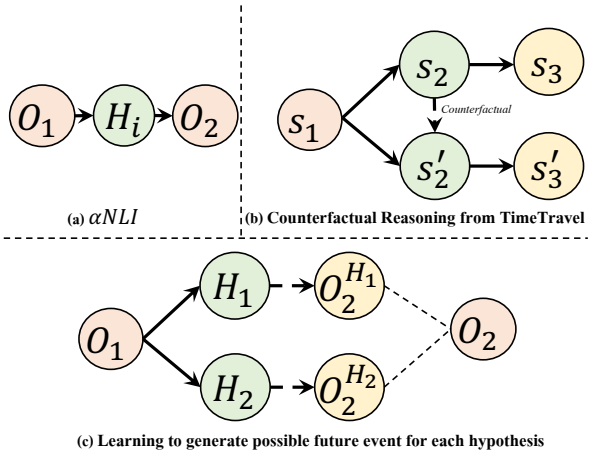


Fig. 5.3 Different reasoning schemes and settings for our task and approach. The arrows denote the direction (temporal flow) of the reasoning chain. The dotted arrow in (b) denotes the derivation of a counterfactual situation s_2' from a factual s_2. In (c), the dotted arrows denote the learned inference; the dotted lines indicate the similarity between O_2 and $O_2^{H_i}$.

in a narrative setting. Training such a model on narrative texts encourages it to learn (*latent learning*) causal and temporal relations between events. We train a conditional language model, $LM_{\mathcal{I}}$, which generates a possible event that could happen next, given some counterfactual scenarios for a given story.

We train this model on the TIMETRAVEL (TT) dataset (Qin et al., 2019), by fine-tuning GPT-2 to learn about possible next events emerging from a situation in a story, given some alternative, counterfactual event. The TT dataset consists of five-sentence instances S = (s_1,s_2,..,s_5)[2] from the ROCStories corpus[1] plus additional crowd-sourced sentences $s'_{2:5}$, where s'_2 is counterfactual[3] to s_2 from the original story[4]. There are two reasons for using the TT dataset for our purposes: a) the domains on which GPT-2 was pretrained are broad[5] and different from the domain of ROCStories, b) the model can see how alternative situations can occur starting from the same premise s_1, resulting in similar or different outcomes. Note that, although intermediate situations may be counterfactual to each other, the future outcome can still be similar to the original ending due to *causal invariance* [6].

Concretely, the language model $LM_{\mathcal{I}}$ reads the premise (s_1) and the alternative event(s) (s_2 or s'_2), the masked token (serving as a placeholder for the missing possible next event(s)

[2] s_1 = *premise*, s_2 = *initial context*, $s_{3:5}$ = original ending

[3] a counterfactual s' states something that is contrary to s

[4] During our experiments we treat them as two separate instances: S_1=($s_{1:5}$) and S_2 = (s_1,$s'_{2:5}$).

[5] GPT-2 was trained on the WebText Corpus.

[6] the future events that are invariant under the counterfactual conditions (Qin et al., 2019)

Observation (O)	**Hypothesis** (H_j)	**Generated Implications** $O_2^{H_j}$
O_1: Dotty was being very grumpy. O_2: She felt much better afterwards	H_1: Dotty ate something bad.	$O_2^{H_1}$:She started to feel sick.
	H_2:**Dotty call some close friends to chat.**	$O_2^{H_2}$: They all tried to make her happy.
O_1: Priya decided to try a new restaurant. O_2: Priya thought her food was delicious.	H_1:**She ordered two shrimp dishes.**	$O_1^{H_2}$: She was excited to try them out.
	H_2: The food that priya ordered was microwaved and precooked.	$O_2^{H_2}$: Priya was disappointed in the quality of the food.
O_1: Jim got ready for his first date. O_2: Since then, she has ignored all of Jim's text messages.	H_1: **Jim's date wasn't attracted to him.**	$O_1^{H_2}$: He tried to get her number. She never responded to him.
	H_2: Jim went on the date and said he didn't like the girl.	$O_2^{H_2}$: He told her that he didn't want to date her.
O_1: Cay had a crush on a boy in her class. O_2: He smiled at her after and said he liked her too!	H_1: **Cay sent a love note to the boy.**	$O_1^{H_2}$: The boy responded and said he liked Cay.
	H_2: She told him she did not like him.	$O_2^{H_2}$: The boy was very sad about it.
O_1: Daniel wanted to buy a toy plane, but he didn't have any money. O_2: He bought his toy plane, and kept working so he could buy another!	H_1: **He opened a lemonade stand.**	$O_1^{H_2}$: He sold lemonade for a lot of money .
	H_2: Daniel stayed home and didn't want to buy a plane.	$O_2^{H_2}$: He decided he needed to get a part time job.
O_1: Ali's mom enrolled her in a karate class. O_2: Ali was so embarrassed she didn't tell any of her friends.	H_1: **Ali did not want to take karate.**	$O_2^{H_1}$: She was afraid she would get hurt.
	H_2: Ali did horribly in her last class.	$O_2^{H_2}$: Ali was so embarrassed. She tried to hide it from her parents.

Table 5.1 Example of generated implications using $\mathcal{FI}$ model. The plausible hypothesis in each example is given in bold text.

($s_{3:i}$ or $s^{'}_{3:i}$), then the rest of the story ($s_{i+1:5}$ or $s^{'}_{i+1:5}$) and again the premise (s_1). We train the model to maximize the log-likelihood of the missing ground-truth sentence(s) ($s_{3:i}$).

$$\begin{aligned}\mathcal{L}^{LM_{\mathcal{I}}}(\beta) = log_{p_\beta}(s_{3:i}|[S]s_1,[M],s_{i+1:5},[E],[S],s_1,s_2) \\ + log_{p_\beta}(s^{'}_{3:i}|[S]s_1,[M],s^{'}_{i+1:5},[E],[S],s_1,s^{'}_2)\end{aligned} \tag{5.1}$$

where $i \in [3,4]$, s_i= $\{w_1^{s_i},..,w_n^{s_i}\}$ a sequence of tokens, $[S]$= start-of-sentence token, $[E]$= end-of-sentence token, $[M]$= mask token.

5.3 Hypothetical Events for αNLI task

We aim to investigate whether models perform better on the αNLI task when explicitly learning about events that could follow other events in a hypothetical scenario. We do so by introducing two methods $LM_{\mathcal{I}}$ + *BERTScore* and $LM_{\mathcal{I}}$ + $\mathcal{MTL}$ for unsupervised and supervised settings, respectively.

We first apply the trained model $LM_{\mathcal{I}}$ on the αNLI task, where the given observations O_1 and O_2, and alternative hypotheses H_j are fed as shown in (2) below.[7]

$$O_2^{H_j} = \beta([S], O_1, [M], O_2, [E], [S], O_1, H_j) \tag{5.2}$$

We generate a possible next event for each hypothetical event H_j, i.e., $O_2^{H_1}$ and $O_2^{H_2}$ (or: what will happen if some hypothesis H_j occurs given the observations), where $j \in [1, 2]$. Table 5.1 illustrate some examples where different $O_2^{H_j}$ are generated using $LM_{\mathcal{I}}$. One of the challenges when generating subsequent events given a hypothetical situation is that there can be infinite numbers of possible next events. Therefore, to constrain this range, we chose to give future events (O_2) as input, such that the model can generate subsequent events in a constrained context.

5.3.1 Unsupervised Setting

In this setting, we do not train any supervised model to explicitly predict which hypothesis is more plausible given the observations. Instead, we apply the fine-tuned $LM_{\mathcal{I}}$ model to the αNLI data, generate possible next events $O_2^{H_j}$ given O_1 and H_j, as described above, and measure the similarity between such possible next events ($O_2^{H_j}$) and the observation (O_2) in an unsupervised way, using *BERTScore* (BS) (Zhang et al., 2020c) [8]. Figure 5.4 represents the overview of our unsupervised model. We evaluate our hypothesis that the generated possible next event $O_2^{H_j}$ given the more plausible hypothesis H_j should be *more similar* to observation O_2. Table 5.1 illustrate some examples where H_2 is the more plausible hypothesis. We impose the constraint that for a correctly

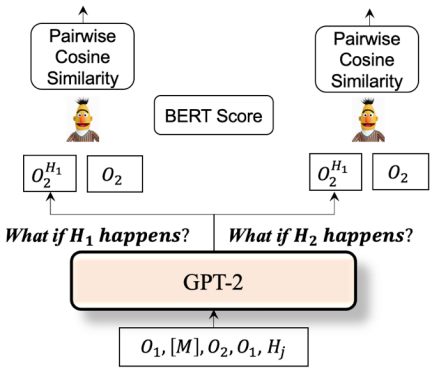


Fig. 5.4 Overview of our $LM_{\mathcal{I}}$ + *BERTScore* model for αNLI

[7] For definition of placeholders see (5.1).

[8] BERTScore is an automatic evaluation metric for text generation that leverages the pre-trained contextual embeddings from BERT and matches words in candidate and reference sentences by cosine similarity.

predicted instance BS($O_2{}^{H^+}, O_2$) > BS($O_2{}^{H^-}, O_2$) should hold, where H^+, H^- are the more plausible vs. implausible hypothesis, respectively. In Table 5.5 we show some examples of generated possible next events and their the bert scores with respect to the observation O_2.

5.3.2 Supervised Setting

In this setting, displayed in Figure 5.5, we explore the benefits of training a multi-task $\mathcal{MTL}$ model that predicts i) the most plausible hypothesis and ii) which possible next event ($O_2^{H_j}$) is more similar to the observation (O_2). Multi-task learning aims to improve the performance of a model for a task by utilizing the knowledge acquired by learning related tasks (Ruder, 2019). We *hypothesize that* a) the possible next event $O_2^{H_j}$ of the more plausible hypothesis H_j should be most similar to observation O_2, and that b) learning which possible next event is more similar supports the model in the αNLI task (*inductive transfer*). The architecture of

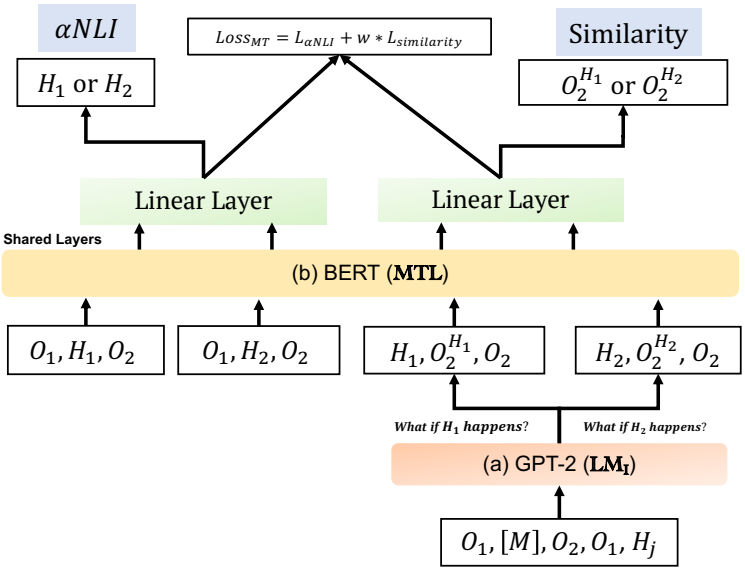


Fig. 5.5 Overview of our $LM_{\mathcal{I}}$ + $\mathcal{MTL}$ model for αNLI: (a) language model $LM_{\mathcal{I}}$ takes the input in a particular format to generate different possible next events, (b) the $\mathcal{MTL}$ model learns to predict the best explanation (H_j) and possible next events ($O_2^{H_j}$) at the same time to perform the αNLI task.

$LM_{\mathcal{I}}$ + $\mathcal{MTL}$ model is shown in Figure 5.5. The model marked (a) in Figure 5.5 depicts the $LM_{\mathcal{I}}$ model as described in §5.2. The outputs of the $LM_{\mathcal{I}}$ model, which we get from Eq. (5.2) for both hypotheses are incorporated as an input to the $\mathcal{MTL}$ model. Concretely, we feed the $\mathcal{MTL}$ classifier a sequence of tokens as stated in part (b) of Figure 5.5, and aim to compute their contextualized representations using pre-trained BERT. The input format

Task	Train	Dev	Test
αNLI	169654	1532	3059
TimeTravel (NLG)	53806	2998	–

Table 5.2 Dataset Statistics: number of instances

is described in Table 6.2. Similar to (Devlin et al., 2019), two additional tokens are added [CLS] at the start of each sequence input and [SEP] at the end of each sentence. In the shared layers (see Fig 5.5(b)), the model first transform the input sequence to a sequence of embedding vectors. Then it applies an attention mechanism that learns contextual relations between words (or sub-words) in the input sequence.

For each instance we get four [CLS] embeddings ($CLS_{H_j}, CLS_{O_2^{H_j}}; j \in [1,2]$) which are then passed through two linear layers, one for the αNLI (main task) and another for predicting the similarity (auxiliary task) between $O_2^{H_j}$ and O_2. We compute the joint loss function $\mathcal{L}$ = $\mathcal{L}_{\alpha NLI}$ + $w * \mathcal{L}_{similarity}$; where w is a trainable parameter, $\mathcal{L}_{\alpha NLI}$ and $\mathcal{L}_{similarity}$ are the loss function for the αNLI task and auxiliary task, respectively.

5.4 Experimental Setup

Data. We conduct experiments on the $\mathcal{ART}$ (Bhagavatula et al., 2020) dataset. Data statistics are given in Table 5.2. For evaluation, we measure accuracy for αNLI.

Hyperparameters. To train the $LM_{\mathcal{I}}$ model we use learning rate of $5e-05$. We decay the learning rate linearly until the end of training; batch size: 12. In the supervised setting for the α**NLI task**, we use the following set of hyperparameters for our $\mathcal{MTL}$ model with integrated $LM_{\mathcal{I}}$ model ($LM_{\mathcal{I}} + \mathcal{MTL}$): batch size: $\{8, 16\}$; epochs: $\{3, 5\}$; learning rate: $\{2e\text{-}5, 5e\text{-}6\}$. For evaluation, we measure accuracy. We use Adam Optimizer, and dropout rate = 0.1. We experimented on GPU size of 11GB and 24GB. Training is performed using cross-entropy loss. The loss function is $\mathcal{L}_{\alpha NLI}$ + $w * \mathcal{L}_{similarity}$, where w is a trainable parameter. During our experiment we initialize $w = 1$. The input format is depicted in Table 5.3. We report performance by averaging results along with the variance obtained for 5 different seeds.

Baselines. We compare to the following baseline models that we apply to the αNLI task, training them on the training portion of the $\mathcal{ART}$ dataset (cf. Table 5.2).

Input Format	Output
[CLS] O_1 [SEP] H_i [SEP] O_2 [SEP]	H_1 or H_2
[CLS] H_i [SEP] $O_2^{H_i}$ [SEP] O_2[SEP]	$O_2^{H_1}$ or $O_2^{H_2}$

Table 5.3 Input and output format for the αNLI task: [CLS] is a special token used for classification, [SEP] a delimiter.

- *ESIM + ELMo* is based on the ESIM model previously used for NLI (Chen et al., 2017). We use (a) ELMo to encode the observations and hypothesis, followed by (b) an attention layer, (c) a local inference layer, and (d) another bi-directional LSTM inference composition layer, and (e) a pooling operation,
- *Infersent* (Conneau et al., 2017) uses sentence encoding based on a bi-directional LSTM architecture with max pooling.
- *BERT* (Devlin et al., 2019) is a LM trained with a masked-language modeling (MLM) and next sentence prediction objective.

As baselines for using the $\mathcal{MTL}$ model, we replace $LM_{\mathcal{I}}$ with alternative generative LMs:

- *GPT-2 + $\mathcal{MTL}$.* In this setup, we directly use the pretrained GPT-2 model and task it to generate a next sentence conditioned on each hypothesis ($O_2^{H_i}$) without finetuning it on the TIMETRAVEL data. We then use the supervised $\mathcal{MTL}$ model to predict the most plausible hypothesis and which of the generated observations is more similar to O_2.
- *COMET + $\mathcal{MTL}$.* In this setting, we make use of inferential *if-then* knowledge from ATOMIC Sap et al. (2019a) as background knowledge. Specifically, we use COMET to generate objects with **Effect**[9] relations for each hypothesis as a textual phrase.

5.5 Results

5.5.1 Automatic Evaluation

In Table 5.4, we compare our models $LM_{\mathcal{I}}$ + BERTScore and $LM_{\mathcal{I}} + \mathcal{MTL}$ against the models proposed in Bhagavatula et al. (2020): a majority baseline, supervised models (*Infersent* and *ESIM+ELMo*), as well as *BERT*$_{Large}$. Bhagavatula et al. (2020) re-train the

[9] as a result PersonX feels; as a result PersonX wants; PersonX then

Model	Dev Acc.(%)	Test Acc.(%)
Majority (*from dev set*)$^{\diamond}$	–	50.8
$LM_{\mathcal{I}}$ + BERTScore	62.27	**60.08**
Infersent $^{\diamond}$	50.9	50.8
ESIM + ELMo $^{\diamond}$	58.2	58.8
BERT$_{Large}$ $^{\diamond}$	69.1	68.9±0.5
GPT-2 +$\mathcal{MTL}$	68.9±0.3	68.8±0.3
COMET +$\mathcal{MTL}$	69.4±0.4	69.1±0.5
$LM_{\mathcal{I}} + \mathcal{MTL}$	72.9±0.5	**72.2**±0.6
Human Performance	-	91.4

Table 5.4 Results on α**NLI task**, $\diamond$: as in Bhagavatula et al. (2020) (no unpublished leaderboard results). For each row, the best results are in bold, and performance of our models are in blue.

ESIM+ELMo and Infersent models on the $\mathcal{ART}$ dataset and fine-tuned the BERT model on the αNLI task and report the results.

We find that our **unsupervised** model with BERTScore ($LM_{\mathcal{I}}$ + BERTScore) outperforms (by $+9.28$ pp. and $+1.28$ pp.) strong ESIM+ELMo and Infersent baseline models. Table 5.5 shows some examples of our generation model $LM_{\mathcal{I}}$ along with the obtained BERTScores.

Unlike the unsupervised $LM_{\mathcal{I}}$ + BERTScore, our **supervised** $LM_{\mathcal{I}} + \mathcal{MTL}$ model also improves over the BERT$_{Large}$ baseline, by $+3.3$ pp. We can attribute the improvement to the model having been jointly trained to assess the similarity and dissimilarity of possible next events $O_2^{H_j}$ and observations (O_2) along with the αNLI task. One of the advantages of training our proposed multi-task learning ($\mathcal{MTL}$) model, instead of directly feeding the possible next events $O_2^{H_j}$ as knowledge inputs is that it adds an explainable component to the model. One can view the generated next events $O_2^{H_j}$ as natural language rationales and our multi-task model explicitly chooses one of them. Hence, the multi-task framework makes the model more expressive. Finally, we compare, for the $\mathcal{MTL}$ model, our embedded generation model $LM_{\mathcal{I}}$ to pre-trained GPT-2 and COMET. Table 5.4 shows that $LM_{\mathcal{I}}$ + $\mathcal{MTL}$ yields better performance compared to both *COMET* + $\mathcal{MTL}$ ($+3.1$ pp.) and *GPT-2* + $\mathcal{MTL}$ ($+3.4$ pp.) – the intuitive reason being that the next events generated by $LM_{\mathcal{I}}$ are more helpful than events generated using pretrained GPT-2 and objects generated by COMET.

Table 5.5 illustrates some examples where our $\mathcal{MTL}$ model not only chooses the correct hypothesis, but also a likely possible next event that is similar to the observation O_2. Interestingly, during training of $\mathcal{MTL}$ we initialize $w = 1$, and after training the model we found the w value had been adjusted to a range between 0.85 and 0.75, which intuitively shows both the effectiveness of our $LM_{\mathcal{I}}$ -generated possible next events, and their similarity to the given observations O_2.

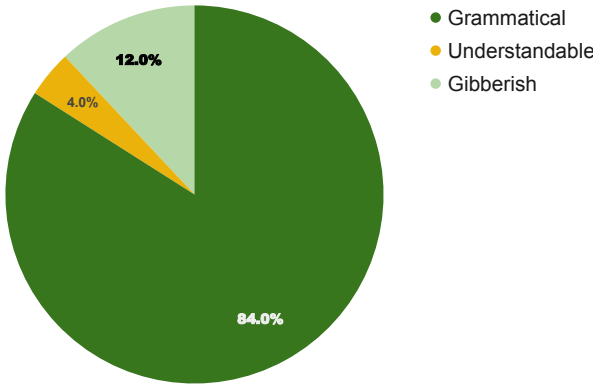


Fig. 5.6 Human evaluation of the *grammaticality* of generated sentences: ratio of i) grammatical, ii) not entirely grammatical but understandable, iii) completely not understandable sentences.

5.5.2 Manual Evaluation

Since the automatic scores only account for word-level similarity between observations and generated possible next events, we conduct a manual evaluation study, to assess the quality of sentences generated by our $LM_{\mathcal{I}}$ model.

Annotation Study on $LM_{\mathcal{I}}$ generations. The annotation was performed by three annotators with computational linguistic background. We provide each of the three annotators with observations, hypotheses and sentences, as produced by our $LM_{\mathcal{I}}$ model, for 50 randomly chosen instances from the αNLI task. They obtain i) *generated sentences for a next possible event* for the *correct* and *incorrect hypothesis*, as well as ii) the *sentence stating observation* O_2. We ask each annotator to rate the sentences according to four quality aspects as stated below.

Grammaticality: the sentence is i) grammatical, ii) not entirely grammatical but understandable, or iii) completely not understandable;

Redundancy: the sentence contains redundant or repeated information;

Contradiction: the sentence contains any pieces of information that are contradicting the given observation O_2 or not;

Relevance: the possible next event is i) relevant, ii) partially relevant, or iii) not relevant.

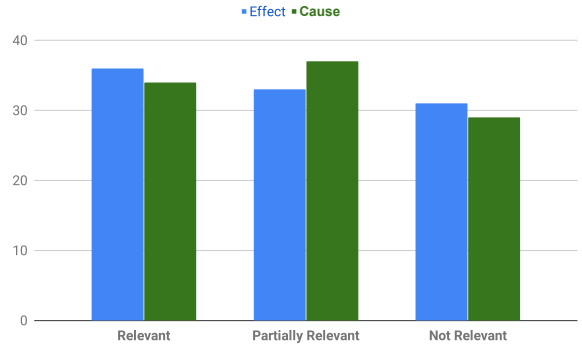


Fig. 5.7 Human evaluation of the *Relevance* of generated sentences for possible next events.

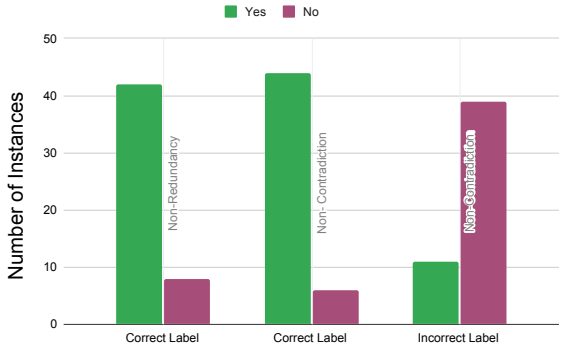


Fig. 5.8 Human evaluation of *Redundancy* and *Contradiction* of generations for possible next events.

