Constrained, Causal, and Knowledge-Grounded Reasoning for Neural Language Generation

Lianhui Qin

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington
2023

Reading Committee:
Yejin Choi, Chair
Luke Zettlemoyer
Fei Xia
Oren Etzioni

Program Authorized to Offer Degree:
Computer Science and Engineering

University of Washington

Abstract

Constrained, Causal, and Knowledge-Grounded Reasoning
for Neural Language Generation

Lianhui Qin

Chair of the Supervisory Committee:
Yejin Choi
Computer Science and Engineering

This thesis aims to establish a connection between reasoning and language generation. Today's language models (LMs, such as GPT-3), despite producing human-like fluent text, essentially act like "a mouth without a brain" – They generate without grounding on the world knowledge, and lack the ability to flexibly reason about everyday situations and events, including counterfactual ("*what if?*") and abductive ("*what might explain the observations?*") reasoning. This thesis bridges the gap from three angles: (1) Differentiable reasoning with constraints: Humans can incorporate any constraints from the context *on the fly* and conduct reasoning in new situations without the need of specific training. I develop a unified inference framework that endows the LMs with the flexibility and efficiency, through a differentiable process to reason over the vast space of discrete language, combined with arbitrary neural and symbolic constraints; (2) Counterfactual and nonmonotonic reasoning in natural language: I establish the first formulation of counterfactual reasoning in language, and used my inference tool to enable the common *monotonic* LMs for the capabilities of *nonmonotonic* reasoning ranging from counterfactual, abductive, and temporal reasoning in complex context; (3) Integration of knowledge and logic in neural language models: I develop mechanisms of integrating rich external knowledge and structures with the neural LMs, to ground and boost the reasoning abilities.

Acknowledgements

I am deeply grateful for the remarkable support I have received throughout my challenging yet rewarding PhD journey. This journey has been filled with a great deal of learning, occasional struggles, and significant growth. Without the support from my advisors, colleagues, collaborators, family, and friends, transforming this task into reality would have been near impossible.

First and foremost, my heartfelt gratitude goes to my advisor, Yejin Choi, for taking a chance on me by hiring me as her PhD student and for the mentorship thereafter. Her insightful guidance, faith in my abilities, and constant encouragement have significantly shaped my journey. Not only did she provide valuable research advice, help on develop skills like presentation and communication, but she also support me on all other aspects of my life and career. Her open-mindedness allowed me to explore my research interests freely and her feedback was crucial in shaping my research tastes. When I stumbled, she was always there, cheering me on, brainstorming for my projects, and linking me to helpful resources and collaborators. When life presented difficulties including health and family issues, it was Yejin who provided unconditional understanding and flexibility. I cannot imagine completing this degree without her.

I would also like to express my sincere gratitude to other faculty members who were crucial in my job search. My committee members, Luke Zettlemoyer, Fei Xia, and Oren Etzioni, have been a great source of support. Luke's advice on my research and career, particularly during our collaborations at Meta, has been invaluable. The reference letters from Luke and Oren were crucial to my job search. Fei's feedback on my teaching talk was very helpful. In addition, I'd like to acknowledge a number of professors who provided feedback on my practice job talk that led to significant improvements in my slides, including Michael Ernst, Kurtis Heimerl, Anna Karlin, Sherry Wu, Hao Peng, Tianyi Zhou, and Danyang Zhuo.

In addition, I am grateful to have had the opportunity to complete four internships at Microsoft, AI2,

Google, and Meta during my PhD journey. My first industry mentor, Jianfeng Gao, deserves special mention for his patient guidance during my internship and the unwavering support afterwards. I am also thankful for the support from my other mentors and collaborators at these internships: Michel Galley, Chris Brockett, Xiaodong Liu, Xiang Gao, Bill Dolan at Microsoft, Chandra Bhagavatula, Jena Hwang, Ronan Le Bras at AI2, Manaal Faruqui, Aditya Gupta, Shyam Upadhyay, Luheng He at Google, and Asli Celikyilmaz, Spencer Poff, Khyathi Chandu, Marjan Ghazvininejad, and Chunting Zhou at Meta. My learning was significantly enhanced through these experiences. Lastly, I am especially grateful to the Microsoft fellowship for financially supporting my research.

I also want to extend my appreciation to my collaborators at UW and other institutions: Rowan Zellers, Sean Welleck, Daniel Khashabi, Ari Holtzman, Elizabeth Clark, Vered Shwartz, Peter West, Antoine Bosselut, Bowen Tan, Maarten Sap, Saadia Gabriel, Ximing Lu, Liwei Jiang, Prithviraj Ammanabrolu, Jaehun Jung, Faeze Brahman, Wenhao Yu, and so forth. The experiences we shared taught me how to collaborate effectively and conduct interesting research. I also want to thank my friends at UW who also offered invaluable feedback during my job search: Xingfan Huang, Sewon Min, Weijia Shi, Wenjun Wu, Yuanyuan Yang, Chenxingyu Zhao, and so on.

Lastly, I want to express my deepest gratitude to my family: My husband, Zhiting Hu, who has consistently been my rock-solid support; my parents, Zhichun Liang and Shouqiang Qin—thank you for your unwavering love and support; and my little girl, Lyra—your presence has brought immense joy and strength to this PhD adventure!

In closing, I am deeply grateful to everyone who has been part of my PhD journey. The shared moments of success, failure, and learning have made the past five years truly memorable and enriching.

DEDICATION

To Zhiting Hu and Lyra Hu

Contents

1 Introduction **23**

1.1 Overview of Dissertation 25

I Differentiable Reasoning with Constraints **29**

2 COLD Decoding: Energy-based Constrained Text Generation with Langevin Dynamics **31**

2.1 Introduction 32

2.2 Background 34

2.3 COLD Decoding with Langevin Dynamics 35

2.3.1 Energy-based Decoding 36

2.3.2 A Collection of COLD Constraints 37

2.3.3 From Soft to Discrete and Fluent Text 39

2.3.4 Implementation of COLD Decoding 40

2.4 Experiments 41

2.4.1 Abductive Reasoning 41

2.4.2 Counterfactual Story Rewriting 44

2.4.3 Lexically Constrained Decoding 45

2.4.4 Additional Analysis 47

2.5 Related Work 48

2.6 Conclusion 49

2.7 Appendix 49

2.7.1 Experimental Configurations . . . 49
2.7.2 Human Evaluation Details . . . 50
2.7.3 Ablation Study: Top-k Filtering . . . 52
2.7.4 Generated Samples . . . 52

II Counterfactual and Nonmonotonic Reasoning in Natural Language 55

3 TimeTravel: Counterfactual Story Reasoning and Generation 57

3.1 Introduction . . . 58
3.2 Background . . . 60
3.3 Counterfactual Story Rewriting . . . 61
3.3.1 Task . . . 62
3.3.2 Dataset: TIMETRAVEL . . . 62
3.3.3 Data Collection . . . 62
3.4 Learning a Counterfactual Rewriter . . . 63
3.4.1 Unsupervised Training . . . 64
3.4.2 Supervised Training (Sup) . . . 66
3.4.3 Hyperparameters . . . 67
3.5 Human Study of Rewritten Sentences . . . 67
3.5.1 Rewritten Sentence Scoring . . . 67
3.5.2 Pairwise Model Preference . . . 69
3.6 Challenges for Automatic Metrics . . . 69
3.6.1 Metrics . . . 70
3.6.2 Human Correlation with Metrics . . . 71
3.7 Conclusion . . . 71
3.8 Appendix . . . 72
3.8.1 Crowdsourcing Details . . . 72
3.8.2 Training Hyperparameters . . . 72

4 DeLorean: Unsupervised Backprop-based Decoding for Counterfactual and Abductive Commonsense Reasoning **75**

4.1 Introduction . . . 76
4.2 Background . . . 78
4.2.1 Abductive Reasoning . . . 78
4.2.2 Counterfactual Reasoning . . . 79
4.3 The DELOREAN Approach . . . 80
4.3.1 Decoding Strategy . . . 81
4.3.2 Ranking . . . 83
4.4 Task 1: Abductive Reasoning . . . 83
4.4.1 Task Setup . . . 84
4.4.2 Experimental Setup . . . 85
4.4.3 Results . . . 85
4.5 Task 2: Counterfactual Reasoning . . . 87
4.5.1 Task Setup . . . 87
4.5.2 Experimental Setup . . . 88
4.5.3 Results . . . 89
4.6 Related Work . . . 91
4.7 Conclusion . . . 92
4.8 Appendix . . . 92

III Integration of Knowledge and Logic in Neural Language Model Reasoning **95**

5 Conversing by Reading: Contentful Neural Conversation with On-demand Machine Reading **97**

5.1 Introduction . . . 98
5.2 Task . . . 99
5.3 Approach . . . 100
5.3.1 Document and Conversation Reading . . . 101
5.3.2 Response Generation . . . 101

5.3.3 Data Weighting Scheme . . . 102
5.4 Dataset . . . 102
5.5 Experiments . . . 104
5.5.1 Systems . . . 104
5.6 Experiment Details . . . 105
5.6.1 Evaluation Setup . . . 106
5.6.2 Automatic Evaluation . . . 107
5.6.3 Human Evaluation . . . 107
5.6.4 Qualitative Study . . . 108
5.7 Related Work . . . 110
5.8 Conclusions . . . 111

6 TimeDial: Temporal Commonsense Reasoning in Dialog 113

6.1 Introduction . . . 114
6.2 Task: Temporal Reasoning in Dialog . . . 115
6.3 Dataset: TIMEDIAL . . . 116
6.3.1 Data Collection . . . 116
6.3.2 Properties of TIMEDIAL . . . 118
6.4 Modeling . . . 119
6.4.1 Modeling Paradigms . . . 120
6.4.2 Dialog Context . . . 121
6.4.3 Training Details . . . 122
6.5 Experiments and Analyses . . . 123
6.5.1 Model Performance . . . 124
6.5.2 Error Analysis . . . 125
6.5.3 Influence of Dialog Context . . . 126
6.5.4 Errors of Reasoning Categories . . . 127
6.6 Related Work . . . 127
6.7 Conclusions . . . 128

6.8 Appendix 129

7 Conclusion and Future Work **137**

List of Figures

2.1 Applying COLD to different constrained generation tasks amounts to specifying an energy function E by plugging in relevant constraint functions. Text in grey boxes is the input, and text in blue boxes is the output. 33

2.2 An overview of the COLD decoding procedure. Given an energy function $E(\tilde{\mathbf{y}}) = \sum_i \lambda_i f_i(\tilde{\mathbf{y}})$ with various constraints, the procedure starts with a soft sequence $\tilde{\mathbf{y}}^{(0)}$ as a sample from an initial energy-based distribution, and performs Langevin dynamics iterations using the gradient $\nabla_{\tilde{\mathbf{y}}} E(\tilde{\mathbf{y}})$ (Eq.2.2). The resulting sequence $\tilde{\mathbf{y}}^{(N)}$ after N iterations is approximately a sample from the desired constrained distribution. We then apply top-k filtering on the soft sequence to produce a discrete text sequence $\mathbf{y}$ (Eq.2.6). 35

2.3 Illustrations of the differentiable constraints introduced in §2.3.2. **(1)** The soft fluency constraint (Eq.2.3) to encourage fluency of $\tilde{\mathbf{y}}_t$ based on LM probabilities. **(2)** The future contextualization constraint in Eq.(2.4) to encourage coherence w.r.t. the future context (`has eight legs`). **(3)** The n-gram similarity constraint in Eq.(2.5), where the left figure shows the case of $n = 1$ which encourages keywords (e.g., `hand`) to appear in the generation, and the right figure shows the case of $n > 1$ which is typically used to encourage sequence similarity with a reference text $\mathbf{y}_*$. 37

2.4 Screenshot of the mechanical turk interface used to gather human judgments for Lexically Constrained Generation. 50

2.5 Screenshot of the mechanical turk interface used to gather human judgments for Abductive Reasoning. 51

2.6 Screenshot of the mechanical turk interface used to gather human judgments for Counterfactual Reasoning. 51

3.1 Given a short story (left column) and a *counterfactual context* ("He decided to be a werewolf this year"), the task is to revise the original story with minimal edits to be consistent with both the original premise ("Pierre loved Halloween") and the new counterfactual situation. The modified parts in the new story (right column) are highlighted in red. 58

3.2 Data annotation process for the TIMETRAVEL dataset. Given a story from the ROCStories corpus, crowdworkers write a counterfactual sentence w.r.t the second sentence of the story. The counterfactual sentence and the original story are then presented to other workers to rewrite the story ending. Models for the task are expected to generate a rewritten ending given the original story and counterfactual sentence. 59

4.1 DELOREAN, our proposed method, with generated reasoning results. **Top**: the goal in abductive reasoning is to generate a hypothesis (Y) of what happened between the observed past (X) and future (Z) contexts. **Bottom**: In counterfactual reasoning, given a story context altered by a counterfactual condition, X, and the original ending Z, the goal is to generate a new ending Y which is coherent with X while remaining similar to Z. The story from TIMETRAVEL [Qin et al., 2019a] consists of five sentences. Our approach alternates forward (left-to-right) and backward (right-to-left) passes that iteratively refine the generated texts w.r.t context from each side. 76

4.2 Illustration of the DELOREAN decoding procedure, using abductive reasoning as an example. At initialization (upper-left box), the language model (LM) initializes the logits $\tilde{Y} = \{\tilde{y}_1, \ldots, \tilde{y}_N\}$ of the hypothesis by reading the past context X and generating a continuation with regular decoding. At each forward-backward iteration, we compute the task-specific **loss** $\mathcal{L}_{\tilde{Y}}$ of the logits based on the future constraint Z (red box). The **backward pass** then performs back-propagation and produces the backward logits $\tilde{Y}^b = \{\tilde{y}_1^b, \ldots, \tilde{y}_N^b\}$. In the subsequent **forward pass**, for each step n, we compute the forward logits $\tilde{y}_n^f$ conditioning on the preceding logits $\tilde{y}_{n-1}$, and then mix it with the respective backward logits to produce the new logits $\tilde{y}_n$ at step n. 80

4.3 Examples of generated hypotheses on three abductive reasoning cases. Given observations O1 and O2, DELOREAN generates a hypothesis explaining the observations. 83

4.4 Human calibration results for counterfactual generation in terms of weighted harmonic mean of coherence and min-edit, $H_\beta = \frac{(1+\beta^2)\cdot\text{coherence}\cdot\text{min\_edit}}{\beta^2\cdot\text{coherence}+\text{min\_edit}}$, as a function of the scaling factor β. Low β values assign more weight to coherence, and high β values emphasize more on min-edit. 88

4.5 Examples of generated story endings on three counterfactual reasoning cases. Given a story context, a counterfactual condition, and a original ending, DELOREAN generates a rewritten ending which is coherent with the counterfactual condition and is similar to the original ending. 90

5.1 Users discussing a topic defined by a Wikipedia article. In this real-world example from our Reddit dataset, information needed to ground responses is distributed throughout the source document. 98

5.2 **Model Architecture for Response Generation with on-demand Machine Reading:** The first blocks of the MRC-based encoder serve as a lexicon encoding that maps words to their embeddings and transforms with position-wise FFN, independently for the conversation history and the document. The next block is for contextual encoding, where BiLSTMs are applied to the lexicon embeddings to model the context for both conversation history and document. The last block builds the final encoder memory, by sequentially applying cross-attention in order to integrate the two information sources, conversation history and document, self-attention for salient information retrieval, and a BiLSTM for final information rearrangement. The response generator then attends to the memory and generates a free-form response. 100

5.3 Attention weights between words of the documents and words of the response. Dark (blue) cells represent probabilities closer to 1. 108

6.1 We study three modeling paradigms for the task, based on BERT and T5, including (1) Classification, (2) Mask Filling, and (3) Generation (§6.4.1). The models are finetuned with various training data, as discussed in §6.4.3. 120

6.2 Percentage of errors on options created by different rules. CLS, MF, and GEN represent classification, mask-filling, and generation models, respectively; and IN and OUT denote in-domain and out-of-domain training. All models are of large size. 125

6.3 Percentage of errors on different reasoning types. CLS, MF, and GEN represent classification, mask-filling, and generation models, respectively. All models are of large size. 127

List of Tables

2.1 Automatic and human evaluation of abductive reasoning (2.4.1). Our proposed method (COLD decoding) outperforms DELOREAN, a recent decoding algorithm achieving strong results in this task. 41

2.2 Automatic and human evaluation of counterfactual story rewriting. As a trivial method, LEFT-ONLY is coherent but fails on minimal-edit. COLD is superior to DELOREAN in terms of most metrics, including human evaluation. 43

2.3 Results of lexically constrained decoding (§2.4.3). For keyword coverage, we report both the average number and average percentage of constraint words present in the generated text. For language fluency, we use perplexity and human judgement. 43

2.4 Ablation for the effect of different constraints in Eq.(2.7). We do human evaluation on 125 test examples. The best overall coherence is achieved when all the constraints are present. . 47

2.5 COLD is more efficient than gradient-free Mix-and-Match [Mireshghallah et al., 2022]. The runtime shown is seconds per sample on Counterfactual Story Rewriting. 47

2.6 Ablation for the effect of k in top-k filtering mechanism (§2.3.3). We use the same setting as Table 2.5. 52

2.7 Examples for abductive reasoning. 53

2.8 Examples for counterfactual reasoning. 53

2.9 Examples for lexically constrained generation. 53

3.1 Examples from TIMETRAVEL 61

3.2 Dataset statistics 63

3.3 Model Outputs 66

3.4 Likert scale scores for different models. The top performing model for each question is **bolded**. 68

3.5 Pairwise human comparison between the best model (GPT2-M + Sup) and comparison models on all three questions. "Neutral" means both are "equally good". Percentage of "equally bad" are omitted. 73

3.6 Pearson correlation between automatic metrics and human scores. **Bolded** numbers are statistically significant at $p < 0.05$. 73

3.7 Results on automatic metrics for the cross-product of the models and loss functions proposed in Section 5.3. **Bolded** results are closest to the human score. 74

4.1 Automatic evaluation results on the abductive task, using the test set of $\mathcal{ART}$. 84

4.2 Human calibration results on test set of $\mathcal{ART}$. All scores are normalized to $[0, 1]$. 86

4.3 Human pairwise comparison results on the test set of $\mathcal{ART}$, between COLD and each of the baselines, by jointly considering all 3 criteria from Table 4.2. "Neutral" means "equally good/bad". 86