For each aspect, they are asked to judge the sentence generated for the correct hypothesis[10]. Only for **Contradiction**, they are asked to judge both sentences, for correct and the incorrect hypotheses.

Results and Discussion. Figures 5.6, 5.8, and 5.7 present the results of manual evaluations of the generation quality, according to the different criteria described above.

For measuring inter-annotator agreement, we computed Krippendorff's α (Hayes and Krippendorff, 2007) for *Grammaticality* and *Relevance*, as it is suited for ordinal values, and Cohen's Kappa κ for *Redundancy* and *Contradiction*. We found α values are 0.587 and 0.462 for *Grammaticality* and *Relevance*, respectively (moderate agreement) and κ values 0.61 and 0.74 for *Redundancy* and *Contradiction* (substantial agreement). We aggregated the annotations from the three annotators using majority vote.

[10]The correct hypothesis was marked for the annotation.

Figure 5.6 shows that the majority of sentences (96%) are grammatical or understandable. Figure 5.8 shows that most sentences for correct labels are non-redundant (84%) and non-contradictory (88%), whereas for incorrect labels 39 instances are found to be contradictory with the observation O_2 (78%).

The manual evaluation supports our hypothesis that the generated sentences for correct labels should be more similar (less contradictory) compared to the sentences generated for incorrect labels. Figure 5.7 shows the ratio of sentences considered by humans as relevant, partially relevant, and irrelevant. The results show that 46% of cases are relevant (based on majority agreement) and 24% of cases are partially relevant. This yields that the generated sentences are (partially) relevant in most cases and thus should support abduction for both unsupervised ($LM_{\mathcal{I}}$ + BERTScore) and supervised ($LM_{\mathcal{I}}$ + $\mathcal{MTL}$) models.

5.6 Analysis

5.6.1 Case Study

Table 5.5 displays possible next events, generated by our $LM_{\mathcal{I}}$ model – along with the BERTscore measured between the possible next events $O_2^{H_j}$ and observation O_2. We see four different scenarios: (i) examples (a), (b) and (d) depicting the scenario where possible next events and observation pairs *correctly* achieve higher BERTscores [11], (ii) example (c) depicting the scenario where an incorrect possible next event and observation pair achieves higher BERTscores than the correct one, (iii) example (e) depicting the scenario where our $LM_{\mathcal{I}}$ model *incorrectly* generated similar next possible events, and (iv) example (f) depicting the scenario where both hypotheses (H_j) are equally plausible and hence the generated possible next events ($O_2^{H_j}$) are similar.

Intuitive reasons for these scenarios are, for example, for (a): there is a higher word overlap and semantic similarity between a correct next event and observation O_2, for example (b): there is higher semantic similarity; whereas for example (c): although there is a higher semantic dissimilarity, the word overlap between the wrong possible next event (*"She started to feel sick."*) and the observation (*"She felt much better afterwards."*) is much higher. Figure 5.9 depicts the bert similarity matrix, where we see that the due to higher token overlap (such as 'she', 'felt') between observation O_2 and possible next event ($O_1^{H_j}$) overall bert-score is higher.

[11] BERTscore matches words in candidate and reference sentences by cosine similarity.

	Observation (O)	**Given Hypotheses (H_j)**	**What *if* H_j happens? Generated possible next events $O_2^{H_j}$**	**MTL**	**Reasoning Type**	**BERT Score**	**Human Evaluation**
a.	O_1: Jim got ready for his first date.	H_1: **Jim's date wasn't attracted to him.**	$O_1^{H_2}$: He tried to get her number. She never responded to him.	$H_1, O_2^{H_1}$	*Reaction*	**47.34**	Non-Contradiction
	O_2: Since then, she has ignored all of Jim's text messages.	H_2: Jim went on the date and said he didn't like the girl.	$O_2^{H_2}$: He told her that he didn't want to date her.			44.19	Contradiction
b.	O_1: Nate had the summer off before college.	H_1: **Nate spent the summer traveling and partying.**	$O_2^{H_1}$: Nate had a lot of fun.	$H_1, O_2^{H_1}$	*Motivation*	**49.02**	Non-Contradiction
	O_2: Nate's last summer before college was a total blast!	H_2:Nate decided to spend the entire summer working in the Mines.	$O_2^{H_2}$: Nate spent the summer working in the mines.			44.34	Redundancy
c.	O_1: Dotty was being very grumpy.	H_1: Dotty ate something bad.	$O_2^{H_1}$: She started to feel sick.	$H_2, O_2^{H_2}$	*Emotional Effect*	**0.604**	Contradiction
	O_2: She felt much better afterwards	H_2: **Dotty call some close friends to chat.**	$O_2^{H_2}$: They all tried to make her happy.			0.405	Non-Contradiction
d.	O_1: Cay had a crush on a boy in her class.	H_1: **Cay sent a love note to the boy.**	$O_2^{H_1}$: The boy responded and said he liked Cay.	$H_1, O_2^{H_1}$	*Emotional Effect*	**0.509**	Non-Contradiction
	O_2: He smiled at her after and said he liked her too!	H_2: She told him she did not like him.	$O_2^{H_2}$: The boy was very sad about it.			0.423	Contradiction
e.	O_1: *Daniel wanted to buy a toy plane, but he didn't have any money.*	H_1: **He opened a lemonade stand.**	$O_2^{H_1}$: He sold lemonade for a lot of money.	$H_1, O_2^{H_1}$	*Motivation*	0.304	Non-Contradiction
	O_2: He bought his toy plane, and kept working so he could buy another!	H_2:Daniel stayed home and didn't want to buy a plane.	$O_2^{H_2}$: He decided he needed to get a part time job.		*if-then Effect*	**0.318**	Non-Contradiction
f.	O_1: *Ali's mom enrolled her in a karate class.*	H_1: **Ali did not want to take karate.**	$O_2^{H_1}$: She was afraid she would get hurt.	$H_2, O_2^{H_2}$	*if-then Effect*	0.324	Non-Contradiction
	O_2: Ali was so embarrassed she didn't tell any of her friends.	H_2: Ali did horribly in her last class.	$O_2^{H_2}$: Ali was so embarrassed. She tried to hide it from her parents.		*if-then Effect*	**0.584**	Non-Contradiction

Table 5.5 Examples of generated possible next events for solving αNLI using our $LM_{\mathcal{I}}$ model. Column 3: Hypothesis and possible next events chosen by our $LM_{\mathcal{I}} + \mathcal{MTL}$ model; Column 4: Reasoning type between the hypothesis H_j and O_2; Column 5: BERTScore between the $O_2^{H_j}$ and O_2; Column5: Human evaluation of the possible next events with respect the observation O_2.

5.6.2 Impact of Reasoning types.

Finally, to better assess the performance of our model, we determine what *types of reasoning* underly the abductive reasoning tasks in our data, and examine to what extent our models capture or not these reasoning types. We consider again the 50 instances that were annotated by our previous annotators and manually classify them into different reasoning types. We broadly divided the data into 6 categories: (i) Motivation, (ii) Spatial-Temporal, (iii) Emotional, (iv) Negation, (v) Reaction, (vi) Situational fact. The most frequent type was Emotional (10), most infrequent was Spatial (7). We ask a new annotator to annotate the reasoning types for these 50 instances. Considering the relevance and contradiction

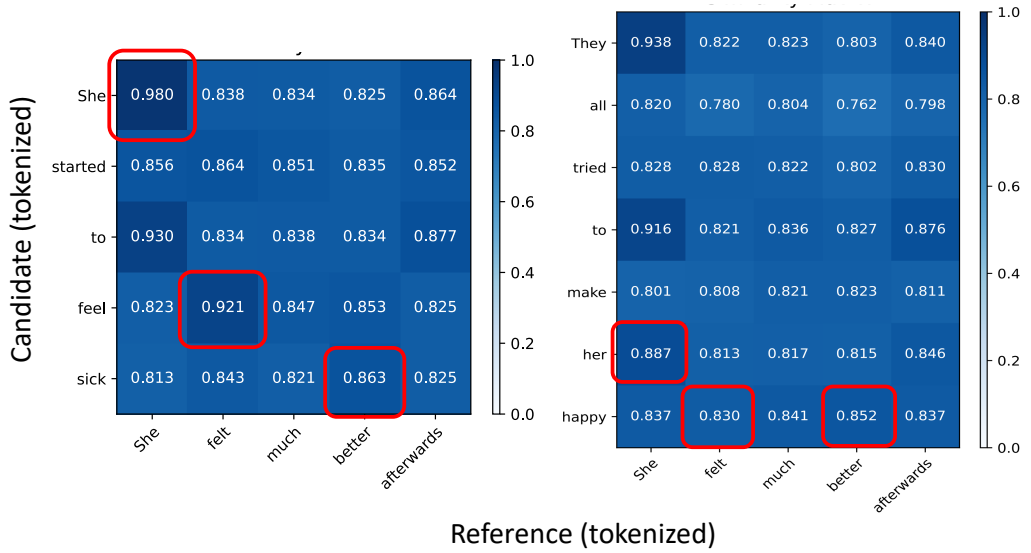


Fig. 5.9 We compute a pair-wise bert score similarity matrix (*without rescaling*) between the observation O_2 and $O_2^{H_j}$ to better understand the observed score in Table 5.5(c). The left hand side matrix is the scores for $O_2^{H_1}$ and the right one is for $O_2^{H_2}$. The red circles highlights some scores for important tokens and where the bert-score gave higher similarity scores.

categories from the previous annotations we determine that for Negation (8), Emotional (10), and Reaction (8) *all* generated events for *correct labels* are *partially or fully relevant and non-contradictory*. An intuitive reason can be that we train our $LM_{\mathcal{I}}$ model to learn how different counterfactual hypothetical events emerging from a single premise can lead to the same or different outcomes through a series of events. Some counterfactual events ($s_2^{'}$) are negations of the original event (s_2) in the TIMETRAVEL dataset. This may support the reasoning class Negation. For the other categories: Motivation, Spatial-temporal, and Situational fact, we detect errors regarding (missing) *Relevance* in 21%, 14% and 28% of cases, respectively. Table 5.6 illustrates an example from the class Situational Fact, where our generated next event is *irrelevant* and *redundant*.

5.7 Summary

In this Chapter, we have introduced a novel method for addressing the abductive reasoning task by explicitly learning what events could follow other events in a hypothetical scenario, and learning to generate such events, conditioned on a premise or hypothesis. We show how a language model – fine-tuned for this capability on a suitable narrative dataset – can be leveraged to support abductive reasoning in the αNLI tasks, in two settings: an unsu-

Observation (O)	**Given Hypotheses** (H_j)	**What *if*** H_j **happens?**
O_1: Jenna hit the weight hard in the gym.	H_1: Her neck pain stopped because of this.	$O_2^{H_1}$: She decided to take a break.
O_2: She took a cold bath in order to alleviate her pain.	H_2: **Jenna pulled a muscle lifting weights.**	$O_2^{H_2}$: Jenna lost weight in the gym.

Table 5.6 Error Analysis: An example of generated possible next event $O_2^{H_j}$ from Situational Fact category.

pervised setting in combination with *BERTScore*, to select the proper hypothesis, and a supervised setting in a multi-task learning setting. Our experiments show that our unsupervised $LM_{\mathcal{I}}$+*BERTScore* model outperforms some of the strong supervised baseline systems on αNLI. We also showed that $LM_{\mathcal{I}}$ + $\mathcal{MTL}$ yields better performance compared to both COMET + $\mathcal{MTL}$. One reason behind the improvement is that the temporal knowledge generated by our method is grounded to the context, whereas COMET generates inferential knowledge based on a single event without considering the context. Our research thus offers new perspectives for training generative models in different ways for various complex reasoning tasks.

Chapter 6

Social Commonsense Reasoning with Multi-head Knowledge Attention

"Inferences of science and commonsense differ from those of deductive logic and mathematics in a very important respect, namely, when the premises are true and the reasoning correct, the conclusion is only probable. "

– Bertrand Russell

In the previous two chapters, we focused on developing methods that can make inferences about the social dynamics of story characters (mental states prediction) and story events (temporal knowledge about events). In this chapter, we pursue the ambitious goal of social commonsense reasoning with NLP systems. In the first section, we give a brief motivation to the role of commonsense knowledge for SCR tasks and introduce a new task named counterfactual invariance prediction for NLP. The second section presents a new method to integrate such knowledge into SOTA transformer-based NLP models to improve their social reasoning capabilities. We evaluate our knowledge integration model on two social commonsense reasoning tasks: language-based abductive reasoning and counterfactual invariance prediction. Finally, we end the chapter with a human analysis of the model's robustness and knowledge incorporation capabilities. This chapter is based on work originally published in (Paul and Frank, 2020).

6.1 Social Commonsense Reasoning

Social Commonsense Reasoning is the ability to infer pragmatic implications that are beyond surface level understanding. For example, in Figure 6.1, we see an abductive reasoning task, where given two observations: *Dotty was being very grumpy* and *She felt much better afterwards* – select a plausible explanation about what could have provoked the change in Dotty's emotion. In order to judge the plausibility of such explanations, we need to infer knowledge about mental states of people and social implications. Such knowledge includes that *calling a close friend*, in general, makes *people feel happy*, *someone is being very grumpy*, wants *to feel better*. In our last two chapters we presented methods to automatically identify such knowledge from social events in a narrative context. In this chapter we consider such knowledge in a structured form to make our model more interpretable. In this chapter, instead of retrieving and selecting knowledge from a static KG (see Chapter 4), we aim to train a model to learn how to dynamically generate such knowledge. Additionally, we build on the hypothesis that models performing such reasoning tasks need to consider multiple knowledge rules jointly (see Fig. 6.1). Hence, we introduce a novel multi-head knowledge attention model which learns to focus on multiple pieces of knowledge at the same time, and is able to refine the input representation in a recursive manner (see sec.6.3).

In this chapter, we investigate social commonsense reasoning in narrative contexts. Specifically, we address two different reasoning tasks: language-based abductive reasoning[1], and counterfactual invariance prediction. We introduce the Counterfactual Invariance Prediction task (CIP), which tests the capability of models to predict whether under the assumption of a counterfactual event, a factual event remains invariant or not in a narrative context. Figure 6.1 illustrates an example: Given a narrative context – *"Dotty was being very grumpy" (premise),*

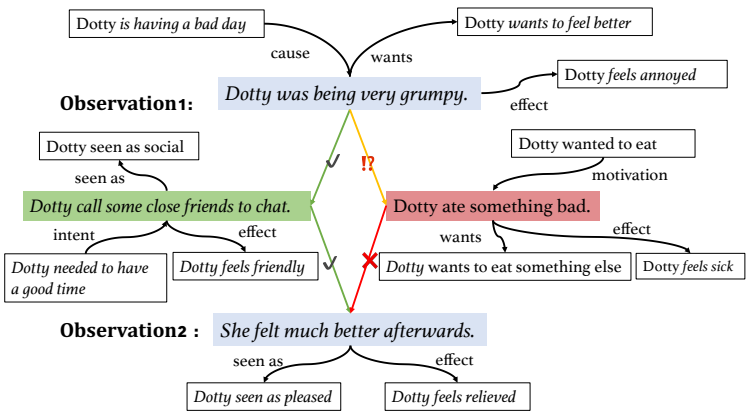


Fig. 6.1 Motivational example: The top and bottom blue boxes show two observations. The green and red box contain a plausible and an implausible hypothesis, respectively. A green line denotes that an event is likely to follow, the yellow line that an event is somewhat unlikely to follow, the red line something unlikely.

[1] More details about the Abductive reasoning task is found in Chapter 5

Context	**Answer**
s_1*: Bob had to get to work in the morning.* s_2*: His car battery was struggling to start the car.* s_3*: He called his neighbor for a jump start.* s_2'*: His car won't start.* s_3*: He called his neighbor for a jump start.*	**[Yes]** or [No]
s_1*: Bill and Teddy were at the bar together.* s_2*: Bill noticed a pretty girl.* s_3*: He went up to her to flirt.* s_2'*: Bill noticed his mom was there.* s_3*: He went up to her to flirt.*	[Yes] or **[No]**
s_1*: I loved to eat honey with my oatmeal.* s_2*: One day I unexpectedly ran out of honey.* s_3*: I did not want to eat my oatmeal without honey.* s_2'*: One day I realized that maple syrup was even better with my oatmeal.* s_3*: I did not want to eat my oatmeal without honey.*	[Yes] or **[No]**

Table 6.1 Examples from CIP task dataset used in this work. The correct choice in each example is given in bold text.

"Dotty called some close friends to chat" (hypothesis), "She felt much better afterwards."(conclusion) – will a counterfactual assumption (alternative hypothesis), e.g., *"Dotty ate something bad"*, still lead to same conclusion? Finally, we assume that a model learned about such counterfactual assumptions learns can help in predicting the best explanation in an abductive reasoning task.

6.1.1 Counterfactual Invariance Prediction Task

Counterfactual Reasoning (CR) is the mental ability to construct alternatives (i.e., counterfactual assumptions) to past events (actual world) and to reason about their (hypothetical) implications (Epstude and Roese, 2008; Roese and Morrison, 2009). In philosophy, there are three broad questions that counterfactuals raise (Starr, 2021):

1. How do we communicate and reason about alternate possibilities which are different from the way things actually are?

2. How can our experience in the actual world justify our thought and how we talk about distant alternative possibilities? (Menzel, 2021a; Mallozzi et al., 2021)

3. Do these distant alternative possibilities exist independently from the actual world, or are they grounded in things that actually exist? (Menzel, 2021b)

In this thesis, we are primarily interested in (*first question*) making NLP systems reason about alternate possibilities. In social psychology, counterfactual thinking is linked to

concepts like *free will, sense of self, wishful thinking,* etc (Alquist et al., 2015). For an in-depth introduction to counterfactual thinking and how it is related to mental states, we refer the reader to (Starr, 2021). Recently, in NLP, Qin et al. (2019) introduced a counterfactual story generation task, where given an intervention (alternate possibility) the task is to complete the narrative grounding it to the actual world. One of the key challenges of CR is judging *causal invariance*, i.e., deciding whether a given factual event is invariant under counterfactual assumptions, or whether it is not (Peters et al., 2016; Qin et al., 2019).

In the field of cognitive science and artificial intelligence, counterfactual reasoning plays a crucial role to explain how a particular states of mind lead (*forward looking*) to certain choices and actions (Chater et al., 2010). Causal invariance is an understudied problem for intuitive inference tasks. Hence, we combine these two topics and define a new Counterfactual Invariance Prediction (CIP) task that tests the capability of models to predict whether under the assumption of a counterfactual event, a (later) factual event remains invariant or not in a narrative context (cf. Table 6.1). This task requires deeper understanding of causal narrative chains and reasoning in forward direction. Qin et al. (2019) proposed a dataset to encourage models to learn to rewrite stories with counterfactual reasoning. We automatically collect counterfactual invariance examples along with non-invariant examples from (Qin et al., 2019) to create a balanced dataset for our proposed CIP task.

The formal setup is: given the first three consecutive sentences from a narrative story s_1 (premise), s_2 (initial context), s_3 (factual event) and an additional sentence s_2' that is counterfactual to the initial context s_2, the task is to predict whether s_3 is invariant given s_1, s_2' or not. Hence, we impose a constraint that for counterfactual examples the (original) s_3 should be same as the (edited) s_3' and for non-invariant examples the (original) s_3 != (edited) s_3'. The train/dev/test data (cf. Table 6.3) are balanced with an equal number of *Yes/No* answers, hence the random baseline is 50%. To compute human performance, we gave 100 instances from the test set to expert evaluators. Human accuracy on the CIP task is at *84.8%*. We aim to study the role of mental states and pragmatic inference (Social Commonsense Knowledge (SCK)) for detecting counterfactual invariant social events.

6.2 Extracting Semantic & Social Commonsense Knowledge

This section details the steps we follow to generate social commonsense knowledge about events mentioned in a narrative. See Figure 6.2 for illustration. Understanding a narrative text requires the ability to identify events and to reason about their causal effects. Beyond

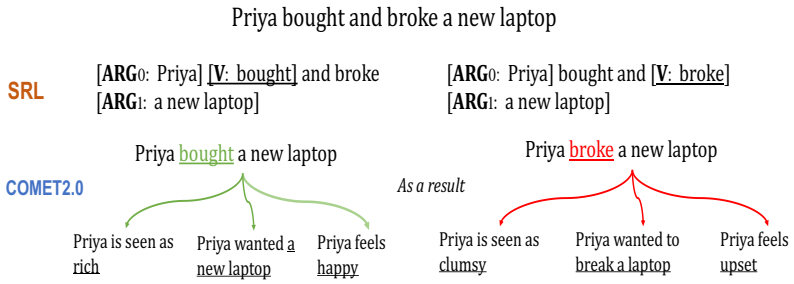


Fig. 6.2 Depicting the steps to extract commonsense knowledge about social events.

causal relations, they require the understanding of narrative relations, as in narrative chains or schemata (Chambers and Jurafsky, 2008b). This is knowledge about characteristic script-like event sequences where semantic roles of consecutive events are referentially bound to roles of preceding events. While Chambers and Jurafsky (2008b) focused on the induction of schemata using corpus statistics, we will combine detected events with deeper commonsense knowledge.

In a first step we apply SRL to extract the basic structure *"who did what to whom, when and where"* from each sentence in the context, using state-of-the-art SRL (Shi and Lin, 2019). In a second step, we use commonsense transformer (COMET2.0,[2] Bosselut et al. (2019)) to extract social commonsense knowledge about the extracted events. COMET2.0 is trained on the ATOMIC (Sap et al., 2019a) inferential knowledge resource which consists of 877K everyday events, each characterized by nine relation types (*xIntent, xNeed, xReact*, etc.) which we call *dimensions*. These dimensions connect the event in question with manifold properties, emotions, as well as other states or events.

In the last processing step we generate, for each event in each sentence from our datasets, all dimensions defined for it using COMET2.0. For example, for: *Dotty ate something bad* we generate the tuple: ⟨*PersonX, xReact, sick*⟩[3] and derive ⟨*Dotty, feels, sick*⟩ by substituting *PersonX* with the logical subject, the filler of the role *ARG0*.

6.3 Multi-Head Knowledge Attention (MHKA) Model

In this section we introduce the MHKA model and discuss some key differences in how MHKA works for the two different Social Commonsense Reasoning tasks. For a model overview see Figure 6.3.

[2]COMET2.0 uses GPT-2 as pretrained model.
[3]where the structure format is: ⟨*event, relation, object*⟩

Task	Input Format	Output
α**NLI**	[CLS] O_1 H_i [SEP] O_2 [SEP]	H_1 or H_2
CIP	[CLS] s_1 s_2 s_3 [SEP] s_1 s_2' s_3 [SEP]	YES or NO

Table 6.2 Different input and output formats: [CLS] is a special token used for classification, [SEP] a delimiter.