4.4 Automatic evaluation results of counterfactual story rewriting, on the test set of TIMETRAVEL. 87

4.5 Human pairwise comparison results on the counterfactual task, between our best model and each baseline with respect to coherence and min-edits. 89

5.1 Our grounded conversational dataset. 103

5.2 **Automatic Evaluation** results (higher is better for all metrics). Our best models (CMR+W and CMR) considerably increase the quantitative measures of Grounding, and also slightly improve Diversity. Automatic measures of Quality (e.g., BLEU-4) give mixed results, but this is reflective of the fact that we did not aim to improve response relevance with respect to the context, but instead its level of grounding. The human evaluation results in Table 5.3 indeed suggest that our best system (CMR+W) is better. 105

5.3 **Human Evaluation** results, showing preferences (%) for our model (CMR+W) vs. baseline and other comparison systems. Distributions are skewed towards CMR+W. The 5-point Likert scale has been collapsed to a 3-point scale. *Differences in mean preferences are statistically significant ($p \leq 0.0001$). 107

5.4 Sample output comparing our best system (CMR+W) against Memory Networks and a SEQ2SEQ baseline. The source documents were manually shortened to fit in the table, without significantly affecting meaning. 109

6.1 Examples from our TIMEDIAL challenge set, demonstrating the need for commonsense knowledge and arithmetic reasoning over the context to infer the correct answers. Key contextual information for reasoning success is highlighted. 114

6.2 Example dialogs and answer options from the TIMEDIAL dataset, categorized by the nature of reasoning required to correctly answer them, along with the percentage of each reasoning category in the set of 100 sampled examples. The relevant key information in the dialog context is highlighted. 117

6.3 Statistics of our TIMEDIAL challenge set. 119

6.4 Number of training and development instances for different settings. An instance is derived by masking one temporal span of a dialog. For classification, we draw 3 negative samples per positive sample. "# Spans" is the size of temporal span pool from which negative samples are drawn for weak supervision. 122

6.5 Model and human performance on TIMEDIAL. BASE and LARGE denote the size of the pre-trained BERT and T5; ZERO, IN, and OUT denote that the model is zero-shot (with no finetuning), fintuned using the in-domain DailyDialog data, or finetuned using the out-of-domain Meena data, respectively. THe full dialog context is used for all models. 123

6.6 Example prediction errors made by different models for cases with challenging options, based on the phrase and numeral matching rules (§6.3). GOLD denotes the true labels. The model predictions show that the models get confused by learning shallow text matching in terms of pre-existing temporal concepts (marked by bold faced text) in the context. 124

6.7 Impact of dialog context on reasoning accuracy. IN and OUT denote in-domain and out-of-domain training, respectively. We use *2-best accuracy* of *target* context as reference and report the absolute changes in performance of *local* and *full* context, respectively. Local dialog context results in better performance to full dialog context on 5 of the 12 cases, which are highlighted in the table. 126

Chapter 1

Introduction

Speaking of reasoning for neural language generation, we can naturally think of recent large language models, such as chatGPT. These models have demonstrated remarkable abilities in generating fluent natural language like humans. In fact, they are getting widely deployed in our daily lives. For example, chatGPT reached 100 million users within two months after its release in December 2022, and has even been considered for use in finance, healthcare, judicial systems, and others. These seem really exciting! However, are these large language models good reasoners? Can they do robust reasoning like humans? And are they ready to be deployed in critical domains like health and law?

Unfortunately, those models can fail surprisingly easily. For example, when presented with a simple question such as, "*It takes 3 hours for 2 cars driving from Seattle to Portland. How long would it take for 4 cars?*", chatGPT answered correctly: "*it would take 3 hours.*"; however, when the question was slightly modified by adding the phrase "*explain briefly*" as in, "*It takes 3 hours for 2 cars driving from Seattle to Portland. Briefly explain how long it would take for 4 cars.*" ChatGPT suddenly gave an incorrect answer: "*6 hours.*" Though the systems would continue to improve by training on more data with more model parameters (e.g., GPT-4), there would still easily be an infinite number of cases, like above, that are straightforward for humans but could cause the systems to fail unexpectedly. This limits the capacity of large language models to function as effective reasoners.

Reasoning for language generation is so challenging for machines due to several aspects:

First, different reasoning requires to satisfy diverse constraints. Those constraints can be from

linguistic structures, social aspects, or commonsense. Specifically, there are many forms of reasoning in our daily life, such as counterfactual reasoning, abductive reasoning, temporal reasoning, and many others. For example, *I went to a sushi bar last night, and wake up with violent stomach aches*. I could come up with an explanation that *I ate bad sushi*. That is the *abductive reasoning* that reasons about the most likely explanations. Then I am thinking, *what if I instead ate salad last night? I would be fine now*. That is the *counterfactual reasoning*, that imagines the alternative situations and answers "what-if" questions. In those reasoning activities, the outcome of the reasoning must satisfy many different constraints, such as being coherent with the context, being consistent with world knowledge, being fluent in language, and so on. What makes the problem even more challenging is the "language generation" aspect. That is, we are not solving a multiple-choice problem, where we may be given 3 options and just need to choose one answer out of the three. Instead, in practice it is an open-ended problem. We are reasoning in the infinite space of language. For example, we may imagine many different counterfactual situations, "*what if I ate beef taco?*", "*what if I cooked at home?*." So it is not a problem of choosing one out of three options, but rather a problem of reasoning for a few out of N to satisfy the constraints. Here N is the total number of possible text outcomes, and essentially is infinite. For example, for text of length 20 and vocabulary size 50000, N is 50000^{20}.

The second challenge of reasoning for language generation is the need for machines to understand the cause and effect of events. As in the above example, the language model needs to understand that changing from two cars to four does not change the outcome—it is still 3 hours. So the model needs to capture the causal "invariance", i.e., the aspects of outcome that are invariant under changing conditions. This is difficult for the models that are based on only data statistics.

The third challenge is about knowledge grounding. Large language models generate words without grounding on the world knowledge, often resulting in vacuous, hallucinatory, or inconsistent outputs. In contrast, humans perform reasoning and communication based on rich commonsense and background knowledge of the world in spatial, temporal, social, and other aspects. It is necessary to integrate neural reasoning with rich external knowledge, which has been a longstanding challenge in the field.

This thesis aims at tackling these three key challenges, to enable large pretrained language models to do flexible reasoning and generation:

In the first part, I will present our approach, **COLD decoding** [Qin et al., 2022], that steers language models to reason with constraints. To perform the many forms of reasoning, our approach develops a unified framework, based on energy-based modeling. It allows pretrained language models to combine with any constraints, without need of any training. Further, to tackle the difficulty of infinite space of language, our approach introduces differentiable reasoning on symbolic text. This makes the reasoning much more efficient than previous discrete methods.

In the second part about the challenge of cause-and-effect, I will dive deep into counterfactual reasoning, a particularly important type of reasoning that helps us understand the causal structure of the world. I will discuss our work **TimeTravel** [Qin et al., 2019a] and **DeLorean** [Qin et al., 2020a], that proposes the first formulation of counterfactual reasoning in language generation, and an algorithm that enables pretrained language models for counterfactual reasoning (and other forms of reasoning such as abductive reasoning). The new formulation exposes the fundamental limitation of the current language models for reasoning with causal invariance. In particular, current language models only do left-to-right prediction. They generate a new word conditioning on the past context on the left. However, causal invariance requires to also do right-to-left (nonmonotonic) inference. As it needs to consider the future context on the right, in order to capture what in the future is not changed. Language models face a fundamental limitation on such nonmonotonic reasoning. DeLorean is a new decoding algorithm to address the limitation.

In the third part, we study knowledge grounding of neural language reasoning and generation, particularly in the complex context of dialog. In **Conversing by Reading** [Qin et al., 2019b], we integrate external knowledge with neural models by augmenting the models to read and comprehend encyclopedic documents during generation. The approach effectively tackles the problems of vacuous and hallucinatory outputs in generating conversations. Additionally, I will present **TimeDial** [Qin et al., 2021] that focuses on the temporal knowledge and reasoning of neural language models in dialog. Specifically, we provide a systematic study and benchmark that reveals the difficulties of current language models on temporal reasoning in complex conversational context, motivating further research of knowledge grounding and contextual reasoning.

1.1 Overview of Dissertation

This dissertation consists of three parts as follow.

Differentiable Reasoning with Constraints. This part discusses "constrained" reasoning for neural language generation. One of the key challenges of language models is the lack of control. Reasoning in different scenarios requires producing text that is not only fluent, but also satisfies various constraints that control the generation semantics. These constraints can be hard (e.g., ensuring certain keywords are included in the output), soft (e.g., contextualizing the output with the left- or right-hand context), compositional (e.g., both following a template structure and taking a positive tone), and evolving over time (e.g., as the dialog unfolds). The dominant approach that fine-tunes a pretrained language model with task-specific data is prohibitively expensive and can hardly scale to the infinite possible combinations of constraints. I develop new inference (or *decoding*) algorithms that support plugging an arbitrary set of constraints in the pretrained language models and steering the generation on the fly. To overcome the longstanding challenge of discrete search in the extreme-scale language space, my work introduces the core methodological insight that performs decoding in a relaxed continuous space, permitting efficient *gradient-based* reasoning and generation.

Chapter 2 proposes COLD decoding Qin et al. [2022], a unified approach that formulates constrained generation as sampling from an energy-based model (EBM). The compositional control thus amounts to simply specifying an energy function by plugging in any desired constraint functions, then sampling from its induced energy-based distribution. We for the first time introduced Langevin dynamics [Welling and Teh, 2011] to the text-based EBMs, which performs sampling by iteratively updating the continuous text representations using gradients of the energy function. The general approach showed strong results on a range of constrained generation tasks, substantially outperforming the previous solutions specifically tailored to each task.

Counterfactual and Nonmonotonic Reasoning in Natural Language. This part discusses "causal" reasoning for neural language generation. Everyday causal reasoning is often *nonmonotonic*: for example, based on the partially observed past and future to reason about what might have happened in between (*abductive* reasoning), or conditioning on counterfactual past to reason about alternatives to the existing future (*counterfactual* reasoning). While humans do it effortlessly in our every waking moment, machines still struggle. Prior research in philosophy, logic, and other fields has studied those important forms of reasoning in classical perspectives detached from natural language. My work first fills this gap. I have worked towards

using natural language to characterize the machines' ability of counterfactual and nonmonotonic reasoning, and on this basis studied new reasoning algorithms that show strong promise.

Chapter 3 defines *counterfactual story revision* [Qin et al., 2019a], the first counterfactual reasoning and generation benchmark that formalizes a large-scale computational testbed. Differing from recent counterfactual classification tasks where models tend to exploit latent artifacts of the datasets, our formulation measures how machines can capture the *causal invariance* and reason about the causal chains in narratives. Specifically, given an original story and a counterfactual condition, machines are required to *minimally edit* the story ending to regain narrative consistency. The problem presents a significant challenge to the large language models even provided with supervisions. Furthermore, the results have motivated the DeLorean algorithm described later.

Chapter 4 introduces the DeLorean decoding algorithm [Qin et al., 2020a]. Specifically, the current common language models are "shortsighted" by facilitating only left-to-right prediction (i.e., generating the next word based on a past context). This presents a fundamental limitation in the problems where "foresight" is needed to *incorporate future context*, such as filling a blank given both left- and right-side text, and all the nonmonotonic reasoning problems including counterfactual reasoning (as above), abductive reasoning, and temporal reasoning in complex context (as below). Based on the key idea of differentiable reasoning over discrete text (as in COLD decoding in Chapter 2), I developed DeLorean, a new approach for the language models to look ahead while decoding. DeLorean augments the conventional left-to-right forward generation with the new right-to-left gradient "back-propagation". By alternating between the forward and backward propagation of information, DeLorean can decode the output that reflects both the past and future contexts.

Integration of Knowledge in Neural Language Model Reasoning. This part discusses "knowledge-grounded" reasoning for neural language generation. Reasoning in real-world situations requires rich background knowledge about how the physical and social world works. Large language models lack knowledge grounding, which often leads to vacuous or hallucinatory outputs especially in new situations. I have worked towards integrating the language models with knowledge in diverse forms, enabling improved contentfulness, faithfulness, and logical consistency of the reasoning outcomes.

Chapter 5 studies the integration of external knowledge with neural models. Humans reason by connecting knowledge from various sources. For example, when conversing about a topic, people often search

and acquire diverse external information as needed to continue a meaningful and informative conversation. Inspired by this, my work [Qin et al., 2019b] augmented conversation models to look at external long-form text (e.g., Wikipedia pages) on the fly. The models perform question-answering-style reading comprehension on this text in response to each conversational turn, thereby allowing for more focused integration of external knowledge than has been possible in prior approaches. The approach effectively improves both the informativeness and diversity of the generated output.

Chapter 6 studies reasoning with temporal concepts and knowledge. Everyday conversations require understanding everyday events, which in turn, requires understanding temporal commonsense concepts (durations, frequency, relative ordering, etc.) interwoven with those events. Temporal reasoning in dialog thus presents another common yet difficult reasoning challenge—inter-dependencies among temporal concepts appear across conversation turns, requiring operations like comparison and arithmetic inference combined with commonsense and world knowledge. Through my work on the TimeDial benchmark [Qin et al., 2021], I made the first systematic study of a broad set of 50+ methods on the problem, spanning different modeling paradigms and training settings, and revealed 23 absolute points of gap in accuracy between model and human-level performance.

Part I

Differentiable Reasoning with Constraints

Chapter 2

COLD Decoding: Energy-based Constrained Text Generation with Langevin Dynamics

The chapter discusses work originally published in [Qin et al., 2022].

Many applications of text generation require incorporating different constraints to control the semantics or style of generated text. These constraints can be hard (e.g., ensuring certain keywords are included in the output) and soft (e.g., contextualizing the output with the left- or right-hand context). In this work, we present *Energy-based Constrained Decoding with Langevin Dynamics* (COLD), a decoding framework which unifies constrained generation as specifying constraints through an energy function, then performing efficient differentiable reasoning over the constraints through gradient-based sampling. COLD decoding is a flexible framework that can be applied directly to off-the-shelf left-to-right language models *without* the need for any task-specific fine-tuning, as demonstrated through three challenging text generation applications: lexically-constrained generation, abductive reasoning, and counterfactual reasoning. Our experiments on these constrained generation tasks point to the effectiveness of our approach, both in terms of automatic and human evaluation.[1]

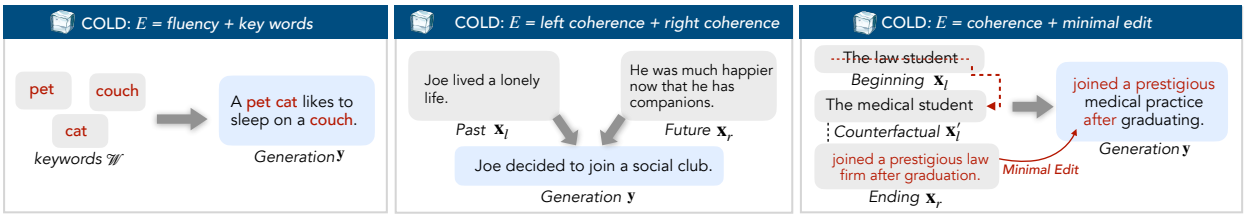
[1]Code is available at `https://github.com/qkaren/COLD_decoding`

2.1 Introduction

Many text generation applications require producing text that is not only fluent, but also satisfies various constraints which control the semantics or style of the generated text. For example (Figure 2.1), for knowledge-grounded or keyword-guided generation, we might want to ensure that certain keywords are included in the generated output as *hard* lexical constraints [Lin et al., 2020b; Welleck et al., 2021]. For other types of text generation, we often wish to incorporate *soft* topical constraints to contextualize the desired output, e.g., abductively [Peirce, 1974] reasoning about what happened in the middle of a story given the past and the future story context [Bhagavatula et al., 2019a]. Yet another class of text generation applications requires revising an input based on a new counterfactual condition [Goodman, 1947], which simultaneously requires semantic coherence as well as *minimal-edit* constraints with respect to the input text [Qin et al., 2019a].

The dominant paradigm to various text generation applications has been supervised learning with task-specific training data. However, different applications require varied and potentially evolving constraints, and annotating a large amount of task-specific training data for each different combination of constraints can be costly. Recent work has explored incorporating constraints through energy-based text modeling that alleviates the need of supervised data [Khalifa et al., 2020; Deng et al., 2020; Parshakova et al., 2019]. Yet those approaches still require expensive training of specific generation models. In addition, training might not even be feasible with recent models that are extreme in scale, like GPT-3 [Brown et al., 2020a]. This motivates the need to enrich *decoding* algorithms that can work directly with pretrained language models without task-specific fine-tuning, and support complex combinations of hard and soft constraints to control the generated text on the fly.

We propose a new constrained decoding approach that formulates decoding as sampling from an energy-based model (EBM) [Hinton, 2002; LeCun et al., 2006]. Constrained generation with our approach amounts to specifying an energy function by plugging in arbitrary constraint functions that are suitable for the task at hand, then sampling from its induced distribution. In particular, to overcome the longstanding challenges of sampling discrete text from EBMs, we for the first time introduce Langevin dynamics [Welling and Teh, 2011] to text-based EBMs for efficient *gradient*-based sampling. As a result, our approach, *Constrained Decoding with Langevin Dynamics* (COLD), performs sampling by iteratively updating a continuous relaxation



Lexically Constrained Generation | Abductive Reasoning | Counterfactual Reasoning

Figure 2.1: Applying COLD to different constrained generation tasks amounts to specifying an energy function E by plugging in relevant constraint functions. Text in grey boxes is the input, and text in blue boxes is the output.

of text using gradients of the energy function. The resulting continuous text samples are then mapped back to the discrete space with a simple guided discretization approach, yielding text sequences that are fluent and adhere to the constraints.

Our work makes unique contributions to a recent line of research investigating decoding algorithms for incorporating different constraints [Qin et al., 2020a; Dathathri et al., 2019; Lu et al., 2021; Kumar et al., 2021] in three distinct aspects. First, our formulation unifies various constrained generation scenarios that involve hard lexical constraints and/or soft contextual constraints: specifying an energy function, then sampling from its induced distribution. Second, we propose a *sampling* method, which complements decoding algorithms that look for a single optimal solution. Finally, we provide new empirical insights into the strengths and weaknesses of existing approaches to discrete search and differentiable reasoning.