6.3.1 Model Architecture

MHKA consists of 3 modules: (a) the *Context Encoding Layer* consists of a pre-trained LM, (b) the *Knowledge Encoding Layer* consists of stacked transformer blocks, (c) the *Reasoning Cell* consists of transformer blocks with *multi-head attention* that allows the model to jointly attend to the input representation and the encoded knowledge. The input format for each task is depicted in Table 6.2.

(a) Context Encoding Layer: For each task, we concatenate the inputs as a sequence of tokens x_n = (x_{n_1}, .. x_{n_m}), and compute contextualized representations with a pre-trained LM. We obtain n different representations for n input options i.e., $h_{x_n} = encode(x_n) = (h_{n_1}, .., h_{n_m})$, where for αNLI n=2 and for CIP n=1. As pre-trained LMs we consider (i) BERT (Devlin et al., 2019) and (ii) RoBERTa (Liu et al., 2019).

(b) Knowledge Encoding Layer: As depicted in Figure 6.3, the knowledge encoding layer is a Transformer-Block (Liu et al., 2018; Alt et al., 2019) as typically used in the decoder part of the transformer model of Vaswani et al. (2017).

The core idea is that the model repeatedly encodes the given knowledge input over multiple layers (i.e., Transformer blocks), where each layer consists of masked multi-head self-attention followed by layer normalization and a feed-forward operation. Similar to the context input format, we concatenate the knowledge inputs as a sequence of tokens k_n = ($k_{n_1}, .., k_{n_w}$), where k_n is the knowledge used for input option x_n. In order to obtain the hidden knowledge representation we do the following:

$$\begin{aligned} h_{k_n^0} &= k_n W_{ke} + W_{kp}, \\ h_{k_n^l} &= tb(h_{k_n^{l-1}}), \forall l \in [1, L] \end{aligned} \tag{6.1}$$

where W_{ke} is the token embedding matrix, W_{kp} the position embedding matrix, tb the transformer block, and L the number of transformer blocks.

(c) Reasoning Cell: The main intuition behind the reasoning cell is that given the context representation, the model should learn to emulate reasoning over the input using the knowledge representation obtained from the knowledge encoder. The Reasoning Cell

is another transformer block, where the model repeatedly performs multi-head attention over the context and knowledge representations, and thus can iteratively refine the context representation. This capability is crucial for allowing the model to emulate complex reasoning steps through composition of various knowledge pieces.

The multi-head attention function has three inputs: a query Q (context representation), key K and value V (both knowledge representation). It relies on scaled dot-product attention

$$\begin{aligned} Q &= h_{x_n} + W_{xp} \\ a_{xk_n} &= softmax(\frac{QK^T}{\sqrt{d_z}})V \end{aligned} \tag{6.2}$$

where $K = V = h_{k_n}$, d_z the dimensionality of the input vectors representing the key and value, and W_{xp} is the position embedding. We project the output representations from the reasoning cell into logit (s) of size n (the number of output values) using a linear classifier. Finally, we compute the scores $y = max(s_i)$ where, $i = 1, .., n$. For CIP, where $n = 1$, we treat a logit score > 0 as predicting yes, otherwise the answer is no.

6.3.2 MHKA model for Social Commonsense Reasoning Tasks

There are some key differences in how MHKA solves the two reasoning tasks:

(a) In the **abductive αNLI reasoning task**, the model must predict – given incomplete observations O_1 and O_2 – which of two hypotheses H_i is more plausible. For example: O_1: *Daniel wanted to buy a toy plane, but he didn't have any money*; O_2: *He bought his toy plane, and kept working so he could buy another*; correct H_i: *He opened a lemonade stand*. Here, the model needs to link O_2 back to O_1 using social inference knowledge relating to the H_i that best supports one of the sequences: O_1, H_i, O_2.

In this case, the model obtains the (encoded) input: O_1, H_i, O_2, and is tasked to predict the correct H_i, using available knowledge rules. [4]

(b) For **Counterfactual Invariance Prediction, CIP**, the model needs to decide whether for given a context C_{s_1,s_2,s_3}, under the assumption of a counterfactual s_2', the given s_3 remains invariant or not. I.e., given: *Dotty was grumpy. Dotty called close friends to chat. She felt better afterwards.* and the counterfactual s_2': *Dotty ate something bad* – can it still be true that *Dolly felt better afterwards*? Here our model gets as input the factual (s_2) and a counterfactual (s_2') context: s_1, s_2, s_3 [SEP], s_1, s_2', s_3 (cf. Table 6.2) and is tasked to predict

[4]Relevant knowledge from COMET2.0 here includes: [O_1: Daniel wanted to have money] $\rightarrow$ [H_i: Daniel wanted to make money, **Daniel then makes money**] $\rightarrow$ [O_2: Daniel needed to have money]. Clearly, H_i is supported by H_1: *He opened a lemonade stand*. So we can judge that the selected knowledge (partially) supports H_1.

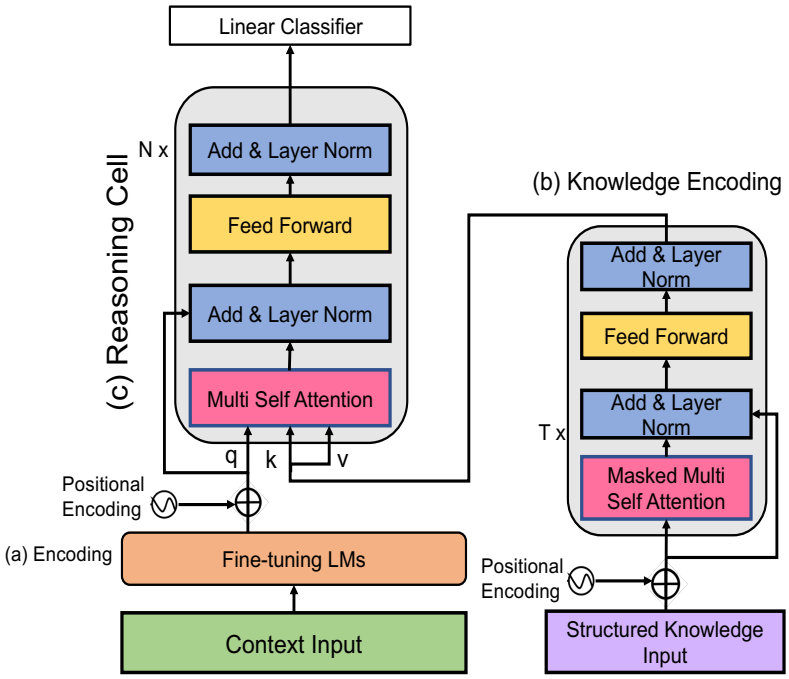


Fig. 6.3 Overview of our Multi-Headed Knowledge Attention Model. It consist of three components (a) the *Context Encoding Layer* (b) the *Knowledge Encoding Layer*, and (c) the *Reasoning Cell*.

whether or not s_3 remains true under the assumption s_2'. Again, the model needs to identify relevant knowledge to substantiate whether s_3 prevails given s_1 and s_2'.

Abduction *meets* Counterfactual Reasoning Clearly, when learning to judge whether s_3 holds true given both a factual (s_1, s_2) and counterfactual (s_1, s_2') context, the CIP model learns how different events can or cannot lead to the very same factual event in a hypothetical reasoning task. Our intuition is that such a model effectively also acquires knowledge about what kinds of events can provide *evidence* for a given event, as is needed to perform *abduction*.

Hence, we hypothesize that a model that has learned to understand and reason about counterfactual situations can also support abductive reasoning (i.e., finding the best explanation for an event). In our experiments, we test this hypothesis, and evaluate the performance of a model on the αNLI task, that we first train on CIP and then finetune it on the abductive inference task.

Task	Train	Dev	Test
αNLI	169654	1532	3059
CIP	12700	1008	1184

Table 6.3 Dataset Statistics: nb. of instances.

6.4 Experiments

Tasks and Settings. We apply our model to the two social reasoning tasks (*abductive reasoning, CIP*) introduced in §5.1 and §6.1.1. We train models for each task using the input settings stated in Table 6.2. Data statistics is given in Table 6.3. We extract, for each event in each input sentence, social commonsense reasoning knowledge from COMET2.0, as detailed in §6.2. For the extraction process we use SRL as implemented in AllenNLP Gardner et al. (2018).

Hyperparameter Details. In all models the Reasoning Cell and the Knowledge Encoder are both instantiated by a Transformer with 4 attention heads and depth=4. For each task, we select the hyperparameters that yield best performance on the dev set. Specifically, we perform a grid search over the hyperparameter settings with a learning rate in $\{1e\text{-}5, 2e\text{-}5, 5e\text{-}6\}$, a batch size in $\{4, 8\}$, and a number of epochs in $\{3, 5, 10\}$. Training is performed using cross-entropy loss. For evaluation, we measure accuracy. We report performance on the test sets by averaging results along with the variance obtained for 5 different seeds.

Baselines. We compare our model to the following baselines:
(a) *OpenAI-GPT* (Radford et al., 2018) is a multi-layer Transformer-Decoder based language model, trained with an objective to predict the next word.
(b) *Transformer Encoder* Model has the same architecture[5] as OpenAI-GPT without pre-training on large amounts of text.
(c) *BERT* (Devlin et al., 2019) is a LM trained with a masked-language modeling (MLM) and next sentence prediction objective, i.e., it is trained to predict words that are masked from the input.
(d) *RoBERTa* (Liu et al., 2019) has the same architecture as BERT, yet without next-sentence prediction objective. *RoBERTa-B(ase)* and *-L(arge)* were trained on more data and optimized carefully.
(e) *McQueen* (Mitra et al., 2019) proposed ways to infuse unstructured knowledge into

[5] 12-layer, 768-hidden, 12-heads

Model	Dev (%)	Test (%)
Majority $^{\diamond}$	50.8	–
GPT $^{\diamond}$	62.7	62.3
BERT -L $^{\diamond}$	69.1	68.9
McQueen (Mitra et al., 2019)	86.68	84.18
Concurrent Work		
$L2R^2$ (Zhu et al., 2020)	–	86.81
COMET + $\mathcal{MTL}$	85.4±0.4	84.6±0.7
$LM_{\mathcal{I}} + \mathcal{MTL}$	86.2±0.5	85.5±0.6
This work		
Transformer Enc. w/o LM−Pretraining	52.12	51.25
+ MHKA	54.96	53.91
RoBERTa−B	71.2±0.3	71.13±0.5
RoBERTa−B + MHKA	73.87±0.2	**74.17**±0.2
RoBERTa−L	85.06±0.7	84.48±0.7
RoBERTa−L + Joint Training	85.58±0.5	84.91±0.7
RoBERTa−L + MHKA	87.44±0.5	**87.12**±0.5
Human Perf.	–	91.4

Table 6.4 Results on αNLI dataset, $^{\diamond}$: as in Bhagavatula et al. (2020), L = Large, B = Base, excluding unpublished leaderboard submissions

pretrained language model (RoBERTa) to address the αNLI task. Mitra et al. (2019) used the original *ROCStories* Corpus (Mostafazadeh et al., 2016) and the Story Cloze Test that were used in creating the αNLI dataset.
(f) $L2R^2$ (Learning to Rank for Reasoning) (Zhu et al., 2020) proposed to reformulate the αNLI task as a ranking problem. They use a learning-to-rank framework that contains a scoring function and a loss function.
(g) Finally, we also compare our current model with our previous methods : COMET + $\mathcal{MTL}$ and $LM_{\mathcal{I}} + \mathcal{MTL}$ (see in Chapter 5). Please note, here we use RoBERTa as a base-model.

6.5 Experimental Results

This section describes the experiments and results of our proposed model in different configurations.

Results on αNLI. Our experiment results for the **αNLI task** are summarized in Table 6.4. We compare performances of the following models: majority baseline, pre-trained

LM baselines, and MHKA fine-tuned on RoBERTa-B(ase)/-L(arge). We observe consistent improvements of our MHKA method over RoBERTa-B ($+3.04$ percentage points, pp.) and RoBERTa-L ($+2.64$ pp.) on αNLI. Since MHKA uses RoBERTa to encode the input, this gain is mainly attributed to the use of knowledge and the multi-head knowledge attention technique. Interestingly, when we compare MHKA with COMET+$\mathcal{MTL}$ we see that there is a gain of $+2.52$ pp., which suggests that explicitly incorporating knowledge helps more than implicitly learning about knowledge. To better understand the impact of knowledge from pre-trained LMs, we trained a transformer encoder model *without* fine-tuning on a pretrained LM (see Table 6.4). Clearly, the overall performance of such a model drops considerably compared to the SOTA supervised models, but the improvement of MHKA by $+2.84$ points suggest that the impact of knowledge and reasoning obtained through multi-head knowledge attention is stable and independent from the power of LMs. Further, we compare our knowledge incorporation technique with *Joint Training*: this method uses pre-trained LMs to jointly encode both task-specific input and the knowledge ([CLS] (K)nowledge [SEP] (I)nput text). More details about the *Joint Training* model are given in §3.2. Table 6.4 shows that *Joint Training* yields limited improvement (+0.43 pp.) over the RoBERTa-L baseline – the intuitive reason being that the pretrained LMs were never trained on such structured knowledge.[6] However, our *MHKA model* shows a solid improvement of 2.64 pp. over the baseline. This suggests the impact of the *Multi-Head Knowledge Attention* integration technique.

Low Resource Setting for αNLI. To better understand the impact of dataset scale on the performance of MHKA, and to test its robustness to data sparsity on αNLI, we investigate low-resource scenarios where we only use $\{1, 2, 5, 10, 100\}\%$ of the training data. Figure 6.4 shows constant advances of MHKA over both RoBERTa-Base and -Large. This result indicates the importance of knowledge in low-resource settings.

Results on CIP. Table 6.5 reports the results of our MHKA model on the CIP task, comparing to both RoBERTa baselines. As this is a new task, we also report the results of RoBERTa-Base with different input formats. We find that providing the model with the full sequence (s_1, s_2, s_3 [SEP] s_1, s_2' , s_3) gives best performance. By extending RoBERTa-Base and -Large with our MHKA reasoning component, we obtain an improvement of $+1.7$ and $+1.1$ percentage points, respectively.

CIP for Transfer Learning. We now test our hypothesis, discussed in §6.3.2, that a model trained on the CIP task can support the αNLI task. We first fine-tune two models: RoBERTa-L and the RoBERTa-L+MHKA model on the CIP task (using the hyperparameters

[6]They also have a disadvantage when the length of context + knowledge increases, as this causes a bottleneck for computation on a GPU with limited memory (8-24GB).

Model	Input format	Dev%	Test%
RoBERTa-B	s_1, [SEP], s_2', [SEP], s_3	63.29	61.8
	s_1, s_2 [SEP] s_1, s_2'	57.44	58.9
	s_1, s_2, s_3 [SEP] s_1, s_2'	64.38	62.8
	s_1, s_2, s_3 [SEP] s_1, s_2' , s_3	66.66	67.98±0.5
+ MHKA	s_1, s_2, s_3 [SEP] s_1, s_2' , s_3	69.34	69.7±0.6
RoBERTa-L	s_1, s_2, s_3 [SEP] s_1, s_2' , s_3	72.4	71.95±0.6
+ MHKA	s_1, s_2, s_3 [SEP] s_1, s_2' , s_3	74.4	73.05±0.3
Human Perf.			**84.8**

Table 6.5 Results on Counterfactual Invariance Prediction (CIP).

Model	Dev	Test
RoBERTa-Large-αNLI	76.3	76.8
Transfer Learning	78.00	**79.04**
Transfer Learning + MHKA	78.6	**80.77**

Table 6.6 Impact of Counterfactual Invariance Prediction on αNLI. Training data size for αNLI is 8.5k (5%)

for the CIP task, Table 6.5). As a transfer-learning method, we fine-tune these models on 5% of the training data for the αNLI task (using the hyperparameters for αNLI, Table 6.4) and report the results in Table 6.6 as “Transfer Learning" and “Transfer Learning + MHKA". Table 6.6 also reports the results for RoBERTa-L trained on 5% of the data of αNLI (called RoBERTa-L-αNLI).[7] We obtain a +2.84 pp. improvement over this baseline by applying the pre-trained CIP model on the αNLI task, and observe a further +1.73 pp. improvement (i.e., overall 3.97 points wrt. the baseline) with the stronger MHKA model. These results confirm our hypothesis, and show that learning to distinguish the outcomes of factual and counterfactual events can help the model to better perform abduction.

Ablation on Reasoning Cell. To give further insight into the factors for the model's capacity, we study the impact of the number of heads and layers in the reasoning cell. The left part of Figure 6.5(a) shows the performance of the MHKA model with different numbers of heads and layers. Note that the hidden dimensions of RoBERTa-Large is 1024 which is not divisible by 3, therefore we have 1, 2, and 4 as our attention heads. We observe that increasing the number of heads and layers improves the performance of the model. The intuitive explanation is that *multiple heads* help the model to focus on multiple knowledge

[7]The training data size of αNLI is 14x larger than CIP. Therefore, in order to study the impact of CIP on αNLI, we made the training data size of CIP and αNLI comparable.

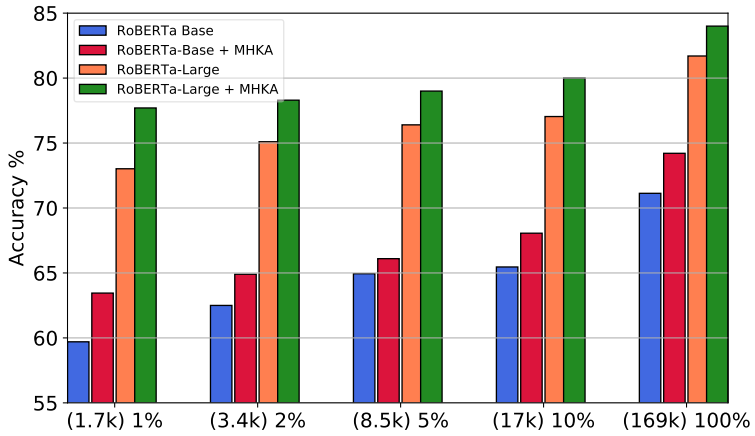


Fig. 6.4 Accuracy for αNLI (Low Resource Setting)

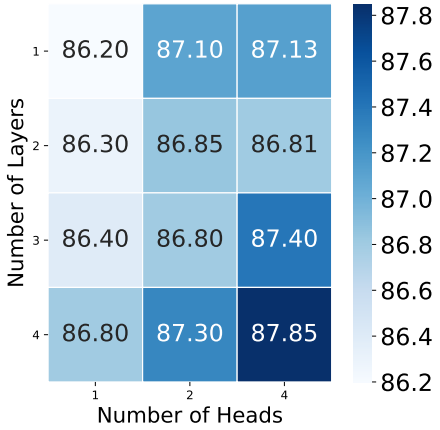


Fig. 6.5 (a) Performance of MHKA model with different numbers of Heads and numbers of Layers.

rules and at the same time *multiple layers* help the model to recursively select the relevant knowledge rules.

6.6 Analysis

Up to now, we have focused on performance analysis with different experimental settings and model ablations to analyze our model's capacities. Now, we turn to leveraging the fact that our model works with semi-structured knowledge in order to obtain deeper insight into its inner workings.

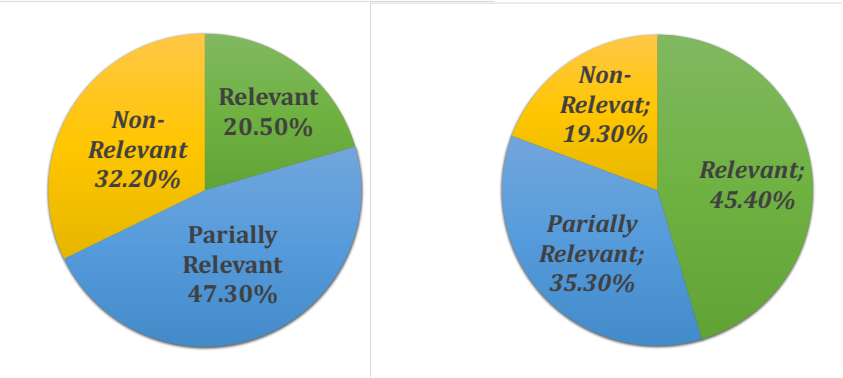


Fig. 6.6 Human evaluation of the relevance of Knowledge Rules a) for 100 instances from the αNLI dev set and b) for the 56 (out of the 100) instances where the MHKA model predicted the correct hypothesis.

	all know-ledge	**w/o irrelevant**	**w/o relevant + partially relevant**	**replacing relevant**
acc	56.2	57.6 (+1.4)	49.4 (−6.8)	45.05 (−11.2)
#	56	54 (−2)	20 (−36)	18 (−38)

Table 6.7 *row 1*: accuracy on 100 random instances from αNLI devset where the RoBERTa-L baseline fails; *row 2*: nb. of instances (#) correctly predicted by MHKA.

6.6.1 Quantitative Analysis.

Analysis on Knowledge Relevance. We conduct human evaluation to validate the effectiveness and relevance of the extracted social commonsense knowledge rules. We randomly select 100 instances from the αNLI dev set for which the RoBERTa-Large Baseline had failed, along with their gold labels and the extracted knowledge. Table 6.7 shows that MHKA correctly predicts 56 instances correctly. We asked two annotators to mark the knowledge rules that are relevant or partially relevant or irrelevant for each all 100 instances. The obtained answers yield that in 20.50% of cases the knowledge rules were relevant, in 47.30% of cases they were partially relevant (see Figure 6.6.a). Figure 6.6.b depicts the relevance of knowledge rules for instances that are *correctly predicted* by MHKA. The inter-annotator agreement had a Fleiss' κ=0.62.

Analysis of Model's Robustness. We then test the robustness of the models' performance by manipulating the knowledge it receives for these instances in different ways: (a) we remove *irrelevant* and (b) *relevant* knowledge rules, (c) we manually change randomly selected rules from those that were found to be relevant by our annotators, and perturb them with artifacts. E.g., where annotators found that "PersonX's feelings" is relevant, we change

All knowledge	Removing relevant relation tuples	Removing relation tuples randomly
87.85	85.4 (−2.45)	86.9 (−0.95)

Table 6.8 Accuracy on αNLI (dev set)

the sentiment by choosing incorrect possible values from ATOMIC; for other relation types, we replace COMET's generated object with an antonym "PersonX wanted to be [nice → mean]", etc. We evaluate the effect of the perturbations i) on all 100 instances, and ii) on the 56 correctly predicted instances. Results are shown in Table 6.7.

We see, for (a), a small improvement over the model results when using all knowledge, whereas for (b) and (c) an important performance drop occurs. For the 56 instances that MHKA resolves correctly, for (b) and (c) we find the same effect, but with a much more drastic drop in performance for (b) and (c). This suggests that when the model is provided with relevant knowledge rules, it is able to utilize the knowledge well to perform the inference task.

In another test, we remove knowledge rules with relations which were found most relevant by our annotators (namely, *'PersonX's intent', 'PersonX's want', 'PersonX's need', 'effect on PersonX', 'effect on other', 'PersonX feels'*) (see *Supplement* for details). Table 6.8 reports the results on dev set.