To test the flexibility and empirical performance of COLD decoding, we experiment with three challenging text generation tasks: lexically constrained generation [Lin et al., 2020b; Hokamp and Liu, 2017], abductive reasoning [Bhagavatula et al., 2019a], and counterfactual story generation [Qin et al., 2019a]. COLD achieves better lexical coverage than NEUROLOGIC [Lu et al., 2021], a beam-based discrete decoding algorithm specifically designed for lexically constrained generation, while producing more coherent and higher quality text than DELOREAN [Qin et al., 2020a], a state-of-the-art gradient-based generation method for abductive reasoning and counterfactual reasoning. COLD supports all three constrained generation settings under a unified framework – specifying an energy function using a collection of fluency and task-specific constraints, then sampling from its induced distribution and achieves strong performance on both automatic and human evaluation.

2.2 Background

Neural text generation. Neural text generation typically involves two stages: modeling a distribution over text sequences, and using a *decoding algorithm* to generate sequences with the model. Let $\mathbf{y} = (y_1, \ldots, y_T)$ denote a discrete sequence where each y_t is a token from a vocabulary $\mathcal{V}$. Common neural language models (e.g., GPT-2/3 [Radford et al., 2019b; Brown et al., 2020a]) factorize the probability of a sequence into the product of per-token conditionals in left-to-right order, $p_\theta(\mathbf{y}) = \prod_{t=1}^{T} p_\theta(y_t|\mathbf{y}_{<t})$, with each conditional parameterized by a shared neural network, such as transformer [Vaswani et al., 2017b]. Popular decoding algorithms, ranging from beam search or greedy decoding to sampling methods such as top-k [Fan et al., 2018] or nucleus [Holtzman et al., 2019] sampling, produce text sequences $\mathbf{y}$ using the model p_θ, often conditioned on a prompt $\mathbf{x}$.

Constrained text generation. We view text generation as the problem of finding a sequence that satisfies a collection of constraints. For instance, the scenario above amounts to generating a sequence $\mathbf{y} = (y_1, \ldots, y_T)$ subject to a soft constraint that the continuation $\mathbf{y}$ should be fluent and logically coherent with the prompt $\mathbf{x}$. Other constrained generation problems impose additional constraints, such as text infilling [Zhu et al., 2019; Donahue et al., 2020] where coherence constraints move beyond a left-hand prefix, lexically constrained generation in which hard constraints require the output to contain given tokens, and various forms of semantically-constrained generation in which the output is softly constrained to be similar to another sequence. Since common decoding algorithms generate text monotonically, relying on $p_\theta(y_t|\mathbf{y}_{<t})$ for determining the next token, it is challenging to enforce these diverse constraints.

Energy-based models and Langevin dynamics. Given an energy function $E(\mathbf{y}) \in \mathbb{R}$, an energy-based model (EBM) is defined as a Boltzmann distribution $p(\mathbf{y}) = \exp\{-E(\mathbf{y})\}/Z$, where $Z = \sum_{\mathbf{y}} \exp\{-E(\mathbf{y})\}$ is the normalizing factor (The sum is replaced with an integral if $\mathbf{y}$ is continuous). EBMs are flexible, in that one can incorporate arbitrary functions such as constraints into the energy function $E(\mathbf{y})$. Recent work has thus made attempts to *train* text-based EBMs each for specific tasks [Hu et al., 2018; Parshakova et al., 2019; Deng et al., 2020; Khalifa et al., 2020]. As discussed earlier, we instead use the energy-based formulation to develop an *inference* (decoding) procedure that enables off-the-shelf pretrained language models to perform arbitrary constrained generation, without any fine-tuning.

Despite the flexibility, however, sampling from an EBM is particularly challenging, as computing Z is

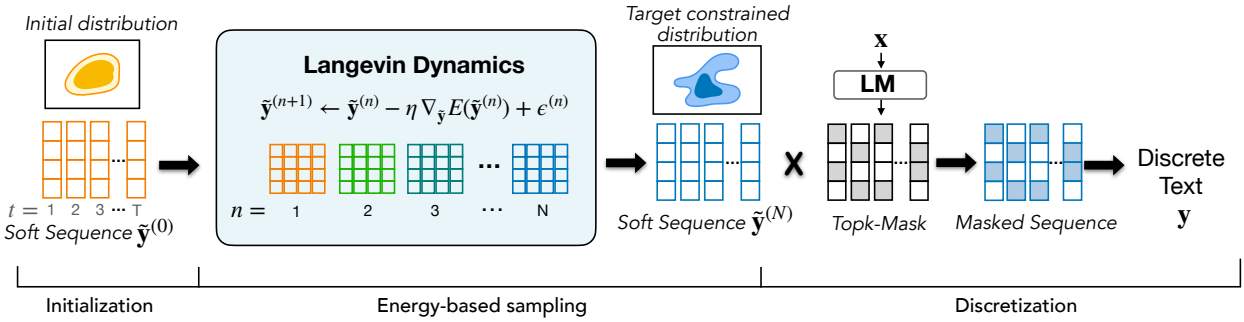


Figure 2.2: An overview of the COLD decoding procedure. Given an energy function $E(\tilde{\mathbf{y}}) = \sum_i \lambda_i f_i(\tilde{\mathbf{y}})$ with various constraints, the procedure starts with a soft sequence $\tilde{\mathbf{y}}^{(0)}$ as a sample from an initial energy-based distribution, and performs Langevin dynamics iterations using the gradient $\nabla_{\tilde{\mathbf{y}}} E(\tilde{\mathbf{y}})$ (Eq.2.2). The resulting sequence $\tilde{\mathbf{y}}^{(N)}$ after N iterations is approximately a sample from the desired constrained distribution. We then apply top-k filtering on the soft sequence to produce a discrete text sequence $\mathbf{y}$ (Eq.2.6).

intractable. Common gradient-free Markov chain Monte Carlo (MCMC) methods such as Gibbs sampling [Bishop and Nasrabadi, 2006] can be used, but they are often prohibitively slow [Duvenaud et al., 2021; Nijkamp et al., 2020]. Langevin dynamics [Welling and Teh, 2011; Neal et al., 2011; Ma et al., 2019], a gradient-based MCMC method, offers more efficient sampling by using the gradient of the energy function $\nabla_{\mathbf{y}} E(\mathbf{y})$, enabling sampling in domains such as image generation [Du and Mordatch, 2019; Song and Ermon, 2019]. However, since text is discrete, the gradient $\nabla_{\mathbf{y}} E(\mathbf{y})$ is not well-defined, making it non-trivial to apply Langevin dynamics for sampling text from an EBM. Our approach bridges this gap with continuous relaxation of text, differentiable constraints, and guided discretization, as described below.

2.3 COLD Decoding with Langevin Dynamics

To enable flexible and diverse constrained generation in off-the-shelf language models, we develop *Constrained Decoding with Langevin Dynamics* (COLD), a decoding approach that treats text generation as sampling from an energy-based distribution, allowing for flexibly composing constraints based on the task at hand. COLD decoding generates text by sampling from an EBM defined over a sequence of "soft" tokens using Langevin dynamics, then maps the continuous sample into discrete, fluent text. We provide our formulation of constrained text generation (§2.3.1), present differentiable constraints that can be composed into energy functions (§2.3.2) along with our discretization method (§2.3.3), and discuss practical details of COLD decoding (§2.3.4). Figure 2.2 provides an overview.

2.3.1 Energy-based Decoding

Constrained text generation aims to produce text samples $\mathbf{y}$ that satisfy a set of constraints (usually conditioned on an input $\mathbf{x}$ omitted for brevity). We assume each constraint can be captured by a constraint function $f_i(\mathbf{y}) \in \mathbb{R}$, where higher values of f_i mean that the text $\mathbf{y}$ better satisfies the constraint. For example, f_i could measure the likelihood of $\mathbf{y}$ as a fluency constraint (more in §2.3.2), while a hard constraint f_i amounts to a large negative penalty when $\mathbf{y}$ does not satisfy the constraint.

The set of constraints induces a distribution over text, written in an energy-based form as:

$$p(\mathbf{y}) = \exp\left\{\sum\nolimits_i \lambda_i f_i(\mathbf{y})\right\}/Z, \tag{2.1}$$

where $\lambda_i \geq 0$ is the weight of the ith constraint, Z is the normalizing factor. Here $E(\mathbf{y}) := -\sum_i \lambda_i f_i(\mathbf{y})$ is the energy function. This energy-based form is flexible, as one can plug in any constraint functions required for a task of interest. Generating text under the constraints can then be seen as sampling from the energy-based distribution $\mathbf{y} \sim p(\mathbf{y})$. One can also draw multiple samples and pick the best if only one sample is needed, as discussed later (§2.3.4).

As mentioned above, for efficient sampling from $p(\mathbf{y})$ we want to use Langevin dynamics, which makes use of the gradient $\nabla_{\mathbf{y}} E(\mathbf{y})$. However, in our case $\mathbf{y}$ is a discrete sequence and the gradient $\nabla_{\mathbf{y}} E(\mathbf{y})$ is not well-defined. As a result, we perform Langevin dynamics with an energy defined on a sequence of continuous token vectors, described below.

Differentiable decoding with Langevin dynamics. Instead of defining the energy function on discrete tokens, we define the energy function on a sequence of continuous vectors $\tilde{\mathbf{y}} = (\tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_T)$, which we call a soft sequence. Each position in the soft sequence is a vector $\tilde{\mathbf{y}}_t \in \mathbb{R}^V$, where V is the vocabulary size, and each element $\tilde{\mathbf{y}}_t(v) \in \mathbb{R}$ corresponds to the *logit* of word v in the vocabulary. Taking the softmax of $\tilde{\mathbf{y}}_t$ yields a distribution over the vocabulary for position t, $\tilde{\mathbf{p}}_t^\tau = \text{softmax}(\tilde{\mathbf{y}}_t/\tau)$. As $\tau \to 0$, $\tilde{\mathbf{p}}_t^\tau$ becomes a one-hot vector, indicating a discrete token.

By specifying an energy $E(\tilde{\mathbf{y}})$ on the soft sequence $\tilde{\mathbf{y}}$, we can use Langevin dynamics to obtain a sample.

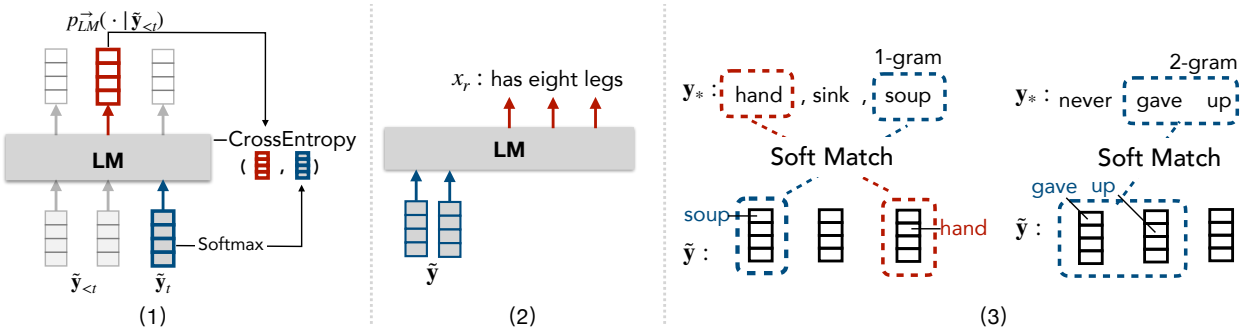


Figure 2.3: Illustrations of the differentiable constraints introduced in §2.3.2. **(1)** The soft fluency constraint (Eq.2.3) to encourage fluency of $\tilde{\mathbf{y}}_t$ based on LM probabilities. **(2)** The future contextualization constraint in Eq.(2.4) to encourage coherence w.r.t. the future context (`has eight legs`). **(3)** The n-gram similarity constraint in Eq.(2.5), where the left figure shows the case of $n = 1$ which encourages keywords (e.g., `hand`) to appear in the generation, and the right figure shows the case of $n > 1$ which is typically used to encourage sequence similarity with a reference text $\mathbf{y}_*$.

Specifically, the sampling is done by forming a Markov chain:

$$\tilde{\mathbf{y}}^{(n+1)} \leftarrow \tilde{\mathbf{y}}^{(n)} - \eta \nabla_{\tilde{\mathbf{y}}} E(\tilde{\mathbf{y}}^{(n)}) + \epsilon^{(n)}, \tag{2.2}$$

where $\eta > 0$ is the step size, and $\epsilon^{(n)} \in \mathcal{N}(0, \sigma)$ is the noise at iteration n. As shown in Welling and Teh [2011], by adding the right amount of noise and annealing the step size, the procedure will converge to samples from the true distribution. That is, if we let $p^{(n)}$ be the distribution such that $\tilde{\mathbf{y}}^{(n)} \sim p^{(n)}$, then as $n \to \infty$ and $\sigma \to 0$, we have $p^{(n)} \to p(\tilde{\mathbf{y}}) := \exp\{-E(\tilde{\mathbf{y}})\}/Z$. That is, the procedure ends up generating samples from the distribution induced by the energy function.

Next, we describe constraint functions defined on the soft sequence $\tilde{\mathbf{y}}$ that can be plugged in as components of the energy function. Later in §2.3.3, we describe how to obtain a discrete sequence from a soft sequence sample $\tilde{\mathbf{y}}$.

2.3.2 A Collection of COLD Constraints

COLD provides a flexible framework for plugging in a wide range of constraint functions for a task of interest. We describe constraint functions that are useful in various constrained generation problems, such as those we consider in the experiments (§2.4). The constraints include language model-based fluency constraints, along with lexical and semantic constraints on the sequence content. More generally, any differentiable function that outputs a goodness score of (soft) text can be used as a constraint function, as long

as it reflects the requirements of the target task.

Soft fluency constraint. Fluency is a common requirement for generated text. To promote fluency, we use a constraint which favors soft sequences that receive high probability according to the underlying left-to-right LM $p^{\rightarrow}_{\text{LM}}$ (e.g., GPT2):

$$f^{\rightarrow}_{\text{LM}}(\tilde{\mathbf{y}}) = \sum\nolimits_{t=1}^{T} \sum\nolimits_{v \in \mathcal{V}} p^{\rightarrow}_{\text{LM}}(v|\tilde{\mathbf{y}}_{<t}) \log \text{softmax}\left(\tilde{\mathbf{y}}_t(v)\right), \tag{2.3}$$

where $p^{\rightarrow}_{\text{LM}}(\cdot|\tilde{\mathbf{y}}_{<t})$ means the next-token distribution when providing the neural language model with the preceding soft tokens $\tilde{\mathbf{y}}_{<t}$ (i.e., feeding the weighted average of word embeddings, with the weights being $\text{softmax}(\tilde{\mathbf{y}}_{t'}/\tau)$ for $t' < t$ [Hu et al., 2017; Qin et al., 2020a]).

Intuitively, the constraint says that each token distribution in the soft sequence, $\text{softmax}\left(\tilde{\mathbf{y}}_t\right)$, must match the "reference" distribution $p^{\rightarrow}_{\text{LM}}(\cdot|\tilde{\mathbf{y}}_{<t})$ predicted by the underlying language model. The match is measured by the (negative) cross-entropy between the two distributions. The constraint thus encourages fluency. In practice, if there is left-side context $\mathbf{x}$ for the generation to condition on, we feed $\mathbf{x}$ to the LM to form the "reference" distribution $p^{\rightarrow}_{\text{LM}}(\cdot|\tilde{\mathbf{y}}_{<t}, \mathbf{x})$. As a result, $\tilde{\mathbf{y}}$ is encouraged to be fluent and coherent with the context $\mathbf{x}$.

We can easily incorporate an additional *reverse* LM constraint, $f^{\leftarrow}_{\text{LM}}$, using a right-to-left LM $p^{\leftarrow}_{\text{LM}}(\cdot|\tilde{\mathbf{y}}_{>t})$, as an additional fluency constraint. Flexibly leveraging multiple models in this way is infeasible with conventional decoding methods such as beam search or nucleus sampling.

Future-token prediction constraint. Applications such as text infilling involve future input tokens that remain fixed, but should contribute to updating past positions. For instance, consider updating the second position of `The ___ has eight legs`. A sample should be coherent with the tokens $\mathbf{x}_r$ on the right (i.e., `has eight legs`).

To this end, we use a constraint that adjusts soft tokens to maximize the likelihood of input tokens $\mathbf{x}_r$,

$$f_{\text{pred}}(\tilde{\mathbf{y}}; \mathbf{x}_r) = \sum\nolimits_{k=1}^{K} \log p^{\rightarrow}_{\text{LM}}(x_{r,k}|\tilde{\mathbf{y}}, \mathbf{x}_{r,<k}), \tag{2.4}$$

where K is the length of $\mathbf{x}_r$. In other words, the constraint adjusts the soft sequence $\tilde{\mathbf{y}}$ such that the underlying LM predicts the future tokens $\mathbf{x}_r$ after seeing $\tilde{\mathbf{y}}$.

Algorithm 1 Constrained Decoding w/ Langevin Dynamics.

input Constraints $\{f_i\}$, length T, iterations N.
output Sample sequence $\mathbf{y}$.
  $\tilde{\mathbf{y}}_t^{(0)} \leftarrow \texttt{init}()$ for all position t `// init soft-tokens`
  for $n \in \{1, \ldots, N\}$ **do**
    $E^{(n)} \leftarrow E(\tilde{\mathbf{y}}^{(n)}; \{f_i\})$ `// compute energy (§2.3.2)`
    $\tilde{\mathbf{y}}_t^{(n+1)} \leftarrow \tilde{\mathbf{y}}_t^{(n)} - \eta \nabla_{\tilde{\mathbf{y}}_t} E^{(n)} + \epsilon_t^{(n)}$ for all t `// update soft tokens (Eq.2.2)`
  end for
  $y_t = \arg\max_v \text{topk-filter}\left(\tilde{\mathbf{y}}_t^{(N)}(v)\right)$ for all t `// discretize (Eq.2.6)`
return: $\mathbf{y} = (y_1, \ldots, y_T)$

N-gram similarity constraint. Many constrained generation scenarios pose requirements on the wording and expression of generated text sequences. For instance, lexically constrained generation tasks [Hokamp and Liu, 2017] require certain keywords to be presented in the text samples, while counterfactual reasoning [Qin et al., 2019a] or text editing [Guu et al., 2018; Lin et al., 2020c] tasks require the text to retain the essence of a reference sequence.