We observe: (a) when we remove the *relevant* relational knowledge rules, the accuracy drops by 2.4 pp. suggesting that the model is benefitting from the knowledge rules. (b) when we remove knowledge rules *randomly*, the accuracy drop is minimal which shows the robustness of our model.

6.6.2 Qualitative Analysis.

Finally, we perform a study to better understand which knowledge rules were *"used or incorporated in the Reasoning Cell*" during the inference.

A case study. Figure 6.7 depicts an example from the *αNLI* task where we see the context at the top, and knowledge rules along with different scores below. The *Human scores* are annotated by the annotators where, 1.0 = Relevant, 0.50 = Partially relevant, 0.0 = Irrelevant. We also show the normalized attention scores over the structured knowledge rules[8].

We also measure a similarity score (using dot product) between the final representation of the Reasoning cell and different knowledge rules. Intuitively, we expect that relevant

[8]Note that we do not consider the attention maps as explanations. We assume that attention exhibits an intuitive interpretation of the model's inner workings.

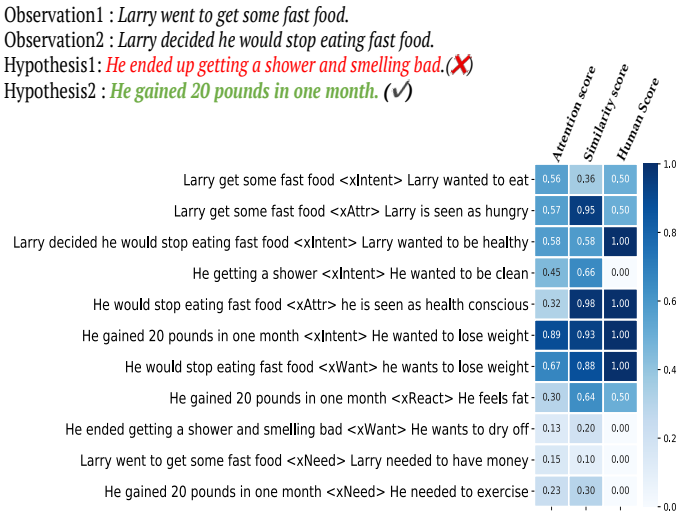


Fig. 6.7 Comparing relevance scores of knowledge.

knowledge rules should be incorporated in the final representation of the Reasoning cell, and therefore, should have a higher similarity score compared to irrelevant knowledge rules. Figure 6.7, illustrates one such example where we see that some relevant knowledge (judged by annotators) – *"He gained 20 pounds in one month ⟨xIntent⟩ He wanted to lose weight"*, and *"He would stop eating fast food ⟨xWant⟩ he wants to lose weight"* – are highly attended, and scored higher in similarity measure compared to others, indicating that the Reasoning Cell incorporated these knowledge rules.

To study this further, we randomly selected 10 instances from the αNLI dev set along with the knowledge rules. We found for 7 out of 10 instances that the MHKA model gave higher similarity scores to relevant or partially relevant knowledge rules than to irrelevant ones.

6.7 Summary

In this chapter, we present a new *multi-head knowledge attention model* to incorporate semi-structured social commonsense knowledge. We show that our model improves over state-of-the-art LMs on two complex commonsense inference tasks. Besides the improvement i) we demonstrate a correlation between abduction and counterfactual reasoning in a narrative context, based on the newly proposed task of counterfactual invariance prediction, which we apply to support abductive inference. Importantly, ii) we confirm the reasoning capacity of our model by perturbing and adding noise to the knowledge, and performing model inspection using manually validated knowledge rules. In future work, we aim to deeper investigate

compositional effects of inferencing, such as the interaction of socially grounded and general inferential knowledge.

Chapter 7

Generate Contextualized Inference Rules for Narrative Story Completion

"The subconscious mind is more susceptible to influence by impulses of thought mixed with 'feeling' or 'emotions', than by those originating solely in the reasoning portion of the mind."

– Napoleon Hill

In the previous chapters, we addressed social commonsense reasoning challenges as deterministic tasks, i.e., NLP systems are required to predict the correct option from a set of choices based on a given context. In this chapter, we delve further into how current NLP systems perform SCR as a NLG task. We study the role of contextualized commonsense knowledge in generating coherent narratives. We design a story generation task that requires NLG models to perform both forward and backward reasoning. We design a framework that aims to jointly learn both generating inference rules and generating narrative stories to address this challenge. Finally, we end the chapter with automatic and manual evaluations of the model's story sentence generation capabilities, especially in terms of coherence. This chapter is based on work originally published in (Paul and Frank, 2021a).

7.1 Introduction

Narrative story understanding, and similarly story generation, requires the ability to construe meaning that is not explicitly stated through commonsense reasoning over events in the story (Rashkin et al., 2018b). Early work in modelling narrative stories focused on *script learning* by defining stereotypical event sequences together with their participants (Schank

and Abelson, 1977). In later works, Chambers and Jurafsky (2008b, 2009); Balasubramanian et al. (2013); Nguyen et al. (2015); Pichotta and Mooney (2014) proposed methods to learn *narrative event chains* using a more straightforward event representation that allows for efficient learning and inference. Chambers and Jurafsky (2009) acquired Narrative Event Schemata from corpora and established the Narrative Cloze Task (Chambers and Jurafsky, 2008b) that evaluates script knowledge by predicting a missing event (verb and its arguments) in a sequence of observed events.

Most recent works have focused on different aspects of story generation such as (a) enhancing the coherence of generated stories (Fan et al., 2018), (b) methods to incorporate commonsense knowledge in NLG models (Guan et al., 2020; Ji et al., 2020), (c) generating stories with controllable styles (Peng et al., 2018; Brahman and Chaturvedi, 2020). We hypothesize that incorporating commonsense knowledge into NLG models can enhance the capability of generating coherent stories. Due to the remarkable improvement in performance, there is a shift in story-generating modelling from using sequence to sequence models (Pichotta and Mooney, 2016; Li et al., 2018; Fan et al., 2018) to transformer-language-model-based text generation model (Xu et al., 2020; Guan et al., 2020). While these pretrained LMs learn probabilistic associations between words and sentences, they still have difficulties in modelling causality (Mostafazadeh et al., 2020b). Also, in narrative story generation, models need to be consistent with everyday commonsense norms. Hence, to address a story generation task, (i) models need to be equipped with suitable knowledge, (ii) they need effective knowledge integration and reasoning methods, and ideally (iii) we want to be able to make the effectiveness of these methods transparent.

This chapter focuses on the aspects (i) to (iii), by investigating new methods that build on pretrained LMs to generate missing sentences from an incomplete narrative story. Specifically, we focus on *Narrative Story Completion (NSC)*, a new task setting for story generation. Given an incomplete story, specified only through its beginning and ending, the task is to generate the missing sentences to complete the story (see Figure 7.1). We hypothesize that in order to obtaining a consistent and coherent narrative story, the task requires a model's ability to perform commonsense inference about events and entities in a story. Unlike other existing tasks, NSC requires: *i)* generating *multiple sentences* to complete a story, and *ii) ensuring that* the generated sentences are *coherent* with respect to both *beginning and ending* of the story. Hence, the NSC task offers a challenging setup for investigating the reasoning capacities of a story generation model. In the next section, we describe our task setup in detail.

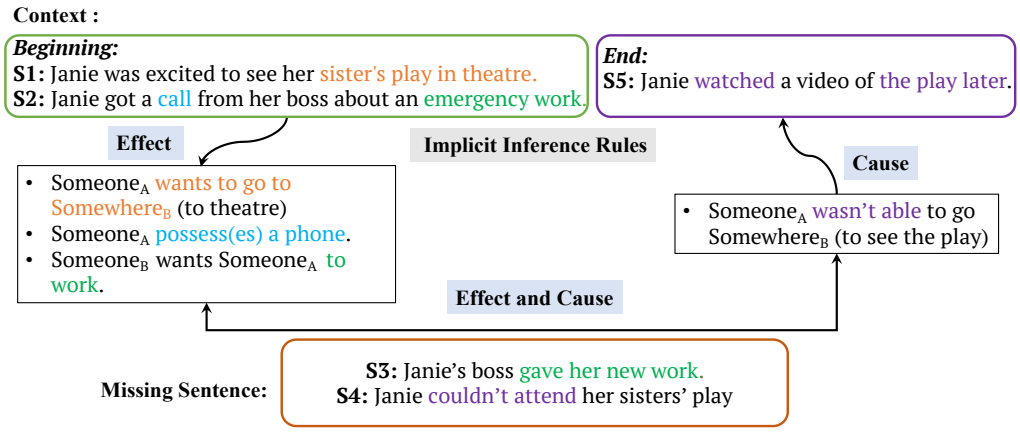


Fig. 7.1 An example of the *Narrative Story Completion Task*. Top and bottom boxes show the context (top) and missing sentences (bottom). The chain of implicit inference rules explains the connection between beginning and end, and allows to infer the missing sentences.

7.2 Narrative Story Completion

We formulate the *Narrative Story Completion task (NSC)* as follows: given an incomplete story (S= s_1, s_2, s_n) as a sequence of tokens $t = \{t_1, t_2, ..., t_{SEP}, ..., t_m\}$ (with t_{SEP} a mask token delimiting s_2 and s_n), the goal is to generate the missing sentences ($s_3, ..., s_{n-1}$) as a sequence of tokens y^{s_i}=$\{y_1^{s_i}, y_2^{s_i}, ..., y_v^{s_i}\}$ (with i = 3, ..., $n-1$ and v the maximum length of each sentence). In the setting of the NSC task, we expect the completed story to be coherent. That is, the generated sentences should exhibit reasonable logical connections, causal relationships, and temporal dependencies with each other and the given beginning and ending of the story. We define a discourse to be coherent if successive sentences that are about the same entities, and the reported events involving them can be construed to reflect common knowledge about how events are typically connected in a temporal sequence or by causal relations. Similar to Hobbs (1985), the criteria to conclude that discourse is coherent include the requirement that there are reflections of causality in the text.

Humans excel in drawing inferences and constructing causal chains that explain the connection between events (Kintsch and Dijk, 1978). Figure 7.1 illustrates this with an example from our *NSC* task.[1] From *Janie was excited to see her sister's play in theatre*$_{(s_1)}$. *Janie got a call from her boss about new work*$_{(s_2)}$ and the outcome *Janie watched a video of the play later.*$_{(s_5)}$ – we can construct inference rules in forward and backward direction: forward via EFFECT: Someone$_B$ (*boss*) gave work to Someone$_A$ (*Janie*); backward via CAUSE: Someone$_A$ (*Janie*) wasn't able to go Somewhere$_B$ (*to the theatre*). By combining

[1] We use the ROCstories dataset to frame the *NSC* task.

Relation Type	Dimensions
Cause (Dim 1-5)	(1) Event that directly causes or enables X; (2) Emotion or basic human drive that motivates X; (3) Location state that enables X; (4) A possession state that enables X; (5) Other attribute that enables X.
Effect (Dim 6-10)	(6) An event that is directly caused or enabled by X; (7) An emotion that is caused by X; (8) A change of location that X results in; (9) A change of possession that X results in; (10) Other change in attribute that X results in.

Table 7.1 Causal Relation types and their mapped relations (Mostafazadeh et al., 2020b).

these inferences, we can obtain a representation from which to generate a connection that completes the story, e.g., *Janie's boss wanted her to look after the issue*$_{(s_3)}$. *She missed the theatre play*$_{(s_4)}$.

We simulate this process by designing a model that incrementally generates contextualized inference rules from the given context and makes use of this knowledge to generate missing story sentences.

7.3 Discourse-Aware Inference Rules

This section details how we construct training data for the NSC task, by enriching stories with automatically predicted contextualized inferences.[2] We utilize the GLUCOSE (Mostafazadeh et al., 2020b) dataset, which contains implicit commonsense knowledge in form of semi-structured general and specific inference rules[3] (cf. Table 7.1) that are grounded in the context of individual stories from ROCStories. In GLUCOSE, given a story S and a selected sentence X from the story, the authors define ten dimensions d of commonsense causal explanations related to X, inspired by human cognitive psychology. Only a small part of ROCStories is annotated with GLUCOSE inferences (Table 7.3).

Given the amount of commonsense knowledge needed for real-world tasks, a static knowledge resource is always incomplete. Thus, we *fine-tune* a pre-trained GPT-2 model on the annotated part of GLUCOSE to *dynamically* generate inference rules for each sentence X_i

[2]For testing we rely on GLUCOSE's manually validated inference rules on a small subset of the ROCStories corpus.

[3]*Specific* means rules grounded in a given context and *general* corresponds to rules that are applicable to other contexts.

(1)Incomplete Story:	s_1: Jane loved cooking. s_2: Everyone else in her family did too. s_5: **Eventually she learned everything there was to teach.**
Gold:	Someone$_A$ loves Something$_A$ (that is an activity) >CAUSES/ENABLES> Someone$_A$ learns everything there is to learn.
	Jane loves cooking >CAUSES/ENABLES> Jane learns everything there is to learn
COINS:	Someone$_A$ is a quick learner >CAUSES/ENABLES> Someone$_A$ learns everything there is to learn.
	Jane is a quick learner >CAUSES/ENABLES> Jane learns everything there is to learn.
(2) Incomplete Story:	s_1: Seth was at a party with his friends. s_2: **Someone dared a kid to climb on a wall.** s_5: He immediately began screaming that his leg was broken.
Missing Sentences:	s_3: The kid climbed to the top and everyone cheered. s_4: Suddenly he slipped and fell to the ground.
Gold:	Some People$_A$ (who should not be there) start daring a Someone$_C$ to climb a Something$_C$ (without safety gear) >Causes/Enables> Someone$_C$ (who should not be there makes it to the top then falls down and Someone$_C$ (who is acting like monkey)).
	The kids start daring a kid to climb the wall >CAUSES/ENABLES> He makes it to the top then falls down and breaks his leg.
Fine-tuned GPT-2:	Some People$_B$ start daring a Someone$_A$ to climb a Something$_C$ >Causes/Enables> Someone$_A$ quickly shouted that his leg was broken.
	Someone start daring a kid to climb the wall >CAUSES/ENABLES> He shouted that his leg was broken.
COINS:	Some People$_B$ start daring a Someone$_A$ to climb a Something$_C$ >Causes/Enables> Someone$_A$ is on top of Somewhere$_A$
	Someone start daring a kid to climb the wall >Causes/Enables> He climbed at the top.

Table 7.2 Example of inference rules generated by COINS (compared to *Gold* from GLUCOSE). Grey: context-specific rules (SR); regular: general rules (GR). Bolded sentence s_5 is X, CAUSE is the relation type r. The second example of inference rules generated by COINS and *Fine-tuned* GPT-2 when 2-sentences are missing (compared to *Gold* from GLUCOSE). Bolded sentence s_2 is X, EFFECT is the relation type r.

Dataset	**Relation Type**	**Train**	**Dev**	**Test**
NSC		88,344	4,908	4,909
GLUCOSE	**Effect**	2949	849	–
	Cause	2944	916	–

Table 7.3 Dataset Statistics: number of unique stories.

of each story S_i from the underlying ROCStories data. We *fine-tune* two separate language models CSI_{gen} and CSI_{spec} for general and specific rules, respectively (Table 7.2).

The 10 dimensions d in GLUCOSE cover implicit *causes* and *effects* of a sentence X in a given story. In our work, we are interested in inference rules that explain a sentence's causes and effects, to study the impact of such inferences on narrative story completion. We therefore cluster all dimensions d into the two categories EFFECT vs. CAUSE (Table 7.1) and aggregate all rules from the respective categories (preserving their dimensions). Once

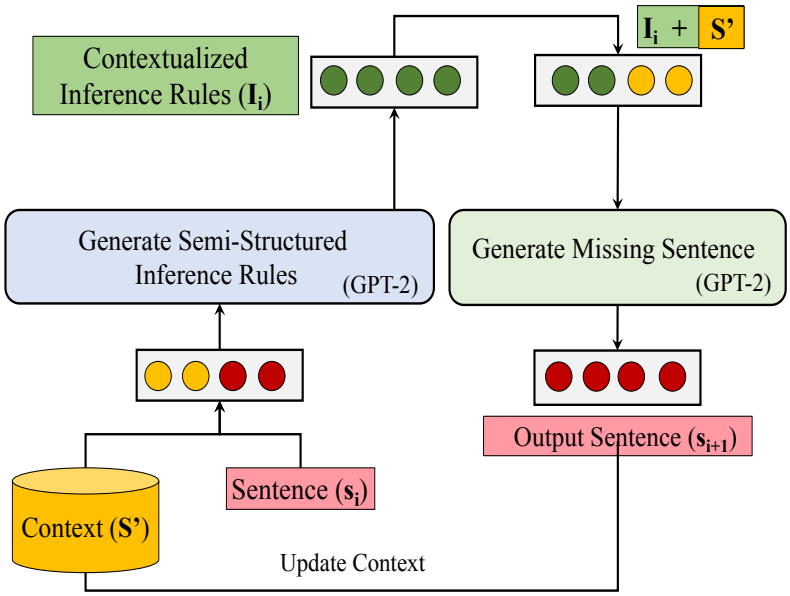


Fig. 7.2 Architecture of the COINS model.

our models (CSI_{gen}, CSI_{spec}) are trained, we apply them to our *NSC* task training data, to enrich it with inference rules for each sentence and story.

7.4 COINS: COntextualized Inference and Narrative Story Completion Model

In this section we introduce a recursively operating reasoning and sentence generation model: COINS. An overview is given in Figure 7.2. In each iteration, the model applies two consecutive steps:
(1) *Inference Step*: Given an *incomplete story context* S'= $X \oplus S_i$ and relation r, an *inference model* CSI (gen or $spec$) generates COntextualized inference rules of type r.
(2) *Generation Step*: a *sentence generator* reads the generated inference rules concatenated with the current context S' and generates the next story sentence s_{i+1}. The context S' is updated with s_{i+1} and steps (1) and (2) are repeated (cf. Algorithm 1).
This formulation allows us to i) examine inference and generation capabilities separately from each other, ii) helps determine the impact of inferential knowledge on story generation, and iii) can give us insight into how knowledge can guide story generation in a recursive inference framework.

Inference Step. We define the initial story context S' = $\{s_1, s_2$,[SEP], $s_n\}$, a selected sentence as s_i, and relation type $r \in \{$EFFECT, CAUSE$\}$, where $i \in [2, \ldots n\text{-}1]$, s_i=$\{w_1^{s_i}, .., w_v^{s_i}\}$. We adopt a pretrained GPT-2 (base) (Radford et al., 2019) transformer model with multiple Transformer blocks of multi-head self-attention and fully connected layers. During training, in each iteration the input to the model is a concatenation of the current source (S', s_i, r) and target sequence i.e., the inference rules (E_i or C_i). Eq. (1) defines the inference rule ($\mathcal{IR}$) generation model:

$$\begin{aligned} h_p^0 &= e_p + P_p, \\ h_p^l &= block(h_{<p}^{l-1}), l \in [1, L] \\ p(y_p|y_{<p}, p) &= softmax(h_p^L W^T) \end{aligned} \tag{7.1}$$

where h_p^0 is a summation of token embedding e_p and position embedding P_p for the p-th token; h_p^l is the l-th layer's output at position p, computed through transformer blocks with the masked multi-head self attention mechanism; h_p^L is the final layer's hidden state and $y_{<p}$ indicates the left context of position p. The softmax layer defines the model to output the most probable target sequence: the most likely inference rules (E_i and C_i) for each relation type (cf. Algorithm Line 4-5).

During training, we minimize the objective (2)

$$\begin{aligned} \mathcal{L}_I(\beta) = &- \sum_{k=m}^{m+N} log\ p(E_i^k|S', s_i, \text{EFFECT}) \\ &- \sum_{k=m}^{m+N} log\ p(C_i^k|S', s_n, \text{CAUSE}) \end{aligned} \tag{7.2}$$

where m, N denote the number of tokens in the source (S', s_i, r) and target sequence (inference rules) respectively; β refers to model parameters.

In this work, we focus on the NSC task, which requires our model to capture temporal dependencies and causal relationships between events. While we designed our sentence generation model in such a way that it can utilize inference rules from both forward and backward directions for each sentence, we here trigger the generation of CAUSE inference rules for s_n, since we expect that *events*, *motivations* or *attributes* that **cause** s_n will be relevant for generating the preceding sentences $[s_3, \ldots s_{n-1}]$. Similarly, we generate EFFECT relations for s_i, assuming that an *event*, changes of *emotion* or changes of *attribute* that are possible **effects** caused by s_i will be most relevant for generating the missing follow-up sentences. In principle, however, for NSC and other story generation tasks, we may consider

Algorithm 1 COINS

Require: Initial Context (S' = $\{s_1, s_2, [SEP], s_n\}$)

$Mem_{\mathcal{IR}} \leftarrow$ empty
$\mathcal{G}en\mathcal{S} \leftarrow$ empty list
for $i \leftarrow 2$ to $n-1$ **do**
    E_i = ***GenInferenceRules***(S', s_i, EFFECT)
    C_i = ***GenInferenceRules***(S', s_n, CAUSE)
    I_i = $E_i \oplus C_i$
    s_{i+1} = ***GenNewSentence***(I_i, S')
    $\mathcal{G}en\mathcal{S}$:= $\mathcal{G}en\mathcal{S}$ + s_{i+1}
    $Mem_{\mathcal{IR}}$:= $Mem_{\mathcal{IR}} \oplus I_i$
    $\mathcal{L}_{\mathcal{S}}$ += $-log_{p_{(\theta)}}(s_{i+1}|I_i, S') - log_{p_{(\beta)}}(I_i|S')$
    $\mathcal{L}_{\mathcal{IR}}$ += $-log_{p_{(\theta)}}(s_{i+1}|I_i, S') - log_{p_{(\beta)}}(I_i|S')$
    $S' := \{s_1, s_2, s_{i+1}, [SEP], s_n\}$
end for
return $\mathcal{G}en\mathcal{S}$, $Mem_{\mathcal{IR}}$

CAUSE and EFFECT relations for all sentences, letting the model freely choose from the full space of inferences.

We concatenate the generated inference rules ($I_i = E_i \oplus C_i$)[4] and store the last hidden representation in $Mem_{\mathcal{IR}} \in \mathbb{R}^{N \times L \times H}$, where N is the number of sentences, L the maximum inference sequence length and H the hidden state dimensions. $Mem_{\mathcal{IR}}$ is updated with the hidden representations of inference rules in each iteration. Hence, $Mem_{\mathcal{IR}}$ could act as an intermediate representation, and as a basis for providing *explanations* for observed story sentence generations. $Mem_{\mathcal{IR}}$ may also be used as a memory for long-form text generation tasks, to keep track of implicit knowledge *triggered by* previously generated text, and could support flexible discourse serialization patterns.[5]

Generation Step. Given the generated inference rules I_i (in form of tokens) and the incomplete story context S', we aim to generate the next missing sentence. We pass the input through another pretrained GPT-2 (base) model (cf. Equation 7.1). The loss function for the sentence generator is

$$\mathcal{L}_S(\theta) = -\sum_{k=1}^{v} log\ P(y_k^{s_{i+1}}|I_i, [EOK], S') \tag{7.3}$$

[4]We use $[SEP]$ token to delimit the individual E_i and C_i when concatenating them.
[5]We leave such extensions to future work.

where y_k denotes the k-th token and v the maximum length of the generated sentence; $i \in [2, n-1]$; $[EOK]$ denotes the end of knowledge rule tokens, and θ refers to model parameters.

Update Story Context. In the final step we update the story context by inserting the generated sentence s_{i+1} into the previous story context (cf. Algorithm 1, line 12).

Training and Inference. We add the losses $\mathcal{L}_{\mathcal{I}}$ for inference generation and $\mathcal{L}_{\mathcal{S}}$ for sentence generation to make the models dependent on each other (Algorithm 1, line. 10-11). For both the inference and the generation step model, we minimize the negative log likelihood loss of the respective target sequence.

7.5 Experimental Setup

7.5.1 Dataset

We apply COINS to the *NSC* and the *Story Ending Generation* tasks.[6] For data statistics see Table 7.3. **Narrative Story Completion.** We follow the task definition as introduced in §7.2.

Data Collection. We construct the *NSC* dataset on the basis of the ROCStories corpus (Mostafazadeh et al., 2016), which contains 98,162 five-sentence stories with a clear beginning and ending, thus making it a good choice for this task. We choose the first two sentences (s_1, s_2) as beginning rather than just s_1 because the first sentence (s_1) tends to be short in length, and usually introduces characters or sets the scene (Mostafazadeh et al., 2016), wherease the second sentence (s_2) provides more information about the initial story.

7.5.2 Hyperparameter Details

Parameter size. For GPT-2 we use the GPT-2 small checkpoint (117M parameters) based on the implementation of HuggingFace (Wolf et al., 2020).

Decoding Strategy. In the inference stage, we adopt beam search decoding with a beam size of 5 for all our models and all baselines we produce.

We used the following set of hyperparameters for our COINS model: batch size: $\{2, 4\}$; epochs: $\{3, 5\}$; learning rate: $\{1e\text{-}5, 5e\text{-}6\}$. We use Adam Optimizer, and dropout rate = 0.1. We ran our experiments with GPU sizes of 11GB and 24GB.

[6]The results for *Story Ending Generation* will corroborate our results for *NSC*. All details are given in the *Appendix*.

7.5.3 Baselines

We compare our COINS model to the following baselines:

(a) **GPT-2** (Radford et al., 2018) (with 12-layer, 768-hidden, 12-heads), trained with an objective to predict the next word. The input to the GPT-2 model is the concatenation of the source and the target story sequence. We follow the standard procedure to fine-tune GPT-2 on the NSC task during training and minimize the loss function:

$$-log(s_3, s_4|[SOS]s_1, s_2, [SEP], s_5[EOS]) \tag{7.4}$$

(b) **Knowledge-Enhanced GPT-2** (**KE**) (Guan et al., 2020) is the current SOTA for ROCStories generation. It first fine-tunes a pre-trained GPT-2 (small) model with knowledge triples from commonsense datasets (ConceptNet [CN] Speer et al. (2017) and ATOMIC [AT] Sap et al. (2020b)). The knowledge triples were converted to sentences using templates. A multitask learning framework further fine-tunes this model on both the *Story Ending Generation task* and classifying corrupted stories from real ones. As our baseline we choose the version without multi-tasking, since the corrupted story setting is not applicable for the *NSC* task. More details about the KE model are given in §3.2.

(c) **GRF** (Ji et al., 2020) is the current SOTA for the *Abductive Reasoning* and the *Story Ending Generation* tasks. GRF enables pre-trained models (GPT-2 small) with dynamic multi-hop reasoning on multi-relational paths extracted from the external ConceptNet commonsense knowledge graph. More details about the GRF model are given in §3.2.

(d) **GLUCOSE-GPT-2** Similar to Guan et al. (2020), we *fine-tune* pretrained GPT-2 (small) on the GLUCOSE dataset using *general rules* (GR). We follow the same procedure as Guan et al. (2020) and (i) first fine-tune a pre-trained GPT-2 , but here on the GLUCOSE dataset, with the following loss:

$$-log(I_i|S, s_i, r), \tag{7.5}$$

where r: CAUSE/EFFECT, I_i: Inference rules. (ii) Then we fine-tune the above model again on the NSC dataset with the following loss:

$$-log(s_3, s_4|[SOS]s_1, s_2, [SEP], s_5[EOS]) \tag{7.6}$$

The main difference between GLUCOSE-GPT-2 and COINS is: **COINS** explicitly learns to generate (contextualized) inference rules *on the fly* during the inference step and incorporates them in the story generation step.

Model	Knowledge	PPL (↓)	BLEU-1/2 (↑)	ROUGE-L (↑)
GPT-2	–	11.56	16.66/6.8	17.2
KE	[CN, AT]	12.61	17.55/7.6	17.9
GLUCOSE-GPT-2	[GL]	12.7	17.9/7.8	17.5
GRF	[CN]	12.18	20.8/8.2	17.6
COINS (SR)	[GL]	**6.7**	22.53/10.10	18.9
COINS (GR)	[GL]	6.9	**22.82/10.52**	**19.4**
COINS Oracle (SR) (Test-only)	[GL]	–	30.75/22.76	32.5
COINS Oracle (GR) (Test-only)	[GL]	–	26.37/17.01	27.38
Human	–	24.53/12.10	20.2	

Table 7.4 Automatic evaluation results for Story Completion. Best performance highlighted in **bold**; used Inference Rule types: specific (SR), general (GR).

7.5.4 Automatic Evaluation Metric

For automatic evaluation in the *NSC* task we use as metrics Perplexity (indicates fluency of text generation), BLEU-1/2 (Papineni et al., 2002) and ROUGE-L (Lin, 2004). We report performance on the test sets by averaging results obtained for 5 different seeds. All improvements across all model variants are statistically significant at $p < 0.05$.

7.6 Evaluation and Results

7.6.1 Automatic Evaluation

Our experimental results are summarised in Tables 7.4 and 7.6.

NSC task. Table 7.4 shows the results for the models described in §6.3 and evaluated as per §6.4. We observe the following: (i) COINS outperforms all strong baseline models that utilize pre-trained language models and incorporate external commonsense knowledge with respect to all automatic evaluation metrics. Note that **GLUCOSE-GPT2** and **COINS** are using the same knowledge resource, hence the clear performance increase of COINS ($+4.92$ BLEU score) indicates that jointly learning to generate contextualized inferences rules and missing sentences in a recursive manner can enhance generation quality.[7] (ii) Similar to Ji et al. (2020) we observe that fine-tuning GPT-2 over knowledge triples ([CN], [AT]OMIC or [GL]UCOSE) doesn't improve the overall performance by much (Table 7.4, line 2: [CN+AT] vs. line 3: [GL] vs. line 1: [no CSK]). (iii) For COINS, *general rules* (GR) boost performance

[7] Since **GRF**'s architecture is specific for ConceptNet, we cannot exclude that the better performance of COINS ($+2.2$ BLEU) is in part due to differences in the used knowledge.

Input	PPL (↓)	BLEU-1/2 (↑)	ROUGE-L (↑)
IR only (GR)	13.05	10.65/4.01	6.31
IR only (SR)	8.01	15.65/6.08	15.31
No IR + w/oSE	11.5	15.12/5.95	12.47
IR (GR) + w/oSE	7.49	21.50/9.78	18.07

Table 7.5 Impact of different inputs to COINS for Story Completion, SR: specific rules, GR: general rules, IR: inference rules, **w/oSE**: w/o the story ending (s_n).

	Full Context		1-Missing Sentence		2-Missing Sentence	
Model	**E**	**C**	**E**	**C**	**E**	**C**
GPT-2†	58.3	**63.3**	56.5	58.3	55.4	53.9
COINS	**59.9**	62.9	**58.6**	**60.3**	**57.5**	**56.8**
GPT-2†	57.7	59.5	55.5	55.3	53.4	51.4
COINS	57.8	**60.1**	**56.3**	**58.2**	**55.1**	**55.2**

Table 7.6 Automatic evaluation of the quality of inference rules in different context settings. Best results in **bold**. Metric: BLEU-1 scores, **E**: EFFECT, **C**: CAUSE, Grey: context-specific rules (SR); regular: general rules (GR), †: *fine-tuned* on GLUCOSE dataset.

more than specific rules, indicating that the sentence generation model generalizes well. (iv) In the oracle settings at inference time we provide the model with the silver inference rules (generated as per §7.3) that use the complete story context as background. The result indicates that SR performs better than GR when the model sees the full story context.

In general we observe that story generation benefits from higher-quality, contextualized inference rules from GLUCOSE (for COINS).[8] The improvement of COINS over GLUCOSE-GPT-2 indicates that our model is well able to utilize and profit from the inference rules. In the oracle setting, SR performs much better than GR. This is expected, since oracle rules with access to the full context will deliver more contextually-relevant inferences, while GR rules may diverge more from the story context. However, in the realistic NSC task setting (Table 7.4, lines 5,6) GR outperforms SR, which again underlines the generalization capacities of COINS.

Impact of different inputs for the Generation Step. In Table 7.5 we investigate the performance of COINS with different inputs to the sentence generation component *at inference time*: (i) When only inference rules (from the inference step) are given to the model without any story context (S' = $\{s_1, s_2$,[SEP], $s_n\}$) (**IR only**), sentence generation benefits

[8] Automatic (silver) GLUCOSE inference rules (cf. §7.3) of type GR yield 60.8 BLEU score i.e., performance of CSI_{gen} (avg. of both relation types).

when specific rules are used. This is expected since the specific rules contain statements with concrete character names and paraphrased events from the story. (ii) When only the story beginning ($s_{1,2}$) is provided to the sentence generation model *without* the ending sentence s_n (**w/oSE**) nor inference rules (**w/oIR**) we observe that the performance drops compared to models given the full incomplete context (S'), indicating that knowing the story ending helps the model to generate missing sentences that are coherent with the story. However, (iii) when adding inference rules **IR** (from the inference step i.e., E_i + C_i) to the context ($s_{1,2}$) without ending sentence (**w/oSE**), performance again improves (+5.85 BLEU scores). Note that the inference rule contains the CAUSE relation for s_n. This indicates that the model is able to utilize inference rules for story generation.[9]

Performance of inference rule generation. We now investigate how difficult it is to generate contextualized inference rules (specific and general) when multiple sentences are missing from a story. For this we compare COINS to a GPT-2 model fine-tuned on GLUCOSE data to generate inference rules (cf. §4). We study the impact of jointly and dynamically learning sentence and inference rule generation (in COINS) on the inference generation task – while the fine-tuned GPT-2 model only learns to generate inference rules conditioned on the static story context. We specifically examine the difficulty of generating inference rules *for two consecutive sentences* (s_3 and s_4) in a 5-sentence context, as opposed to shorter sequences, in three different scenarios: i) when the *complete story context* S is given; ii) when *the incomplete context* S' (i.e., s_1, s_2 and s_5) is given, plus either s_3 or s_4 (**1-missing sentence**), and iii) when S' is given, but neither of the intermediate sentences s_3 and s_4 (**2-missing sentences**). In each setting, we generate EFFECT and CAUSE rules for the targeted sentences s_3, s_4, and compare their quality. The results are reported in Table 7.6. We observe that in the **2-missing sentences** setting, COINS outperforms GPT-2 (by $+2.3$ BLEU score on average). This indicates that learning to perform inference rule generation jointly with sentence generation is beneficial for filling-in multiple story sentences. Interestingly, for increasing numbers of missing sentences, performance drops drastically for CAUSE (as opposed to EFFECT), but less so for COINS as opposed to GPT-2.

A possible reason for this may be the conditional, uni-directional nature of the underlying GPT-2 language model, which is trained to predict follow-up words in forward direction. This may favor future-directed EFFECT rules – as opposed to CAUSE relations.

[9]Here, we report the results with generalized rules as GR works better than SR when context is given (cf. Table. 7.4).

Models	Knowledge of Base Model	Coherence Win(%)	Tie(%)	Loss(%)	κ	Grammaticality Win(%)	Tie(%)	Loss(%)	κ
COINS vs **GPT-2**	None	54.7	32.0	13.3	0.52	45.7	41.3	13.0	0.49
COINS vs **GLUC.-GPT-2**	GLUCOSE	52.0	33.0	15.0	0.43	31.7	54.3	14.0	0.45
COINS vs **KE**	CN + ATOMIC	50.0	32.0	18.0	0.44	21.3	69.7	9.0	0.37
COINS vs **GRF**	CN	50.5	30.5	19.0	0.48	20.5	70.0	9.5	0.35

Table 7.7 Manual evaluation of sentence generation quality of COINS (GR) for 100 stories. Scores are percentages of *Win, Loss,* or *Tie* when comparing COINS to baselines. Fleiss' kappa κ: fair agreement or moderate agreement.

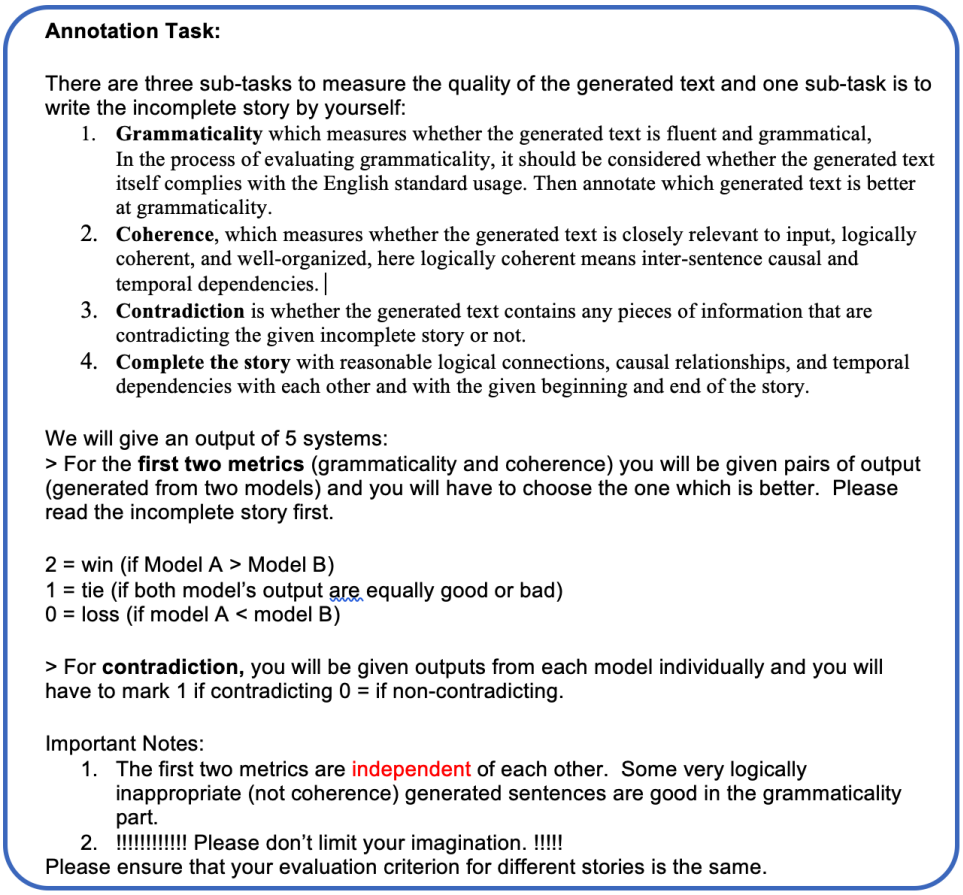


Fig. 7.3 A screenshot of the annotation guidelines for manual evaluation.

The milder effect on COINS could indicate that the concurrent inference model supports the sentence generation model to overcome this weakness.[10]

[10]In future work, we will test the above hypothesis by experimenting with a bi-directional transformer generation model.

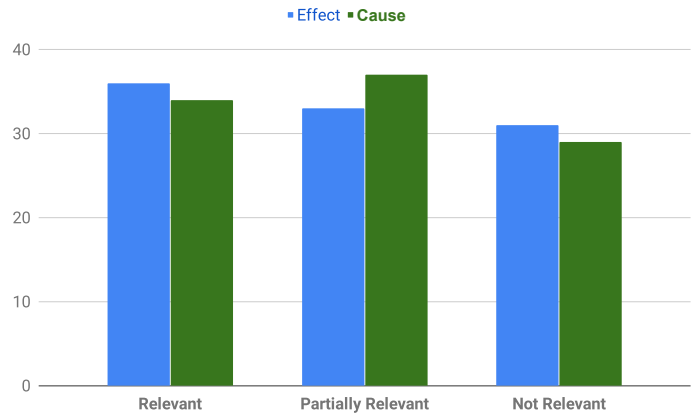


Fig. 7.4 Human evaluation of the relevance of Inference Rules generated by COINS.

7.6.2 Manual Evaluation

Automatic metrics can give us some indication of NLG quality, however, these metrics do not necessarily reflect the coherence of generated story sentences. We thus conduct a human evaluation focusing on the grammaticality and coherence of the generated sentences in their story context. We conduct pairwise comparisons for 100 randomly sampled instances of our best model, i.e., COINS with GR (according to automatic metrics) with four strong baseline models (GPT-2, GLUCOSE-GPT-2, GRF, KE). For each pair of instances (one from COINS, the other from a baseline model), we present the generated sentences in their story context, and asked three annotators to give a *preference rating* (*win*, *tie*, *lose*) according to the criteria *grammaticality* and *coherence*. For grammaticality, we present each sentence in isolation and ask the annotators to rate which sentence is more fluent, readable, and compliant with the English standard usage. For coherence, we ask the annotators to assess which of the two generated sentences are more logically coherent with each other and the story beginning and ending, in terms of causal and temporal dependencies. We applied majority voting among the three annotators (who are with a *linguistic* background) to obtain final decisions. Figure 7.3, shows a screenshot of the annotation guidelines.

The human evaluation results are presented in Table 7.7.[11] The results show that our model produces more coherent and more grammatically correct sentences compared to all baselines. This indicates that with support of learned contextualized inference rules based on GLUCOSE knowledge, our model generates more coherent story sentences that are causally and temporally well connected.

[11] We report inter-annotator agreement scores calculated with Fless' kappa κ Fleiss (1971), calculated for each comparison. We find moderate or fair agreement.

Incomplete Story:	s_1: Ken was driving around in the snow. s_2: He needed to get home from work. s_5: His tires lost traction and he hit a tree.
Missing Sentences:	s_3: He was driving slowly to avoid accidents. s_4: Unfortunately the roads were too slick and Ken lost control.
COINS (I_{GR})	Someone$_A$ is going Somewhere$_B$ ≻Cause/Enables≻ Someone$_A$ is at Somewhere$_B$, Someone$_A$ **is driving Something$_A$ fast** ≻Cause/Enables≻ Something$_A$ hits Something$_B$ (that is a tree), Someone$_A$ possess(es) Something$_A$ (that is **a car**) ≻Enables≻> Something$_A$ (tires) lost Something$_B$ (traction)
COINS (I_{SR})	He posses(es) a car ≻result in≻ His tires lost traction, He needed to get home ≻Enables≻ He drove home, He was **driving on ice** ≻ Causes/Enables ≻ His tires lost traction, He was driving on ice ≻Causes/Enables≻ He **lost control** of his vehicle.
COINS(MS_{GR})	He was **driving too fast**. He lost control of his **car**.
COINS(MS_{SR})	He was **driving on ice**. He **lost control of his vehicle**.
GPT-2	He stopped at a gas station. He filled his tank.
GPT-2 GLUCOSE	When he got to the house he realized he was stuck. Ken had to pull over to get help.
GRF	He pulled over to see what was wrong. He saw that his car was stuck in the snow.
Human	He was going very fast. The street was slippery from the snow.

Table 7.8 Example 1: inference rules and missing sentences generated by COINS (compared to *Gold* from GLUCOSE, Green), as well as baseline model generations. Gray: COINS (SR); Regular: COINS (GR); MS: missing sentences, I: inference rules. The orange and blue colors denoting the overlap in tokens between the inference rules and generated missing sentences.

Relevance of Generated Inferences Rules. We further conduct human evaluation to validate the effectiveness and relevance of the generated inference rules. We randomly select 50 instances from the NSC development set. We asked three annotators to evaluate the (GR) inference rules[12]. We define an inference rule to be relevant if (a) it captures implicit causes and effects of a selected sentence X given an incomplete story S', and (b) it is providing useful explanations for the incomplete story S'. The result for this evaluation is shown in Fig.7.4, for EFFECT and CAUSE relations. We find that in 36% and 34% of cases for effects and causes, respectively (computed on the basis of majority agreement), our algorithm was able to generate relevant inference rules. Our annotations yielded fair inter-annotator agreement of Fleiss' $\kappa = 0.45$.

[12]We report only COINS (GR), our best model according to automatic metrics.

Incomplete Story:	s_1: Danielle dreamed of living in California. s_2: After college she had to decide where to live. [mask] s_5: She loved it there.
Missing Sentences:	s_3: She could move back home or move to California. s_4: Danielle decided to take a leap and move to California.
COINS (I_{GR})	Someone$_A$ decide Something$_A$ (where to live) >Causes/Enables> Someone$_A$ decides to live in Somewhere$_A$.
COINS (I_{SR})	She had to decide where to live >Causes/Enables> She chose to **live in California**.
COINS(MS_{GR})	She decided to live in California. She settled in California.
COINS(MS_{SR})	She decided to **live in California**. She went to the beach.
GPT-2	She finally settled in California. She loved it there.
GPT-2 GLUCOSE	She decided to move to NH. She found a nice apartment there.
GRF	She decided to move to California. She found a great place to live.

Table 7.9 Example 2: Comparing COINS with different baselines on generating inference rules and missing story sentence

Case Study. We provide an example from NSC with different generation outputs (Table 7.8and 7.9). We noticed that among the baseline models, GRF performs better. Note that the generated sentences are grounded to the inference rules obtained from the inference step. Hence, the rules provide controllability, an intermediate representation and explanations for how knowledge can guide or influence story generation. In Table 7.8, comparing gold missing sentences with human written missing sentences, we observe the first sentence is semantically opposite. The result reflects the difficulty of the task and the requirement of better evaluation for natural language generation tasks.

7.7 Summary

In this chapter, we addressed a Narrative Story Completion task that allows us to probe the coherence capabilities of a neural generation model. We proposed COINS, a model that iteratively generates commonsense inference rules grounded in the context and generates story sentences, using the generated inferences as a guide.

Human and automatic evaluations show that the model outperforms strong commonsense knowledge-based generation models. By individuating the inference rule and sentence generation steps, COINS can make the contribution of commonsense knowledge on story generation transparent. The recursive nature of the inference-driven generation model holds potential for knowledge-driven control in the generation of longer sequences. In future work we will explore how an enhanced memory of generated inferences can realize more complex narrative patterns that diverge from strictly ordered narrative sequences.

Chapter 8

Conclusions & Future Work

8.1 Conclusions

"The truth of a theory can never be proven, for one never knows if future experience will contradict its conclusions."