We formulate these requirements as an n-gram similarity constraint which favors sequences that overlap with a reference $\mathbf{y}_*$ at the n-gram level,

$$f_{\text{sim}}(\tilde{\mathbf{y}}; \mathbf{y}_*) = \texttt{ngram-match}(\tilde{\mathbf{y}}, \mathbf{y}_*), \tag{2.5}$$

where $\texttt{ngram-match}(\cdot, \cdot)$ is a recent differentiable n-gram matching function [Liu et al., 2021] which can be seen as a differentiable approximation to the BLEU-n metric [Papineni et al., 2002]. When $n = 1$ and $\mathbf{y}_*$ a sequence of keywords, the constraint in effect enforces $\tilde{\mathbf{y}}$ to assign higher values to the keywords (1-grams). When n is larger and $\tilde{\mathbf{y}}_*$ is a reference sequence, the constraint encourages $\tilde{\mathbf{y}}$ to resemble the reference by assigning high values to tokens making up n-grams from $\mathbf{y}_*$.

2.3.3 From Soft to Discrete and Fluent Text

After receiving a soft sequence sample $\tilde{\mathbf{y}}$ from running Langevin dynamics (Eq. 2.2), we map the soft sequence to a discrete text sequence which we consider as the output of COLD decoding. A simple method would be selecting the most-likely token at each position t, $y_t = \arg\max_v \tilde{\mathbf{y}}_t(v)$. However, the resulting text can suffer from fluency issues even if the soft fluency constraint (Eq. 2.3) is used, due to competing con-

straints that sacrifice fluency. To overcome this, we use the underlying LM (e.g., GPT2-XL) as a "guardian" for obtaining the discrete sequence. Specifically, at each position t, we first use the LM to produce the top-k most-likely candidate tokens based on its generation distribution conditioning on preceding tokens, which we denote as $\mathcal{V}_t^k$. We then select from the top-k candidates the most likely token based on the soft sample $\tilde{\mathbf{y}}$:

$$y_t = \arg\max_{v \in \mathcal{V}_t^k} \tilde{\mathbf{y}}_t(v). \tag{2.6}$$

We refer to this method as "top-k filtering". The resulting text tends to be fluent because each token is among the top-k most probable tokens from the LM [Fan et al., 2018]. In practice, to ease the satisfaction of certain constraints (e.g. n-gram similarity), we expand the candidate set $\mathcal{V}_t^k$ to include constraint tokens (e.g., in the tasks of abductive reasoning §2.4.1 and lexically constrained decoding §2.4.3).

Figure 2.2 illustrates the decoding procedure to get one output from COLD decoding. Algorithm 1 summarizes the algorithm. Next, we move to practical considerations of applying COLD.

2.3.4 Implementation of COLD Decoding

Sample-and-select. COLD decoding allows for drawing multiple text samples from the distribution induced by the energy function $E(\tilde{\mathbf{y}})$. Depending on task requirements, we could either present the set of samples as output, or select one from the set based on some criteria (e.g., different energy terms) and return a single sequence, as in those tasks considered in the experiments (§2.4). This "sample-and-select" approach differs from deterministic constrained decoding methods, which optimize only one sequence [e.g., Lu et al., 2021; Kumar et al., 2021], and is used widely in various generation settings [e.g., Lee et al., 2021; Eikema and Aziz, 2021; Chen et al., 2021].

Initialization. We initialize the soft sequence $\tilde{\mathbf{y}}$ by running greedy decoding with the LM p_{LM} to obtain output logits. In our preliminary experiments, the initialization strategy had limited influence on the generation results.

Noise schedule. Each iteration of Langevin dynamics adds noise $\epsilon^{(n)} \sim \mathcal{N}(0, \sigma^{(n)})$ to the gradient (Eq. 2.2). We gradually decrease $\sigma^{(n)}$ across iterations, which intuitively transitions the decoding procedure from exploration to optimization. In our experiments, we typically used the schedule which sets/reduces σ

Models	Automatic Eval				Human Eval			
	$BLEU_4$	ROUGE-L	CIDEr	BERTScore	Grammar	Left-coherence ($\mathbf{x}_l\mathbf{y}$)	Right-coherence ($\mathbf{y}\mathbf{x}_r$)	Overall-coherence ($\mathbf{x}_l\mathbf{y}\mathbf{x}_r$)
LEFT-ONLY	0.88	16.26	3.49	38.48	**4.57**	3.95	2.68	2.70
DELOREAN	1.60	19.06	7.88	41.74	4.30	**4.23**	2.83	2.87
COLD (ours)	**1.79**	**19.50**	**10.68**	**42.67**	4.44	4.00	**3.06**	**2.96**

Table 2.1: Automatic and human evaluation of abductive reasoning (2.4.1). Our proposed method (COLD decoding) outperforms DELOREAN, a recent decoding algorithm achieving strong results in this task.

to $\{1, 0.5, 0.1, 0.05, 0.01\}$ at iterations $\{0, 50, 500, 1000, 1500\}$, respectively.

Long sequences. COLD decoding produces a fixed-length sequence $\mathbf{y} = (y_1, \ldots, y_T)$. To produce longer sequences, e.g. in cases where y_T is not the end of a sentence, we use p_{LM} to produce a continuation of $\mathbf{y}$ using greedy decoding.

2.4 Experiments

We evaluate COLD on three constrained generation tasks. Using COLD for each task amounts to specifying a set of task-specific constraints (instances of those in §2.3.2). Our focus is enabling constrained generation for settings in which fine-tuning is infeasible, through changing the decoding method. Thus, our experiments (i) use off-the-shelf LMs without fine-tuning, and (ii) compare COLD primarily against alternative decoding methods. As our base LM, we use GPT2-XL [Radford et al., 2019b].

2.4.1 Abductive Reasoning

We study a specific formulation of *abductive reasoning* [Peirce, 1974] as a language generation challenge. Specifically, given a beginning sentence $\mathbf{x}_l$ and an ending sentence $\mathbf{x}_r$, the abductive language generation (αNLG) problem [Bhagavatula et al., 2019a] consists of generating a bridge sentence $\mathbf{y}$ that fills in between the two sentences and forms a coherent full story (see Figure 2.1 for example). The task is particularly challenging for conventional monotonic left-to-right LMs (such as GPT-2 and GPT-3) since it requires non-monotonic reasoning that not only conditions on the past context ($\mathbf{x}_l$, on the left), but also the future story ending ($\mathbf{x}_r$, on the right).

The COLD Solution

COLD decoding can readily accommodate the abductive reasoning task by simply plugging in appropriate constraints to specify an energy function. Specifically, the generated text needs to be (1) fluent and consistent with the left context $\mathbf{x}_l$, and (2) coherent with the right context $\mathbf{x}_r$. Accordingly, we compose an energy using relevant constraints from §2.3.2:

$$E(\tilde{\mathbf{y}}) = \lambda_a^{lr} f_{\text{LM}}^{\rightarrow}(\tilde{\mathbf{y}}; \mathbf{x}_l) + \lambda_a^{rl} f_{\text{LM}}^{\leftarrow}(\tilde{\mathbf{y}}; \mathbf{x}_r) + \lambda_b f_{\text{pred}}(\tilde{\mathbf{y}}; \mathbf{x}_r) + \lambda_c f_{\text{sim}}(\tilde{\mathbf{y}}; \text{kw}(\mathbf{x}_r) - \text{kw}(\mathbf{x}_l)). \tag{2.7}$$

That is, we combine **(a)** a soft fluency constraint (Eq. 2.3) conditioning on the left sentence $\mathbf{x}_l$ to enforce fluency and consistency with the left context, and a reverse fluency constraint with a right-to-left LM conditioning on $\mathbf{x}_r$ to encourage coherence with the right context; **(b)** a future-token prediction constraint (Eq. 2.4) that enforces consistency between the generation $\mathbf{y}$ and the story ending $\mathbf{x}_r$; **(c)** a 1-gram similarity constraint (Eq. 2.5) between the generation $\mathbf{y}$ and keywords (non-stopwords) in $\mathbf{x}_r$ (excluding those in $\mathbf{x}_l$), i.e., $\text{kw}(\mathbf{x}_r) - \text{kw}(\mathbf{x}_l)$, which intuitively promotes a 'smooth transition' between $\mathbf{x}_l$, $\mathbf{y}$, and $\mathbf{x}_r$.

For the energy function in Eq.(2.7), we select the constraint weights on the dev set. Throughout the experiments, we set the number of Langevin dynamics steps to $N = 2000$, with a step size $\eta = 0.1$ (Eq. 2.2). We discuss more details of the configurations in the appendix.

Baselines. We compare with previous decoding approaches for this task. In particular, we compare with DELOREAN [Qin et al., 2020a] which outperformed a wide range of supervised and unsupervised methods on the abductive reasoning task in Qin et al. [2020a]. Following Qin et al. [2020a], we also compare with a LEFT-ONLY method that generates the continuation of $\mathbf{x}_l$ without considering the right-side $\mathbf{x}_r$, i.e., $\mathbf{y} \sim p_{\text{LM}}(\mathbf{y}|\mathbf{x}_l)$.