– Albert Einstein

This dissertation investigates methods for integrating implicit knowledge into NLP systems to address social commonsense reasoning in text. Our study focused on two sub-problems of endowing machines with social commonsense reasoning: learning implicit social dynamics in text and explicitly integrating and reasoning over such knowledge for social commonsense reasoning. This chapter summarises our contributions, sheds light on a few shortcomings of our methods, and discusses the scope for future research.

We studied the importance of implicit knowledge in understanding of social dynamics. In **Chapter 4**, we presented a method that leverages graph structure to extract multi-hop commonsense knowledge from large KGs. We showed that 34% of the extracted knowledge is relevant and 42% is partially relevant. We presented an end-to-end model that incorporates multi-hop commonsense knowledge using an attention mechanism to predict the mental states of story characters. We found that implicit knowledge is crucial for predicting story characters' human needs and motives better.

We also studied the role of temporal knowledge of social events on abductive commonsense reasoning tasks. In **Chapter 5**, we proposed to fine-tune LMs to learn what events could follow other events in a social situation. We presented methods for addressing the abductive reasoning task both in unsupervised and supervised settings. We find that our unsupervised model outperforms strong supervised natural language inference baseline models.

The relatively strong performance of our proposed models demonstrates that learning to choose from generated hypothetical next events, the one that is most similar to the observation supports the prediction of the most plausible hypothesis.

Next, we explored methods to explicitly reason over implicit social commonsense knowledge. In **Chapter 6**, we proposed a multi-head knowledge attention method that encodes semi-structured inferential knowledge rules and learns to incorporate them into transformer-based models. We designed a new task for counterfactual invariance prediction. This complex task requires causal narrative chains and reasoning in the forward direction. We showed that a model that has learned to understand and reason counterfactual situations could also support abductive reasoning in a narrative context. We manually analyzed the reasoning capabilities of our model and showed that our knowledge-enhanced model is more robust than other SOTA models.

We explored the importance of grounding a commonsense inference knowledge for downstream natural language generation (NLG) task. In **Chapter 7**, we propose a model named COINS that recursively performs an inference step (*generate inferential knowledge*) and a generation step (*generate next sentence*) using the generated inferences as a guide. We introduce a new task setting named as narrative story completion task. We observed that filling in multiple story sentences benefits from contextualized inference rules. Our finding suggests that grounding commonsense knowledge is useful for explaining a NLG system.

Recently, much research has argued that pretrained LMs already contain commonsense knowledge (Davison et al., 2019; Petroni et al., 2019; Zhou et al., 2020). Language models have several advantages over structured knowledge bases, such as no human supervision to train, no schema engineering, etc. In this thesis, we argue that structured knowledge plays an essential role in (a) making black-box deep learning models more interpretable (Chapter 4, 6 and 7)), (b) providing controllability (Chapter 7), (c) improving the overall performance of current NLP systems (Chapter 4, 6 and 7). We hope our study will shed light on the importance of structured knowledge and encourage researchers to build more interpretable and controllable NLP systems.

8.2 Discussions

"The important thing in science is not so much to obtain new facts as to discover new ways of thinking"

– Willam Bragg

In this section, we discuss some limitations and address some open research questions related to our proposed methods in this thesis.

Extract Contextualized Knowledge from large KGs. To better assess the performance of our knowledge integration methods, we need better knowledge. In this thesis, we have observed that the extracted knowledge is only 34% (Chapter 4, section 4.6), 46% (Chapter 5, section 5.5), 20% (Chapter 6, section 6.6) and 35% (Chapter 7, section 7.6.2) relevant. This result suggests that either (a) commonsense knowledge is incomplete in KGs or (b) our knowledge extraction methods need improvements or (c) we need better-contextualized knowledge. In Chapter 7, we attempted to address the problem (c) by learning to generate commonsense knowledge grounded in the context. While we show that generating contextualized knowledge is useful, the GLUCOSE (Mostafazadeh et al., 2020b) knowledge graph is small in size. Therefore, learning to generate relevant knowledge from large KGs is still a bottleneck. One potential way to address this challenge is by training a graph neural network on a knowledge graph grounded to the context to automatically learn about contextualized knowledge (Bordes et al., 2013; Riedel et al., 2013; Lin et al., 2019; Yu et al., 2019). While we focused on understanding the role of CSK, the importance of knowledge representation is understudied. One potential way to improve the performance of our method is by providing the model with better knowledge representations using GNNs.

Temporal Knowledge for understanding social dynamics. In Chapter 5, we assumed that pretrained LMs are poor in "temporal awareness," and hence, we fine-tuned LMs to extract temporal knowledge to support a commonsense reasoning task. However, the role of different temporal knowledge relations such as co-occurrence and causal are understudied. Let us consider the following pairs of events to better understand the problem,

- $Event_1$: " Paul went to a supermarket." , $Event_2$: " He bought vegetables." ;
- $Event_3$: " Paul was hungry." , $Event_4$: " He went to a restaurant." ;
- $Event_5$: " Paul felt sick." , $Event_6$: " He went to a hospital." ;
- $Event_7$: " Paul went to a restaurant" , $Event_8$: " He ordered pasta." ;

Although the precedence from $Event_7$ to $Event_8$ is logical, it might be less a "cause" compared with *Paul was "hungry"* ($Event_3$). While the event pairs ($Event_3$, $Event_4$) and ($Event_5$, $Event_6$) are causally related, the event pairs ($Event_1$, $Event_2$) and ($Event_7$, $Event_8$) are less a cause and more co-occurring events. Our current fine-tuned LM (section 5.3) aims to capture such temporal knowledge in model parameters. One potential way to better understand the role of temporal knowledge for commonsense reasoning is by explicitly studying the importance of each temporal relations (co-occurrence vs causal events).

Commonsense Reasoning as Natural Language Generation Task. The success of an NLG system can be estimated from two different perspectives: a user's success in a task and the system's success in fulfilling its purpose (Celikyilmaz et al., 2020). In practice, NLG evaluation can be categorised into three categories: (i) human-centric evaluation, (ii) untrained automatic metrics, and (iii) machine trained metrics. Evaluating commonsense reasoning as a NLG task brings a new evaluation challenge because of its plausible nature.

When *human-centric evaluation methods* are used we often observe a low inter-annotator agreement (IAA) scores (see Chapter 5 and 7). One intuitive reason for low IAA scores is highlighted in table 7.8, where we observe that the human-written missing sentences and the gold missing sentences are semantically opposite, but both are equally plausible. The *untrained automatic metrics* compares the machine-generated texts to human-generated texts based on n-gram overlap. Since commonsense reasoning is not constrained only to token overlaps, such automatic metrics are insufficient for evaluation. For example, in table 7.4, we see the overall performance of human-written text is low using automatic evaluation. Finally, the *machine-learned metrics* are also biased towards token-overlap. In chapter 5, we used BERT-score, which is a metric for evaluating generated text against gold-standard references. In table 5.5 and Figure 5.9, we observe how BERTScore is biased towards token overlap.

In this dissertation, we do not address improving the evaluation metrics for commonsense reasoning as NLG tasks. One potential future work will be to design: (i) better evaluation metrics and (ii) a revised inter-annotator metric for commonsense reasoning.

8.3 Future Research Plans

"The important thing is not to stop questioning.
Curiosity has its own reason for existing."
– Albert Einstein

We anticipate that the research presented in this dissertation will encourage different opportunities to pursue the open challenges that have not been addressed so far. Below, we

outline future directions for addressing new challenges arising from our research, and we think are worth exploring.

Evaluation of Commonsense Reasoning and beyond. Recently, several works have demonstrated that human agreement for commonsense reasoning evaluation is low (Rashkin et al., 2018a; Qin et al., 2019). One particular reason is that it involves plausible reasoning. Hence, automatically evaluating models on NLG tasks involving commonsense reasoning becomes even more challenging. It would be particularly important to focus on multidimensional metrics for evaluation. In our future work, we will focus on designing evaluation metrics that target different dimensions of commonsense reasoning such as *temporal, spatial, causal, distinctness* etc. and also on model robustness, transparency, generalization, reasoning capabilities.

Role of Structured Knowledge in Continual Learning. We humans learn and understand new concepts by building on our own memories and applying prior knowledge. In contrast, most NLP systems learn about a new task in isolation. Recently, with the advancement in transfer learning[1] methods, researchers have focused on addressing continual learning[2] for NLP tasks (Ruder et al., 2019; Chen et al., 2020). However, studies have shown when a model is incrementally fine-tuned on new data distribution; it risks forgetting (*concept drift, catastrophic forgetting*) how to treat instances of the previously learned ones (Mosbach et al., 2021; Sun et al., 2019). As our research community move towards building NLP systems that are environmentally friendly (less training time), we think continual learning will be an important research direction. Rather than storing training data and re-training from scratch, the use of a knowledge-based memory system can be useful. Hence, it will be potentially relevant to investigate the role of structured knowledge (one can treat it as prior memory) to address concepts like concept drift, catastrophic forgetting etc.

Societal Application of NLP. Research in the social commonsense reasoning area has far-reaching value for designing NLP applications that are able to interact with humans in a more natural way. Understanding and reasoning about the user's intent or writer's intent are relevant in building systems that can recommend products to end-users or detect subtle bias in social media posts. Another application where our research could be beneficial is dialogue systems, where generating dialogue in a more empathetic, natural way is important. Therefore, a future direction of our research is to connect improvements in social commonsense understanding to such user applications.

Towards Diverse and Inclusive NLP systems. Commonsense Knowledge is the knowledge that ***all*** humans typically possess, which helps them make sense of daily situations.

[1]Transfer learning methods deals with transferring knowledge from a source task to a target task to improve the performance of the target task

[2]Continual Learning is the process of building complicated skills on top of those already developed.

However, most current research efforts have focused predominantly on the English language. To expand research on social commonsense reasoning, one needs to take into account the cultural, linguistic, historical, societal and geophysical differences between people. Therefore, we will focus on developing resources that consider diverse social norms and build NLP systems that are more inclusive and equitable.

Appendix A

Application: Argumentation Relation Classification

Automatically identifying relations between argumentative text units (e.g., *support* and *attack* relations) has attracted much attention (Cabrio and Villata, 2012; Stab and Gurevych, 2014a,b, 2017). *Argumentative relation classification* (henceforth *ARC*) is the task of determining the type of relation that holds between two argumentative units (AUs, for short). This task has some overlap with *stance detection*, but differs in important aspects: while stance detection aims at determining the relation of AUs *towards a topic* or conclusion, argumentative relation classification analyzes relations *between argumentative units*. In this work we consider both *argument-topic relations* and *argument-argument relations* – since only a system that captures both types of relations can be applied in a real debate. We propose a ranking-based knowledge- knowledge-enhanced argumentative relation classification approach that we successfully apply to both (closely related) argumentative relation classification tasks.

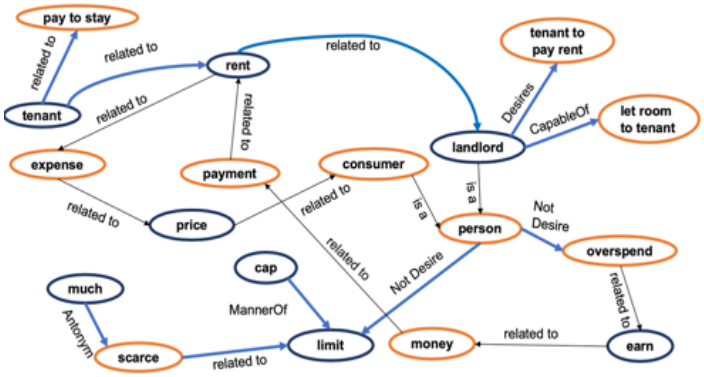


Fig. A.1 A subgraph extracted from ConceptNet. Blue edges portray relevant knowledge paths from ConceptNet. Concepts from the text in blue; intermediate nodes in orange.

Defining abstract semantic patterns is one way to explain argumentative relations (Reisert et al., 2017). Let us consider two argumentative units Arg_1 : *"Landlords may want to earn as much as possible."* and Arg_2 : *" Rent prices should be limited by a cap when there's a change of tenant."* We can observe that Arg_1 implies that *x* is ***good for*** landlords, while Arg_2 implies that *x* is ***bad for*** tenants, with *x = 'rise in price'*. This pattern can indicate *attack*. But Arg_2 states that *x should be limited* and thus the correct relation is *support* (Arg_1, Arg_2). Hence, we not only need good analysis of the text, but also further, so-called commonsense knowledge about the events, entities and relations mentioned in it, in order to gain true understanding of an argument. For example, we need to know that *landlords* and *tenants* are in a relation where one pays the other, with conflicting interests in the amount to be paid (see Fig. A.1).

In this work we propose to leverage commonsense knowledge from ConceptNet (Speer and Havasi, 2012) in order to connect pairs of concepts in argumentative units with implicit background knowledge relations. Fig. A.1 shows a semantic (sub)graph with nodes representing concepts and edges (e.g., *'not desire'*) indicating relations between them. The graph captures semantic relations between entities (*tenant – landlord*) and properties (*much – limited*).

Our hypothesis is that capturing commonsense knowledge relations within and between AUs is essential for deeper understanding of arguments, especially for aspects of practical reasoning, cf. (Walton, 2015). We investigate this hypothesis by devising a system that constructs subgraphs over pairs of AUs based on relevant concepts and multi-hop knowledge from the ConceptNet graph (Speer and Havasi, 2012). We propose a graph-based ranking method to extract relevant paths from these subgraphs that connect the argumentative units.

A.1 Argumentative Relation Classification with Commonsense Knowledge

We propose a neural Argumentative Relation Classification (ARC) system that (i) encodes pairs of argumentative units (AUs) using a cross-sentence attention mechanism over attentive BiLSTM encoders to understand their contextual features and structures; (ii) we leverage commonsense knowledge by linking concepts from the AUs to concepts from ConceptNet, and construct instance-specific subgraphs from which we extract relevant knowledge paths using graph-based ranking methods; finally (iii), we incorporate lexical knowledge from WordNet – *Synonyms* and definitions – to expand the meaning of terms in the AUs. Recently, Bauer et al. (2018b) and Paul and Frank (2019) proposed methods to select multi-hop

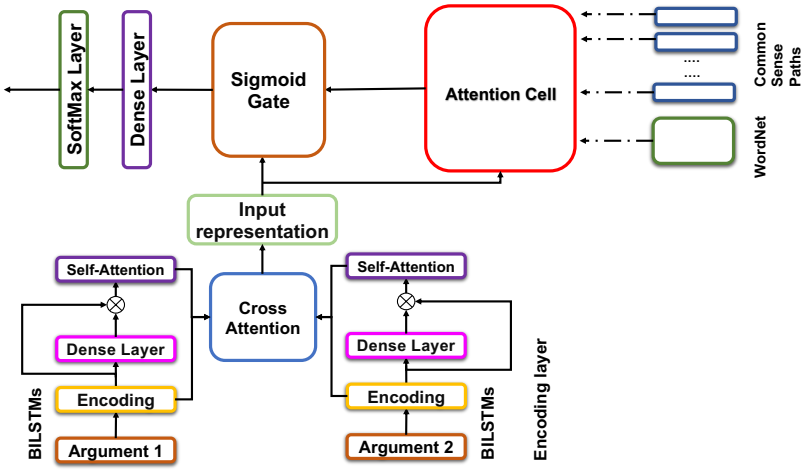


Fig. A.2 **ARK:** Argumentative relation classification (ARC) with self-attention and knowledge (ARK)

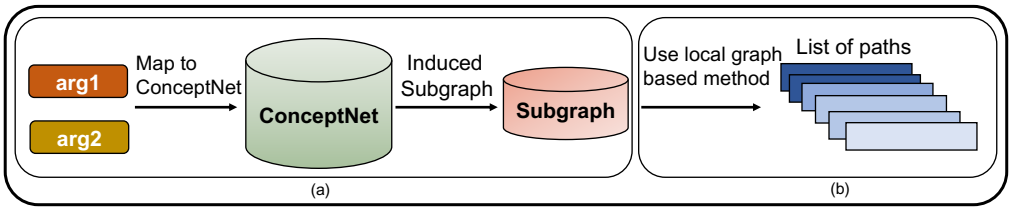


Fig. A.3 Commonsense Knowledge Extraction. Left: Subgraph Construction. Right: Ranking & Path Selection

Fig. A.4 (a) ARC model with knowledge (ARK) and (b) Commonsense Knowledge Extraction

knowledge paths for reading comprehension and human needs classification: the former use heuristics, the latter graph-based measures for selection. In our work, we construct a knowledge subgraph over AUs and use local graph measures to select relevant knowledge for predicting the correct argumentative relation class. The selected knowledge paths along with *Synonyms* and definitional knowledge are encoded and incorporated into the relation prediction component. We use an attention cell that jointly encodes the encoded argument pair representations and the selected knowledge paths to predict implicit knowledge relations during inference. Figure A.4 gives an overview of the model.

A.1.1 Argumentative Relation Classifier

The core of our model consists of three components: (1) encoding layer, (2) attention layer with *self-attention and cross-attention*, (3) output layer. The BiLSTM encoder takes two AUs $arg \in [arg1, arg2]$ as inputs: sequences of tokens $w_1^{arg}, ..., w_n^{arg}$ (or $w_{1:n}^{arg}$).

Encoding Layer We map the sequence of tokens of both AUs to sequences of word representations using word embeddings, and encode them with a single-layer BiLSTM.[1]

Attention Layer We apply ***self-attention*** to capture the contribution of each token in the argument Yang et al. (2016). We obtain argument representations x^{arg1} and x^{arg2} by taking the weighted sum of the attention scores and the hidden states that were generated by the BiLSTM.

We capture the relevance of the hidden representations of the arguments with ***cross-attention***. We calculate soft attention weights, this time across arguments and taking into account the self-attention weighted token representations from (A.1) and (A.2):

$$\hat{h_i}^{arg1} = \frac{\sigma(x_i^{arg2} {h_i}^{arg1})}{\sum_{j=1}^{N} \sigma(x_j^{arg2} {h_j}^{arg1})} \tag{A.1}$$

$$\hat{h_i}^{arg2} = \frac{\sigma(x_i^{arg1} {h_i}^{arg2})}{\sum_{j=1}^{M} \sigma(x_j^{arg1} {h_j}^{arg2})} \tag{A.2}$$

$$x_i^{arg1} = \sum_{j=1}^{N} \hat{h_j}^{arg1} h_i^{arg1}; \quad x_i^{arg2} = \sum_{j=1}^{M} \hat{h_j}^{arg2} h_i^{arg2} \tag{A.3}$$

with N, M the number of tokens in $arg1$ and $arg2$.

Output Layer We apply a final dense layer followed by softmax to predict the classes *support* or *attack*. As input y_i to this final layer we concatenate the output representations x_i^{arg1} and x_i^{arg2} from the cross-attention layer, and their difference vector $x_i^{arg1} - x_i^{arg2}$ and feed them through a projection layer: $y_i = ReLU(W_y[x_i^{arg1}; x_i^{arg2}; x_i^{arg1} - x_i^{arg2}] + b_y)$.

[1]The final state of the forward and backward pass is composed by taking the max over each dimension.

A.1.2 Commonsense Knowledge Extraction for Argumentative Relation Classification

Models for ARC will often require knowledge that is not overtly stated in the AUs or their context Rajendran et al. (2016). We aim to solve this issue by leveraging commonsense and lexical knowledge from resources such as ConceptNet and WordNet. We begin by extracting connections between concepts mentioned in pairs of AUs from ConceptNet. For each pair we (i) collect all potentially relevant relations and concepts in a subgraph and (ii) select the top-ranked paths using local graph measures. Figure A.3, gives an overview of the extraction method.

Subgraph Construction For each pair $arg1$, $arg2$ we construct a subgraph $G' = (V', E')$ from ConceptNet $G = (V, E)$ by initializing V' with all concepts $c_{arg1} \in arg1$ and $c_{arg2} \in arg2$. To do so, we remove stop words, lemmatize tokens and perform n-gram matching of the remaining tokens to concepts in G. Similar to the subgraph construction in Bauer et al. (2018b) and Paul and Frank (2019), we extend G' by including all concepts contained in the shortest paths between all concepts $c_i \in V'$ as well as all neighbouring nodes of concepts c_{arg} from $arg1$ and $arg2$. The final subgraph G' collects all edges E' from E that have both endpoints in V'.

Ranking and Selecting Paths We apply a two-step method: (i) **Collect top-n concepts**: Although most concepts in the AUs may be useful, considering all of them may introduce noise. For example, in Figure A.1, the concept *possible* in arg_0 is not especially relevant in the given context. Therefore, we filter and collect the top-n concepts from each AU arg_i by ranking all the concepts $c_{arg_i} \in arg_i$ using personalized page rank Haveliwala (2002) given the subgraph G' and all concepts $c_{arg_j} \in arg_j$ ($i \neq j$), i.e., the concepts mentioned in the other argumentative unit. (ii) **Select top-k paths**: We then collect all shortest paths between the remaining concepts (of length ≤ 4 hops). We rank each node in the path with *closeness centrality* Bavelas (1950) scores. We select the top-k paths that connect any pair of filtered concepts $c_{arg1} \in arg1$ and $c_{arg2} \in arg2$, which we denote as ***S*elected Knowledge Paths (SKP)**.

Lexical Knowledge WordNet[2] Miller (1995) is a widely used lexical resource. It defines the meaning of words and their relations for English. We employ WordNet's lexical knowledge by mapping each lemmatized token from the AUs to the WordNet graph, selecting the

[2]https://wordnet.princeton.edu/

most frequent sense. We extract its SYNONYMS and sense definition. We denote WordNet knowledge as WN and knowledge acquired from WN as **Lexical Knowledge LK**.

A.1.3 Injecting Knowledge for ARC

We leverage commonsense knowledge for the ARC task from three sources: structured knowledge from ConceptNet via *Selected Knowledge Paths (SKP)* and *Enriched Knowledge (EK)*, and unstructured *Lexical Knowledge (LK)* from WordNet. SKP, EK and LK (SYNONYMS & Definitions) can all be represented as sets of (multi- or single-hop) paths $p_{1:l}$, i.e., sequences (of length l) of nodes (concepts) and edges (relation types). For LK, each path $p_{1:l}$ consists of the sequence of words from the sense definition of word w.[3]

Encoding Layer We use a single-layer BiLSTM to obtain encodings ($h^{k,i}$) for each knowledge path (h^k the encoded knowledge path, i the path index).

Attention Cell We define a cell that allows the model to attentively encode the knowledge paths (see Figure A.2). We use an attention layer, where each encoded knowledge path interacts with the argument representations x^{arg} (A.4) (to receive attention weights ($\hat{h}^{k,i}$) from (A.5). In (A.5) we use sigmoid to calculate attention weights,

$$x^{arg} = [x_i^{arg1}; x_i^{arg2}; x_i^{arg1} - x_i^{arg2}] \tag{A.4}$$

$$\widetilde{h}^{k,i} = \sigma(x^{arg} h^{k,i}), \quad \hat{h}^{k,i} = \frac{\widetilde{h}^{k,i}}{\sum_{i=1}^{N} \widetilde{h}^{k,i}} \tag{A.5}$$

To obtain the argument-aware commonsense knowledge representation x_i^k, we pass the output of the attention layer through a feedforward layer. W_k, b_k are trainable parameters.

$$x_i^k = ReLU(W_k(\sum_{j=1}^{N} \hat{h}^{k,j} h^{k,i}) + b_k) \tag{A.6}$$

$$o_i = sigmoid(W_z[x_i^{arg}; x_i^k] + b_z) \tag{A.7}$$

To distill the selected and weighted knowledge into the model, we concatenate the argument x_i^{arg} and the knowledge x_i^k representation and process it by a dense layer (Eq. A.8), with $\odot$ element-wise multiplication, $b_{\widetilde{y}_z}$ and $W_{\widetilde{y}_z}$ trainable parameters, y_i from *Output Layer*.

[3] We use the most frequent sense of w, as defined in WordNet. We embed each path $p_{1:l}^{k,i}$ with pretrained GloVe Pennington et al. (2014) embeddings ($k \in$ {SKP, EK, LK}).

Then, a sigmoid gate helps the model select when to incorporate knowledge x_i^k (Eq. A.8).

$$z_i = \textit{softmax}(W_{\widetilde{y}_z}(o_i \odot y_i + (1 - o_i) \odot x_i^k) + b_{\widetilde{y}_z}) \tag{A.8}$$

We finally pass the representation to a softmax classifier to form a probability distribution over the two classes *attack* and *support*.

A.2 Experiments

4.1. Data There is are only few datasets for the ARC task. We use these two datasets:[4]
Student Essays. This well-established dataset comprises argumentative essays in English written by students. We use the extended v.02 with 402 essays Stab and Gurevych (2017). An issue with this data is that many of the relations can be easily identified by observing shallow discourse clues (*however*, *moreover*). Therefore, we we use the more difficult *content-based* setup Opitz and Frank (2019), where the relations between argumentative units have to be determined without looking at the textual discourse context of unit clauses.

Debatepedia The Debatepedia website[5] collects user-generated debates that each contain several arguments in favor of or opposed to the debate's topic. Topics are usually formulated as polar questions. Cabrio and Villata (2012)

created a small dataset from Debatepedia consisting of 200 pairs of topics (questions) and associated pro vs. con arguments, as well as further dependent pairs of pro and con arguments among each topic. But the pairing of coherent pro and con arguments is difficult to establish automatically. We thus restrict ourselves to pairs of directly connected questions and pro/con arguments. To construct high-quality data, we manually reformulate the questions to statements. If an argument is in favor of the debated topic, the claim *supports* the topic. Else it *attacks* it.

4.2. Linear Classifier Baseline Among other text classification tasks, linear SVMs have been successfully applied to ARC Pradhan et al. (2005); Kim (2014); Stab and Gurevych (2017); Aker et al. (2017).

Next to our neural system we thus implement an SVM model w/ and w/o knowledge enhancement. Below we describe text classification features used by our baseline SVM and

[4]Below we summarize the data statistics:

Student Essay	train:	2803 / 273 (support / attack)	dev:	1017 / 132 (support / attack)
Debatepedia	train:	3240 / 3251 (support / attack)	dev:	1121 / 1042 (support / attack)

[5]Debatepedia: `http://www.debatepedia.org`

explain ways of modeling and abstracting the knowledge paths to make them accessible for the SVM.

Text features. We feed the SVM a concatenation of the uni- and bigram (TF-IDF) representation of (i) source, (ii) target and (iii) the text overlap of source and target. We also concatenate averaged GloVe vectors to the bag-of-words feature representation; the vectors are separately averaged over (i), (ii) and (iii). We further concatenate to the vector the element-wise subtraction and multiplication of the averaged source from the averaged target GloVe vector, to model the argumentative relation as a directional vector.

Modeling paths as features. We investigate whether the extracted and selected knowledge paths (SKP) can improve the SVM classifier. But encoding paths is not straightforward for an SVM compared to encoding sequential paths with a recurrent NN. We thus apply the following steps: we represent every selected path as the mean vector of the token-wise GloVe vectors in a path. We then retrieve different path selections, e.g., the mean vector of all paths or the path-vector with the maximum and minimum norm. To determine the optimal selection jointly with the optimal SVM margin, we run a greedy hyper-parameter search on the development data. Details will be provided with the code.

4.3. Training Details **Objective** During training we minimize the cross-entropy loss between the predicted and the actual distribution. We use Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 0.001, and batch size of 8/32 for Student Essays/Debatepedia. We use pretrained GloVe (Pennington et al., 2014), ELMo (Peters et al., 2018) embeddings, a hidden size of 100 for all Dense Layers and L2 regularization with λ = 0.01. We use k = 3 for selecting top-ranked paths. For filtering the number of concepts with *personalized page rank* we use $n \leq 5$ concepts per AU. **Metrics** We report macro-averaged Precision (P), Recall (R), F1 scores.

A.3 Results

We examine 8 different systems: **random** baseline guesses labels according to the training data label distribution. **SVM** is a knowledge-agnostic linear classifier baseline. When we add selected knowledge paths via aggregation features, we denote this as **SVM+CN** and as **SVM+CWN** (for the latter (+CN) extended with WordNet). **BiLSTM** is a neural knowledge-agnostic baseline and **Bi-ATT** denotes the BiLSTM with self- and co-attention (see Fig. A.2 *w/o* Attention Cell and Sigmoid Gate). By further enriching Bi-ATT with knowledge paths through the Attention Cell, we obtain our main model: **ARK** (again in different varieties: **+CN**, etc.).

Model	WE	Student essays P	R	F1	Debatepedia P	R	F1
(1) random	-	49.68	49.66	49.65	50.04	50.03	50.01
(2) BiLSTM	G_{300d}	53.53	52.89	53.13	55.67	55.68	55.63
(3) KOB2019	G_{300d}	52.79	51.85	52.05 (2) ✗	58.06	57.75	57.04 (2) ✓
(4) KOB2019	ELMo	55.72	53.16	54.37 (2) ✓	59.16	59.17	59.11 (2) ✓
(5) SVM	G_{300d}	54.11	52.59	52.95	54.73	54.71	54.52
(6) SVM + CN	G_{300d}	54.11	54.23	54.17	56.12	56.00	55.58
(7) SVM + CWN	G_{300d}	55.80	56.38	**56.06** (5, 3) ✓	56.60	56.57	**56.37** (5) ✓ (3) ✗
(8) Bi-ATT	G_{300d}	54.46	53.31	53.70	56.20	56.19	56.18
(9) ARK + WN	G_{300d}	57.68	55.71	56.44	57.49	57.48	57.48
(10) ARK + CN	G_{300d}	57.64	57.71	57.67	57.38	57.25	57.31
(11) ARK + CWN	G_{300d}	60.70	55.55	**58.03** (8, 3) ✓	58.78	58.43	**58.60** (8, 3) ✓
(12) Bi-ATT	ELMo	56.44	54.77	55.16	59.10	59.08	59.09
(13) ARK + WN	ELMo	57.13	56.26	56.69	63.00	62.70	62.85
(14) ARK + CN	ELMo	59.13	**58.68**	58.89 (12) ✓	63.64	63.45	63.50 (12) ✓
(15) ARK + CWN	ELMo	**63.43**	55.90	**59.43** (12, 4) ✓	63.72	63.65	**63.69** (12, 4) ✓

Table A.1 Classification results. Bi-ATT BiLSTM+Attention model, ARK = ARC model + Knowledge, where CN = ConceptNet; WN = WordNet; Superscripts mark significant improvement ✓ or not ✗ of the result relative to the model the index names.

Table A.1 reports our experiment results in averaged scores over five runs. Our models enhanced with knowledge (including SVM) perform significantly better ($p < 0.05$) compared to their baselines, and similarly for ARK+CWN vs. KOB2019.

Knowledge helps The results show that adding selected knowledge to any of our baseline models improves their overall performance on both datasets and for both types of embeddings. Our full model **ARK** profits most from the added knowledge when compared to its knowledge-agnostic counterpart **Bi-ATT** (using ELMo: +4.27 pp. (percentage points) macro F1 in Student essays; +4.6 in Debatepedia; when using GloVe: +4.33 pp. in Student essays; +2.42 in Debatepedia). This finding not only applies to the global F1 metric, but also to macro Precision and Recall: we obtain considerable gains in Recall on Student essays of over 4 pp., i.e., a relative increase of more than 8%. Deeper analysis in §6 will show that knowledge helps especially for classifying rare *attack*-examples. We compare our knowledge representation and extraction method with the method in Kobbe et al. (2019). We empirically show that across two datasets and different embeddings we gain +4 F1 (on average) improvement. Knowledge also helps the linear SVM baseline (**SVM** vs. **SVM+CN/+CWN**).

For both datasets we see gains. Adding only knowledge from ConceptNet improves over SVM by +1.22 pp. macro F1 in Student essays; +1.06 in Debatepedia. With access to the full knowledge we observe a more notable gain: +3.11 pp. macro F1 in Student essays; +1.85 pp. in Debatepedia (**SVM+CWN**). The fact that a linear classifier profits less from added knowledge compared to the neural system (**Bi-ATT** vs. **ARK**) is expected: the knowledge paths are sequential and thus easier to model with recurrent computations of the neural model. When computing path aggregates to make knowledge paths accessible for the SVM, we lose important structural information.

Appendix B

COINS: Story Ending Generation Task

SEG task. We also investigate how COINS performs when applied to the task of generating a story ending when given a 4-sentence story (SEG). In this task our model takes only one iteration step to generate the story ending, where in the inference step it generates EFFECT inference rules for sentence (s_4).

Dataset	Train	Dev	Test
SEG	90,000	4,080	4,081

Table B.1 Dataset Statistics: nb. of unique stories

Data. This task is to generate a reasonable ending given a four-sentence story context Guan et al. (2019). The stories are from ROCStories Mostafazadeh et al. (2016). We use the same data splits as Guan et al. (2019).

Automatic Metrics. For Story Ending Generation (SEG) we follow the metrics used in Guan et al. (2019); Ji et al. (2020): they use BLEU-1/2 to measure n-gram overlap between generated and human-written story endings, and Distinct-n Li et al. (2016b) to measure the generation diversity using maximum mutual information.

Baselines. For the *Story Ending Generation task*, we compare COINS to the **IE+GA** model Guan et al. (2019). It is based on incremental encoding and multi-source graph attention Guan et al. (2019). We also compare to a Seq2Seq model Luong et al. (2015) based on gated recurrent units (GRU) and attention mechanism.

Model	BLEU-1/2 (↑)	Distinct-2/3 (↑)
Seq2Seq†	19.1 / 5.5	0.181 / 0.360
IE+GA†	20.8 / 6.4	0.140 / 0.280
GPT†	25.5 / 10.2	0.304 / 0.505
GPT2-OMCS†	25.5 / 10.4	0.352 / 0.589
GPT2-GLUCOSE	25.6 / 10.2	0.361 / 0.609
GRF†	26.1 / 11.0	0.378 / 0.622
COINS (GR)	**27.4 / 12.3**	**0.428 / 0.724**
COINS (Oracle)	41.80/28.40	0.479/0.786

Table B.2 Result: Automatic evaluation results on the Story Ending Generation Task, † Ji et al. (2020)

Result. In Table B.2, we observe that the COINS model outperforms all previous strong baselines, including GPT2-GLUCOSE that uses the same knowledge resource. Interestingly, we also observe that fine-tuning on GLUCOSE or ConceptNet knowledge improves the text generation diversity, indicating that the models leverage concepts and event knowledge during generation (cf. Table B.2 line.4-8).

Appendix C

Data Management

The heiDATA repository available at heiDATA/AIPHES contains the code for reproducing experiments presented in this thesis.

Resources for Chapter 4. The heiDATA repository and the instruction to run the code available at Readme contains the code for reproducing experiments presented in Chapter 4 and the corresponding NAACL-HTL paper (Paul and Frank, 2019). In particular,

- To construct the ConceptNet graph, run the code `conceptnet2graph.py`
- To constrcut the subgraph per sentence, run the code `make_sub_graph_server.py`
- To extract relevant knowledge path, run `extract_path.py`
- To train the MHKA model, run `run_experiment.sh`

Resources for Chapter 5. The heiDATA repository available at Readme contains the code for reproducing experiments presented in Chapter 5 and the corresponding StarSem paper (Paul and Frank, 2021b). In particular,

- To create the counterfactual data, run the code `create_counterfactual_data.py`
- To run the unsupervised script and get the Bert score, run the code `get_bert_score.py`

Resources for Chapter 6. The heiDATA repository available at Readme contains the code for reproducing experiments presented in Chapter 6 and the corresponding EMNLP, Findings paper (Paul and Frank, 2020). In particular,

- To extract the basic structure *who did what to whom, when and where* from each sentence in the context, we use `SRL` code from AI2.
- To generate commonsense knowledge for each events, run the code `run_generate.sh`
- To train the MHKA model run the code `run_multiple_choice_know.py`

Resources for Chapter 7. The heiDATA repository available at Readme contains the code for reproducing experiments presented in Chapter 7 and the corresponding ACL paper (Paul and Frank, 2021a).

- To train and evaluate the COINS framework, run the script `run_train.sh` and `run_test.sh` respectively.

List of Figures

List of Tables

List of Abbreviations

COPA Choice of Plausible Alternatives 14

CR Counterfactual Reasoning 79, 80

CSK Commonsense Knowledge 16, 115

KGs knowledge graphs 3, 4, 31–33, 113, 115

NLG Natural Language Generation 7, 36, 95, 96, 109, 114, 116, 117

NLP Natural Language Processing 23, 33, 34, 39, 95, 114

NLU Natural Language Understanding 3

NNs Neural Networks 22

SCK Social Commonsense Knowledge 80

SCR Social Commonsense Reasoning 1, 20, 77, 95

SOTA state-of-the-art 7, 35, 77, 87, 114

References

Ahn, S., Choi, H., Pärnamaa, T., and Bengio, Y. (2016). A neural knowledge language model. *ArXiv*, abs/1608.00318.

Aker, A., Sliwa, A., Ma, Y., Lui, R., Borad, N., Ziyaei, S., and Ghobadi, M. (2017). What works and what does not: Classifier and feature analysis for argument mining. In *Proceedings of the 4th Workshop on Argument Mining*, pages 91–96, Copenhagen, Denmark. Association for Computational Linguistics.

Alquist, J. L., Ainsworth, S. E., Baumeister, R. F., Daly, M., and Stillman, T. F. (2015). The making of might-have-beens. *Personality and Social Psychology Bulletin*, 41:268 – 283.

Alt, C., Hübner, M., and Hennig, L. (2019). Fine-tuning pre-trained transformer language models to distantly supervised relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1388–1398, Florence, Italy. Association for Computational Linguistics.

Apperly, I. (2010). *Mindreaders: the cognitive basis of" theory of mind"*. Psychology Press.

Aristotle, P. A. (1989). Hackett publishing company. *Indianapolis/Cambridge*.

Arras, L., Horn, F., Montavon, G., Müller, K.-R., and Samek, W. (2016). Explaining predictions of non-linear classifiers in NLP. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 1–7, Berlin, Germany. Association for Computational Linguistics.

Ba, J., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *Advances in NIPS 2016 Deep Learning Symposium*, abs/1607.06450.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *In Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, abs/1409.0473.

Balasubramanian, N., Soderland, S., Mausam, and Etzioni, O. (2013). Generating coherent event schemas at scale. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1721–1731, Seattle, Washington, USA. Association for Computational Linguistics.

Banerjee, P., Pal, K. K., Mitra, A., and Baral, C. (2019). Careful selection of knowledge to solve open book question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6120–6129, Florence, Italy. Association for Computational Linguistics.

Bar-Hillel, Y. (1960a). A demonstration of the non feasibility of fully automatic translation. appendix iii of 'the present status of automatic transla-tion of languages', reprinted in bar-hillel y, 1964. *Language and Information, Reading, Mass. Addi-son-Wesley*, pages 174–179.

Bar-Hillel, Y. (1960b). The present status of automatic translation of languages**this article was prepared with the sponsorship of the informations systems branch, office of naval research, under contract nr 049130. reproduction as a whole or in part for the purposes of the u. s. government is permitted. volume 1 of *Advances in Computers*, pages 91–163. Elsevier.

Bauer, L., Wang, Y., and Bansal, M. (2018a). Commonsense for Generative Multi-Hop Question Answering Tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230.

Bauer, L., Wang, Y., and Bansal, M. (2018b). Commonsense for generative multi-hop question answering tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230, Brussels, Belgium. Association for Computational Linguistics.

Bavelas, A. (1950). Communication patterns in task-oriented groups. *Journal of the Acoustical Society of America*, 22(6):725–730.

Becker, M., Korfhage, K., and Frank, A. (2021). COCO-EX: A tool for linking concepts from texts to ConceptNet. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. In *Journal of machine learning research*, page 1137–1155.

Besold, T. R., Garcez, A., Bader, S., Bowman, H., Domingos, P. M., Hitzler, P., Kühnberger, K.-U., Lamb, L., Lowd, D., Lima, P., Penning, L., Pinkas, G., Poon, H., and Zaverucha, G. (2017). Neural-symbolic learning and reasoning: A survey and interpretation. *ArXiv*, abs/1711.03902.

Bhagavatula, C., Bras, R. L., Malaviya, C., Sakaguchi, K., Holtzman, A., Rashkin, H., Downey, D., tau Yih, W., and Choi, Y. (2020). Abductive commonsense reasoning. In *International Conference on Learning Representations*.

Bhatt, M. and Flanagan, G. (2010). Spatio-temporal abduction for scenario and narrative completion (a preliminary statement). In *ECAI*.

Bian, N., Han, X., Chen, B., and Sun, L. (2021). Benchmarking knowledge-enhanced commonsense question answering via knowledge-to-text transformation. *AAAI*, abs/2101.00760.

Birnbaum, L. (1991). Rigor mortis: A response to nilsson's "logic and artificial intelligence". *Artif. Intell.*, 47:57–77.

Bisk, Y., Zellers, R., Bras, R. L., Gao, J., and Choi, Y. (2020). Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Boratko, M., Li, X., O'Gorman, T., Das, R., Le, D., and McCallum, A. (2020). ProtoQA: A question answering dataset for prototypical common-sense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1122–1136, Online. Association for Computational Linguistics.

Bordes, A., Chopra, S., and Weston, J. (2014). Question Answering with Subgraph Embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 615–620.

Bordes, A., Usunier, N., García-Durán, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *NIPS*.

Borgatti, S. P. (2005). Centrality and network flow. *Soc. Networks*, 27:55–71.

Bosselut, A., Levy, O., Holtzman, A., Ennis, C., Fox, D., and Choi, Y. (2017). Simulating action dynamics with neural process networks. *CoRR*, abs/1711.05313.

Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., and Choi, Y. (2019). COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Brahman, F. and Chaturvedi, S. (2020). Modeling protagonist emotions for emotion-aware storytelling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5277–5294, Online. Association for Computational Linguistics.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Cabrio, E. and Villata, S. (2012). Natural language arguments: A combined approach. In *ECAI*, pages 205–210.

Cambria, E., Li, Y., Xing, F. Z., Poria, S., and Kwok, K. (2020). *SenticNet 6: Ensemble Application of Symbolic and Subsymbolic AI for Sentiment Analysis*, page 105–114. Association for Computing Machinery, New York, NY, USA.

Celikyilmaz, A., Clark, E., and Gao, J. (2020). Evaluation of text generation: A survey. *ArXiv*, abs/2006.14799.

Chambers, N. and Jurafsky, D. (2008a). Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 698–706, Honolulu, Hawaii. Association for Computational Linguistics.

Chambers, N. and Jurafsky, D. (2008b). Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797, Columbus, Ohio. Association for Computational Linguistics.

Chambers, N. and Jurafsky, D. (2009). Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610, Suntec, Singapore. Association for Computational Linguistics.

Chambers, N. and Jurafsky, D. (2010). A database of narrative schemas. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Chambers, N., Wang, S., and Jurafsky, D. (2007). Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 173–176, Prague, Czech Republic. Association for Computational Linguistics.

Chater, N., Oaksford, M., Hahn, U., and Heit, E. (2010). Bayesian models of cognition. *Wiley Interdisciplinary Reviews: Cognitive Science*, 1(6):811–823.

Chaturvedi, S., Goldwasser, D., and Daumé, H. (2016). Ask, and shall you receive? understanding desire fulfillment in natural language text. In *AAAI*.

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

Chen, S., Hou, Y., Cui, Y., Che, W., Liu, T., and Yu, X. (2020). Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7870–7881, Online. Association for Computational Linguistics.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B., and Sen, P. (2020). A survey of the state of explainable AI for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459, Suzhou, China. Association for Computational Linguistics.

Davis, E. and Marcus, G. (2015). Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103.

Davison, J., Feldman, J., and Rush, A. (2019). Commonsense knowledge mining from pretrained models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1173–1178, Hong Kong, China. Association for Computational Linguistics.

de Morgan, A. (2002). Formal logic: Or, the calculus of inference, necessary and probable.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ding, H. and Riloff, E. (2018). Human needs categorization of affective events using labeled and unlabeled data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1919–1929.

Ding, S., Xu, H., and Koehn, P. (2019). Saliency-driven word alignment interpretation for neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 1–12, Florence, Italy. Association for Computational Linguistics.

Douven, I. (2017). Abduction. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2017 edition.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14:179–211.

Epstude, K. and Roese, N. J. (2008). The functional theory of counterfactual thinking. *Personality and social psychology review*, 12(2):168–192.

Fan, A., Lewis, M., and Dauphin, Y. (2018). Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Freund, Y. and Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Machine learning*, 37(3):277–296.

Garcez, A. and Lamb, L. (2020). Neurosymbolic ai: The 3rd wave. *ArXiv*, abs/2012.05876.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N. F., Peters, M., Schmitz, M., and Zettlemoyer, L. (2018). AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Ghosal, D., Majumder, N., Mihalcea, R., and Poria, S. (2021). STaCK: Sentence ordering with temporal commonsense knowledge. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8676–8686, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Goldberg, A. B., Fillmore, N., Andrzejewski, D., Xu, Z., Gibson, B., and Zhu, X. (2009). May all your wishes come true: A study of wishes and how to recognize them. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 263–271.

Goldberg, Y. (2016). A primer on neural network models for natural language processing. *J. Artif. Int. Res.*, 57(1):345–420.

Gordon, A. (2016a). Commonsense interpretation of triangle behavior. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

Gordon, A. S. (2016b). Commonsense interpretation of triangle behavior. In *AAAI*.

Gordon, A. S. (2019). The theory of mind in strategy representations. *Proceedings of the Twenty-Fourth Annual Conference of the Cognitive Science Society*.

Gordon, J. and Van Durme, B. (2013). Reporting bias and knowledge acquisition. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, AKBC '13, page 25–30, New York, NY, USA. Association for Computing Machinery.

Goyal, A., Riloff, E., and Daumé, H. (2013). A computational model for plot units. *Computational Intelligence*, 29.

Graham, S., Depp, C., Lee, E. E., Nebeker, C., Tu, X., Kim, H. C., and Jeste, D. V. (2019). Artificial intelligence for mental health and mental illnesses: an overview. *Current psychiatry reports*.

Grandy, R. E. and Warner, R. (2020). Paul Grice. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2020 edition.

Guan, J., Huang, F., Zhao, Z., Zhu, X., and Huang, M. (2020). A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, 8:93–108.

Guan, J., Wang, Y., and Huang, M. (2019). Story ending generation with incremental encoding and commonsense knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6473–6480.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5).

Gunning, D. (2018). Machine common sense concept paper. *ArXiv*, abs/1810.07528.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., and Smith, N. A. (2018). Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112.

Hamilton, W. L., Clark, K., Leskovec, J., and Jurafsky, D. (2016). Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 595. NIH Public Access.

Han, R., Ning, Q., and Peng, N. (2019). Joint event and temporal relation extraction with shared representations and structured prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 434–444, Hong Kong, China. Association for Computational Linguistics.

Haveliwala, T. H. (2002). Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM.

Hayes, A. and Krippendorff, K. (2007). Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures*, 1:77 – 89.

Henaff, M., Weston, J., Szlam, A., Bordes, A., and LeCun, Y. (2016). Tracking the world state with recurrent entity networks. *CoRR*, abs/1612.03969.

Higgins, E. T. (1996). Knowledge activation: Activation: Accessibility, and salience. *Social psychology: Handbook of basic principles*, pages 133–168.

Hobbs, J. R. (1985). On the coherence and structure of discourse.

Hobbs, J. R., Stickel, M. E., Appelt, D. E., and Martin, P. (1993a). Interpretation as abduction. *Artificial Intelligence*, 63(1):69–142.

Hobbs, J. R., Stickel, M. E., Appelt, D. E., and Martin, P. A. (1993b). Interpretation as abduction. *Artif. Intell.*, 63:69–142.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Huang, L., Le Bras, R., Bhagavatula, C., and Choi, Y. (2019). Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Hwang, J. D., Bhagavatula, C., Bras, R. L., Da, J., Sakaguchi, K., Bosselut, A., and Choi, Y. (2021). Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. In *AAAI*.

Ilievski, F., Oltramari, A., Ma, K., Zhang, B., McGuinness, D. L., and Szekely, P. A. (2021). Dimensions of commonsense knowledge. *Knowledge-Based Systems*, 229:107347.

Jain, S. and Wallace, B. C. (2019). Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Ji, H., Ke, P., Huang, S., Wei, F., Zhu, X., and Huang, M. (2020). Language generation with multi-hop reasoning on commonsense knowledge graph. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 725–736, Online. Association for Computational Linguistics.

Ji, S., Pan, S., Cambria, E., Marttinen, P., and Yu, P. S. (2021). A survey on knowledge graphs: Representation, acquisition and applications. *IEEE transactions on neural networks and learning systems*, PP.

Kahneman, D. (2003). Maps of bounded rationality: Psychology for behavioral economics. *The American Economic Review*, 93(5):1449–1475.

Kakas, A. C., Kowalski, R. A., and Toni, F. (1992). Abductive logic programming. *Journal of logic and computation*, 2(6):719–770.

Kavumba, P., Inoue, N., Heinzerling, B., Singh, K., Reisert, P., and Inui, K. (2019). When choosing plausible alternatives, clever hans can be clever. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 33–42, Hong Kong, China. Association for Computational Linguistics.

Kiddon, C., Zettlemoyer, L., and Choi, Y. (2016). Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 329–339, Austin, Texas. Association for Computational Linguistics.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751.

Kingma, D. P. and Ba, J. L. (2014). Adam: Amethod for stochastic optimization. In *Proc. 3rd Int. Conf. Learn. Representations*.

Kintsch, W. and Dijk, T. A. (1978). Toward a model of text comprehension and production. *Psychological Review*, 85:363–394.

Kobbe, J., Opitz, J., Becker, M., Hulpus, I., Stuckenschmidt, H., and Frank, A. (2019). Exploiting background knowledge for argumentative relation classification. In *LDK*, pages 1–8.

Koons, R. (2021). Defeasible Reasoning. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2021 edition.

Kuipers, T. A. (1992). Naive and refined truth approximation. *Synthese*, 93(3):299–341.

Kuipers, T. A. (2013). *From instrumentalism to constructive realism: On some relations between confirmation, empirical progress, and truth approximation*, volume 287. Springer Science & Business Media.

Lakoff, G. (2004). Linguistics and natural logic. *Synthese*, 22:151–271.

Lan, Y., Wang, S., and Jiang, J. (2019). Knowledge base question answering with topic units. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5046–5052. International Joint Conferences on Artificial Intelligence Organization.

Lehnert, W. G. (1981). Plot units and narrative summarization. *Cognitive Science*, 5(4):293–331.

Lenat, D. B. and Guha, R. V. (1989). *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc., USA, 1st edition.

Levesque, H., Davis, E., and Morgenstern, L. (2012). The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Li, A., Wang, X., Wang, W., Zhang, A., and Li, B. (2019a). A survey of relation extraction of knowledge graphs. In *APWeb/WAIM Workshops*.

Li, J., Chen, X., Hovy, E., and Jurafsky, D. (2016a). Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691, San Diego, California. Association for Computational Linguistics.

Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, W. B. (2016b). A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Li, J. and Hovy, E. (2017). Reflections on sentiment/opinion analysis. In *A Practical Guide to Sentiment Analysis*, pages 41–59. Springer.

Li, J., Song, Y., Zhang, H., Chen, D., Shi, S., Zhao, D., and Yan, R. (2018). Generating classical Chinese poems via conditional variational autoencoder and adversarial training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3890–3900, Brussels, Belgium. Association for Computational Linguistics.

Li, Q., Wang, B., and Melucci, M. (2019b). CNM: An interpretable complex-valued network for matching. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4139–4148, Minneapolis, Minnesota. Association for Computational Linguistics.

Lin, B. Y., Chen, X., Chen, J., and Ren, X. (2019). KagNet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2829–2839, Hong Kong, China. Association for Computational Linguistics.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Lin, S.-T., Chambers, N., and Durrett, G. (2021). Conditional generation of temporally-ordered event sequences. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7142–7157, Online. Association for Computational Linguistics.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. (2018). Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*.

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

Liu, Y., Wan, Y., He, L., Peng, H., and Yu, P. S. (2021). Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. In *AAAI*.

Luo, L., Ao, X., Pan, F., Wang, J., Zhao, T., Yu, N., and He, Q. (2018). Beyond polarity: Interpretable financial sentiment analysis with hierarchical query-driven attention. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4244–4250. International Joint Conferences on Artificial Intelligence Organization.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Ma, Y., Peng, H., and Cambria, E. (2018). Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. In *AAAI*.

Madaan, A. and Yang, Y. (2021). Neural language modeling for contextualized temporal graph generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 864–881, Online. Association for Computational Linguistics.

Mallozzi, A., Vaidya, A., and Wallner, M. (2021). The Epistemology of Modality. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2021 edition.

Mani, I., Verhagen, M., Wellner, B., Lee, C. M., and Pustejovsky, J. (2006). Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 753–760, Sydney, Australia. Association for Computational Linguistics.

Manning, C. D. and Schütze, H. (2002). Foundations of statistical natural language processing. In *SGMD*.

Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., and Wu, J. (2019). The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. In *International Conference on Learning Representations*.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Maslow, A. H. (1943). A theory of human motivation. *Psychological review*, 50(4):370.

Mausam (2016). Open information extraction systems and downstream applications. In *IJCAI*.

Mausam, Schmitz, M., Soderland, S., Bart, R., and Etzioni, O. (2012). Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea. Association for Computational Linguistics.

McCarthy, J. (1960). Programs with common sense. RLE and MIT computation center.

McDowell, B., Chambers, N., Ororbia II, A., and Reitter, D. (2017). Event ordering with a generalized model for sieve prediction ranking. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 843–853, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Menzel, C. (2021a). Actualism. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2021 edition.

Menzel, C. (2021b). Possible Worlds. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2021 edition.

Menzies, T. (1996). Applications of abduction: knowledge-level modelling. *Int. J. Hum. Comput. Stud.*, 45:305–335.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. (2018). Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Mihaylov, T. and Frank, A. (2018a). Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832, Melbourne, Australia. Association for Computational Linguistics.

Mihaylov, T. and Frank, A. (2018b). Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832.

Mikolov, T., Chen, K., Corrado, G. S., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *ICLR*.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Min, S., Wallace, E., Singh, S., Gardner, M., Hajishirzi, H., and Zettlemoyer, L. (2019). Compositional questions do not necessitate multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4249–4257, Florence, Italy. Association for Computational Linguistics.

Minsky, M. (2000). Commonsense-based interfaces. *Commun. ACM*, 43(8):66–73.

Mitra, A., Banerjee, P., Pal, K. K., Mishra, S., and Baral, C. (2019). Exploring ways to incorporate additional knowledge to improve natural language commonsense question answering. *arXiv preprint arXiv:1909.08855*.

Moore, C. (2013). *The development of commonsense psychology*. Psychology Press.

Moradi, M. and Samwald, M. (2021). Evaluating the robustness of neural language models to input perturbations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1558–1570, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mosbach, M., Andriushchenko, M., and Klakow, D. (2021). On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*.

Mostafazadeh, N., Chambers, N., He, X., Parikh, D., Batra, D., Vanderwende, L., Kohli, P., and Allen, J. (2016). A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Mostafazadeh, N., Kalyanpur, A., Moon, L., Buchanan, D., Berkowitz, L., Biran, O., and Chu-Carroll, J. (2020a). GLUCOSE: GeneraLized and COntextualized story explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4569–4586, Online. Association for Computational Linguistics.

Mostafazadeh, N., Kalyanpur, A., Moon, L., Buchanan, D., Berkowitz, L., Biran, O., and Chu-Carroll, J. (2020b). GLUCOSE: GeneraLized and COntextualized story explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4569–4586, Online. Association for Computational Linguistics.

Mudrakarta, P. K., Taly, A., Sundararajan, M., and Dhamdhere, K. (2018). Did the model understand the question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1896–1906, Melbourne, Australia. Association for Computational Linguistics.

Neal, R. M. (2007). Pattern recognition and machine learning. *Technometrics*, 49:366 – 366.

Nguyen, K.-H., Tannier, X., Ferret, O., and Besançon, R. (2015). Generative event schema induction with entity disambiguation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 188–197.

Opitz, J. and Frank, A. (2019). Dissecting content and context in argumentative relation analysis. In *Proceedings of the 6th Workshop on Argument Mining*, pages 25–34, Florence, Italy. Association for Computational Linguistics.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Paul, D. and Frank, A. (2019). Ranking and selecting multi-hop knowledge paths to better predict human needs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3671–3681, Minneapolis, Minnesota. Association for Computational Linguistics.

Paul, D. and Frank, A. (2020). Social commonsense reasoning with multi-head knowledge attention. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2969–2980, Online. Association for Computational Linguistics.

Paul, D. and Frank, A. (2021a). COINS: Dynamically generating COntextualized inference rules for narrative story completion. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5086–5099, Online. Association for Computational Linguistics.

Paul, D. and Frank, A. (2021b). Generating hypothetical events for abductive inference. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 67–77, Online. Association for Computational Linguistics.

Paul, D., Opitz, J., Becker, M., Kobbe, J., Hirst, G., and Frank, A. (2020). Argumentative Relation Classification with Background Knowledge. In *Proceedings of the 8th International Conference on Computational Models of Argument (COMMA 2020)*, volume 326 of *Frontiers in Artificial Intelligence and Applications*, pages 319–330. Computational Models of Argument.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., CA.

Peirce, C. S. (1883). A Theory of Probable Inference. In Peirce, C. S., editor, *Studies in Logic by Members of the Johns Hopkins University*, pages 126–181. Little, Brown, and Company, Boston, MA. From the Commens Bibliography | http://www.commens.org/bibliography/collection_article/peirce-charles-s-1883-theory-probable-inference-studies-logic.

Peirce, C. S. (1903). *Pragmatism as the Logic of Abduction*. https://www.textlog.de/7663.html.

Peirce, C. S. (1965a). *Collected papers of Charles Sanders Peirce*, volume 5. Harvard University Press. http://www.hup.harvard.edu/catalog.php?isbn=9780674138001.

Peirce, C. S. (1965b). *Pragmatism and pragmaticism*, volume 5. Belknap Press of Harvard University Press. https://www.jstor.org/stable/224970.

Peng, N., Ghazvininejad, M., May, J., and Knight, K. (2018). Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*, pages 43–49, New Orleans, Louisiana. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Pereira, G., Prada, R., and Santos, P. A. (2016). Integrating social power into the decision-making of cognitive agents. *Artificial Intelligence*, 241:1–44.

Peters, J., Bühlmann, P., and Meinshausen, N. (2016). Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *NAACL*, pages 2227–2237.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., and Miller, A. (2019). Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Pichotta, K. and Mooney, R. (2014). Statistical script learning with multi-argument events. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 220–229.

Pichotta, K. and Mooney, R. J. (2016). Using sentence-level LSTM language models for script inference. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 279–289, Berlin, Germany. Association for Computational Linguistics.

Plutchik, R. (1980). A general psychoevolutionary theory of emotion. *Theories of emotion*, 1(4):3–31.

Pollack, M. E. (2005). Intelligent technology for an aging population: The use of ai to assist elders with cognitive impairment. *AI Magazine*, 26(2):9.

Pople, H. E. (1973). On the mechanization of abductive logic. In *Proceedings of the 3rd international joint conference on Artificial intelligence*, pages 147–152.

Pradhan, S., Hacioglu, K., Krugler, V., Ward, W., Martin, J. H., and Jurafsky, D. (2005). Support vector learning for semantic argument classification. *Machine Learning*, 60(1–3):11–39.

Premack, D. and Woodruff, G. (1978). Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4):515–526.

Qin, L., Bosselut, A., Holtzman, A., Bhagavatula, C., Clark, E., and Choi, Y. (2019). Counterfactual story reasoning and generation. In *2019 Conference on Empirical Methods in Natural Language Processing.*, Hongkong, China. Association for Computational Linguistics.

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

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.

Rahimtoroghi, E., Wu, J., Wang, R., Anand, P., and Walker, M. (2017). Modelling protagonist goals and desires in first-person narrative. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 360–369.

Raina, R., Ng, A., and Manning, C. D. (2005). Robust textual inference via learning and abductive reasoning. In *AAAI*.

Rajendran, P., Bollegala, D., and Parsons, S. (2016). Contextual stance classification of opinions: A step towards enthymeme reconstruction in online reviews. In *Workshop on Argument Mining*, pages 31–39.

Rashkin, H. (2020). *Commonsense reasoning about social dynamics in text*. University of Washington.

Rashkin, H., Bosselut, A., Sap, M., Knight, K., and Choi, Y. (2018a). Modeling naive psychology of characters in simple commonsense stories. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2289–2299, Melbourne, Australia. Association for Computational Linguistics.

Rashkin, H., Bosselut, A., Sap, M., Knight, K., and Choi, Y. (2018b). Modeling naive psychology of characters in simple commonsense stories. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2289–2299.

Rashkin, H., Sap, M., Allaway, E., Smith, N. A., and Choi, Y. (2018c). Event2Mind: Commonsense inference on events, intents, and reactions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 463–473, Melbourne, Australia. Association for Computational Linguistics.

Rei, M. and Søgaard, A. (2018). Zero-shot sequence labeling: Transferring knowledge from sentences to tokens. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 293–302.

Reisert, P., Inoue, N., Okazaki, N., and Inui, K. (2017). Deep argumentative structure analysis as an explanation to argumentative relations. In *ACL*, pages 38–41.

Reiss, S. (2002). *Who am I?: 16 basic desires that motivate our actions define our persona*. Penguin.

Reiss, S. (2004). Multifaceted nature of intrinsic motivation: The theory of 16 basic desires. *Review of general psychology*, 8(3):179.

Ribeiro, M. T., Wu, T., Guestrin, C., and Singh, S. (2020). Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Riedel, S., Yao, L., McCallum, A., and Marlin, B. M. (2013). Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84, Atlanta, Georgia. Association for Computational Linguistics.

Roemmele, M., Bejan, C. A., and Gordon, A. S. (2011). Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Roese, N. J. and Morrison, M. (2009). The psychology of counterfactual thinking. *Historical Social Research/Historische Sozialforschung*, pages 16–26.

Rothe, S., Narayan, S., and Severyn, A. (2020). Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Ruder, S. (2019). *Neural transfer learning for natural language processing*. PhD thesis, NUI Galway.

Ruder, S., Peters, M. E., Swayamdipta, S., and Wolf, T. (2019). Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota. Association for Computational Linguistics.

Sakaguchi, K., Le Bras, R., Bhagavatula, C., and Choi, Y. (2020). Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8732–8740.

Sap, M., Bras, R. L., Allaway, E., Bhagavatula, C., Lourie, N., Rashkin, H., Roof, B., Smith, N. A., and Choi, Y. (2019a). ATOMIC: an atlas of machine commonsense for if-then reasoning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019.*, pages 3027–3035.

Sap, M., Gabriel, S., Qin, L., Jurafsky, D., Smith, N. A., and Choi, Y. (2020a). Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490, Online. Association for Computational Linguistics.

Sap, M., Rashkin, H., Chen, D., Le Bras, R., and Choi, Y. (2019b). Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Sap, M., Shwartz, V., Bosselut, A., Choi, Y., and Roth, D. (2020b). Commonsense reasoning for natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 27–33, Online. Association for Computational Linguistics.

Schank, R. C. and Abelson, R. P. (1977). *Scripts, plans, goals, and understanding : an inquiry into human knowledge structures*. Hillsdale, N.J. : Lawrence Erlbaum Associates.

Selman, B. and Levesque, H. J. (1990). Abductive and default reasoning: A computational core. In *AAAI*.

Serrano, S. and Smith, N. A. (2019). Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Shi, P. and Lin, J. (2019). Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer.

Singh, P., Lin, T., Mueller, E., Lim, G., Perkins, T., and Zhu, W. L. (2002). Open mind common sense: Knowledge acquisition from the general public. In *OTM*.

Singla, P. and Mooney, R. J. (2011). Abductive markov logic for plan recognition. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Smith, R. (2020). Aristotle's Logic. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2020 edition.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Speer, R., Chin, J., and Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Speer, R. and Havasi, C. (2012). Representing general relational knowledge in ConceptNet 5. In *LREC*, pages 3679–3686.

Stab, C. and Gurevych, I. (2014a). Annotating argument components and relations in persuasive essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Stab, C. and Gurevych, I. (2014b). Identifying argumentative discourse structures in persuasive essays. In *EMNLP*, pages 46–56.

Stab, C. and Gurevych, I. (2017). Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Starr, W. (2021). Counterfactuals. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2021 edition.

Storks, S., Gao, Q., and Chai, J. (2019). Recent advances in natural language inference: A survey of benchmarks, resources, and approaches. *arXiv: Computation and Language*.

Strasser, C. and Antonelli, G. A. (2019). Non-monotonic logic. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2019 edition.

Sun, C., Qiu, X., Xu, Y., and Huang, X. (2019). How to fine-tune bert for text classification? In *CCL*.

Talmor, A., Herzig, J., Lourie, N., and Berant, J. (2019). CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Tamilselvam, S., Nagar, S., Mishra, A., and Dey, K. (2017). Graph Based Sentiment Aggregation using ConceptNet Ontology. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 525–535.

Tandon, N., de Melo, G., and Weikum, G. (2017). WebChild 2.0 : Fine-grained commonsense knowledge distillation. In *Proceedings of ACL 2017, System Demonstrations*, pages 115–120, Vancouver, Canada. Association for Computational Linguistics.

Trichelair, P., Emami, A., Trischler, A., Suleman, K., and Cheung, J. C. K. (2019). How reasonable are common-sense reasoning tasks: A case-study on the Winograd schema challenge and SWAG. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3382–3387, Hong Kong, China. Association for Computational Linguistics.

Valiant, L. (2008). Knowledge infusion: In pursuit of robustness in artificial intelligence. In *FSTTCS*.

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

Walton, D. (2015). *Goal-based Reasoning for Argumentation*. Cambridge University Press.

Weissenborn, D., Kočiskỳ, T., and Dyer, C. (2018). Dynamic integration of background knowledge in neural NLU systems.

Weston, J., Bordes, A., Chopra, S., and Mikolov, T. (2016). Towards ai-complete question answering: A set of prerequisite toy tasks. In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Wiegreffe, S. and Pinter, Y. (2019). Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.

Winograd, T. (1972). Understanding natural language. *Cognitive Psychology*, 3(1):1–191.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xiao, H., Huang, M., and Zhu, X. (2016). TransG : A generative model for knowledge graph embedding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2316–2325, Berlin, Germany. Association for Computational Linguistics.

Xie, Q., Ma, X., Dai, Z., and Hovy, E. (2017). An interpretable knowledge transfer model for knowledge base completion. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 950–962, Vancouver, Canada. Association for Computational Linguistics.

Xu, P., Patwary, M., Shoeybi, M., Puri, R., Fung, P., Anandkumar, A., and Catanzaro, B. (2020). MEGATRON-CNTRL: Controllable story generation with external knowledge using large-scale language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2831–2845, Online. Association for Computational Linguistics.

Xu, Z., Liu, B., Wang, B., Sun, C., and Wang, X. (2017). Incorporating loose-structured knowledge into conversation modeling via recall-gate LSTM. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 3506–3513. IEEE.

Yang, B. and Mitchell, T. (2017). Leveraging knowledge bases in LSTMs for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1436–1446, Vancouver, Canada. Association for Computational Linguistics.

Yang, B., Yih, W., He, X., Gao, J., and Deng, L. (2015). Embedding entities and relations for learning and inference in knowledge bases. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yang, Z., Blunsom, P., Dyer, C., and Ling, W. (2017). Reference-aware language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1850–1859, Copenhagen, Denmark. Association for Computational Linguistics.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *NAACL*, pages 1480–1489.

Yu, W., Yu, W., Zhu, C., Li, Z., Hu, Z., Wang, Q., Ji, H., and Jiang, M. (2022). A survey of knowledge-enhanced text generation. *ACM Computing Survey (CSUR)*, abs/2010.04389.

Yu, W., Zhou, J., Yu, W., Liang, X., and Xiao, N. (2019). Heterogeneous graph learning for visual commonsense reasoning. In *NeurIPS*.

Zadeh, L. (1975). The concept of a linguistic variable and its application to approximate reasoning—i. *Information Sciences*, 8(3):199–249.

Zhang, H., Liu, X., Pan, H., Song, Y., and Leung, C. W. (2020a). ASER: A large-scale eventuality knowledge graph. In *WWW*, pages 201–211.

Zhang, H., Liu, X., Pan, H., Song, Y., and Leung, C. W.-K. (2020b). Aser: A large-scale eventuality knowledge graph. In *Proceedings of The Web Conference 2020*, pages 201–211.

Zhang, L., Wang, S., and Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020c). BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

Zhou, B., Khashabi, D., Ning, Q., and Roth, D. (2019). "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3363–3369, Hong Kong, China. Association for Computational Linguistics.

Zhou, X., Zhang, Y., Cui, L., and Huang, D. (2020). Evaluating commonsense in pre-trained language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9733–9740.

Zhu, Y., Pang, L., Lan, Y., and Cheng, X. (2020). $l2r^2$: Leveraging ranking for abductive reasoning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1961–1964, New York, NY, USA. Association for Computing Machinery.