Evaluation. We perform both automatic and human evaluation. We adopt the standard automatic metrics on the task [Bhagavatula et al., 2019a] that measure the minimal edit between the generated text and the human-written references on the test set, including BLEU [Papineni et al., 2002], ROUGE [Lin, 2004], CIDEr [Vedantam et al., 2015], and BERTScore [Zhang et al., 2019a]. For the human evaluation, we follow [Qin et al., 2020a] and let crowdworkers from Amazon Mechanical Turk rate the generations on 200 test examples. Workers were presented a pair of observations ($\mathbf{x}_l$ and $\mathbf{x}_r$) and a generated hypothesis $\mathbf{y}$, and asked to rate the coherence of the hypothesis with respect to the observation $\mathbf{x}_l$ (i.e., $\mathbf{x}_l\mathbf{y}$), the observation $\mathbf{x}_r$ (i.e., $\mathbf{y}\mathbf{x}_r$), and both (i.e., $\mathbf{x}_l\mathbf{y}\mathbf{x}_r$), as well as the grammaticality of the hypothesis $\mathbf{y}$ itself, on a 5-point

Models	Min-Edit		Coherence	
	Overlap	Human	BERTS.	Human
LEFT-ONLY	50.56	1.21	73.83	2.30
Mix-Match [131]	85.07	–	65.20	–
Mix-Match$_L$ [131]	84.79	–	66.03	–
DELOREAN	52.90	1.81	73.66	1.92
COLD (ours)	**56.84**	**1.82**	73.47	**2.12**

Table 2.2: Automatic and human evaluation of counterfactual story rewriting. As a trivial method, LEFT-ONLY is coherent but fails on minimal-edit. COLD is superior to DELOREAN in terms of most metrics, including human evaluation.

Models	Coverage		Fluency	
	Count	Percent	PPL	Human
TSMH	2.72	71.27	1545.15	1.72
NEUROLOGIC	3.30	91.00	**28.61**	**2.53**
COLD (ours)	**4.24**	**94.50**	54.98	2.07

Table 2.3: Results of lexically constrained decoding (§2.4.3). For keyword coverage, we report both the average number and average percentage of constraint words present in the generated text. For language fluency, we use perplexity and human judgement.

Likert scale. The average ordinal Krippendorff alpha ($0 \leq \alpha \leq 1$) [Krippendorff, 2007] is 0.36, indicating a fair inner-annotator agreement.

Results

Table 2.1 shows the evaluation results on the abductive reasoning task. Under automatic evaluation (the left panel), COLD consistently outperforms the previous best unsupervised decoding algorithm DELOREAN, as well as the LEFT-ONLY method, in terms of both the lexical overlap metrics (BLEU, ROUGE and CIDEr) and semantic similarity metric BERTScore. The human evaluation (the right panel) provide more fine-grained insights. **COLD achieves the best overall coherence**, meaning that the generated $\mathbf{y}$ from COLD fits best with both the left-side context $\mathbf{x}_l$ and the right-side context $\mathbf{x}_r$ compared to the other methods. In contrast, DELOREAN excels only in terms of the left-side coherence (with $\mathbf{x}_l$), with inferior right-coherence (with $\mathbf{x}_r$). We speculate this is because of DELOREAN's complex interleaving of forward and backward decoding passes that make it difficult to balance the left- and right-coherence constraints. In terms of grammaticality, unsurprisingly, LEFT-ONLY obtains the best score as it ignores any other constraints (and fails this task with low coherence scores). More importantly, **COLD achieves a high grammaticality score** along with its high coherence, substantially improving over DELOREAN. Example generations in Appendix Table 2.7 show how COLD can reason with the right-hand context (e.g. 'no heels'), while DELOREAN's generations are contradictory ('red shoes' vs. 'white pair') or equivalent to those from LEFT-ONLY.

2.4.2 Counterfactual Story Rewriting

Next, we consider counterfactual story rewriting [Qin et al., 2019a]. Given a story context $\mathbf{x}_l$ with ending $\mathbf{x}_r$, the task is to generate a new story ending $\mathbf{y}$ that is (i) similar to the original ending $\mathbf{x}_r$, yet (ii) consistent with a new story context $\mathbf{x}'_l$ (see Figure 2.1 for example). The task is challenging as it requires capturing the aspects of future events that are invariant under the new (counterfactual) context, while only making necessary edits for coherence.

The COLD Solution

To tackle this task, we use COLD with an energy composed of constraint functions that promote coherence with the new context $\mathbf{x}'_l$, and minimal edits to the original ending $\mathbf{x}_r$:

$$E(\tilde{\mathbf{y}}) = \lambda_a^{lr} f_{\text{LM}}^{\rightarrow}(\tilde{\mathbf{y}}; \mathbf{x}'_l) + \lambda_a^{rl} f_{\text{LM}}^{\leftarrow}(\tilde{\mathbf{y}}) + \lambda_b f_{\text{sim}}(\tilde{\mathbf{y}}; \mathbf{x}_r). \tag{2.8}$$

These constraints combine: **(a)** a soft fluency constraint (Eq. 2.3) conditioned on $\mathbf{x}'_l$ to promote coherence between the generation $\mathbf{y}$ and the new (counterfactual) context $\mathbf{x}'_l$; a reverse LM constraint to improve fluency; **(b)** a n-gram similarity constraint (Eq. 2.5, $n = \{2, 3\}$) to encourage generating an ending $\tilde{\mathbf{y}}$ that is close to the original ending $\mathbf{x}_r$. We largely follow the configurations in §2.4.1 with some exceptions described in the appendix.

Baselines. Similar to the setup in §2.4.1, we compare with DELOREAN [Qin et al., 2020a], a recent state-of-the-art decoding algorithm. As a reference, we also report the performance of a trivial solution, LEFT-ONLY, that generates a continuation of $\mathbf{x}'_l$ without considering the minimal edit constraint with the original ending $\mathbf{x}_r$. Thus the method is expected to generate a coherent ending which however does not necessarily resemble the original ending. Finally, we compare with Mix-and-Match [Mireshghallah et al., 2022], a recent energy-based decoding method with discrete MCMC sampling, using BERT-base and BERT-Large.

Evaluation. We use the benchmark dataset TIMETRAVEL [Qin et al., 2019a]. The original data contains three sentences in a story ending. Due to computation constraints, we use the first sentence as the original ending and generate a new single-sentence ending accordingly. Following [Qin et al., 2019a, 2020a] we conduct both automatic and human evaluation. For automatic evaluation, we measure BERTScore [Zhang

et al., 2019a], and Minimal Edit which computes the overlap of text edits (insertion, deletion, replacement, etc.) [Šošić and Šikić, 2017] needed to produce the gold ending $\mathbf{y}_*$ and the generated ending $\mathbf{y}$, starting from the original ending $\mathbf{x}_r$. We do not use other common metrics such as BLEU since they were shown to be ineffective [Qin et al., 2019a]. For human evaluation, each crowdworker is presented with the original story ($\mathbf{x}_l$, $\mathbf{x}_r$), the counterfactual condition $\mathbf{x}'_l$, and the generated ending $\mathbf{y}$, and the worker is asked to rate (1) the coherence of $\tilde{\mathbf{y}}$ with respect to $\mathbf{x}'_l$ and (2) the extent to which the generated ending $\mathbf{y}$ preserves the details of the original ending $\mathbf{x}_r$ ("minimal edit"), on a 3-point Likert scale for 200 test examples. The average ordinal Krippendorff alpha is 0.52, indicating a moderate inner-annotator agreement. We exclude Mix-and-Match from human evaluation given the significant performance gap in automated evaluation.

Results

Table 2.2 shows the results of automatic and human evaluation in terms of both minimal-edit and coherence. As expected, the reference method LEFT-ONLY that completely ignores the minimal edit constraint can easily generate a new ending that is coherent with the new context $\mathbf{x}'_l$. Compared to the baseline approach DELOREAN, our method COLD achieves overall superior performance, with substantially improved coherence score and comparable minimal-edit score by human evaluation. Mix-and-Match, based on discrete MCMC sampling, performs poorly. Intuitively, its discrete sampling tends to get stuck in a mode of the target distribution (i.e., the region surrounding the original story ending), and struggles to explore further to find samples of interest. COLD's gradient-based sampling with continuous approximation leads to more efficient and effective exploration and mixing, as evidenced by samples that better meet the task requirements. See Appendix for examples.

2.4.3 Lexically Constrained Decoding

Next, we use COLD for lexically constrained decoding. Given a set of words $\mathcal{W}$, the task aims to generate a coherent sentence that contains these words (Figure 2.1). The task is challenging as it requires proper planning to coherently include the constraint words.

The COLD Solution

We specify an energy function of the following form:

$$E(\tilde{\mathbf{y}}) = \lambda_a^{lr} f_{\text{LM}}^{\rightarrow}(\tilde{\mathbf{y}}) + \lambda_a^{rl} f_{\text{LM}}^{\leftarrow}(\tilde{\mathbf{y}}) + \lambda_b f_{\text{sim}}(\tilde{\mathbf{y}}; \mathcal{W}) + \lambda_c f_{\text{pred}}(\tilde{\mathbf{y}}; c(\mathcal{W})). \tag{2.9}$$

Specifically, this energy function incorporates: **(a)** a soft fluency constraint (Eq. 2.3) and a reverse LM fluency constraint as in the previous tasks; **(b)** a 1-gram similarity constraint (Eq. 2.5) between the generation $\tilde{\mathbf{y}}$ and the given words $\mathcal{W}$; **(c)** a future-token prediction constraint, where we concatenate the constraint words (in an arbitrary order), denoted as $c(\mathcal{W})$, and use it as the right-side content $\mathbf{x}_r$ in Eq. (2.4). Again we use similar configurations as in §2.4.1. More details can be found in appendix.

Baselines. We compare with a recent state-of-the-art method NEUROLOGIC [Lu et al., 2021], a beam-search variant specifically designed for lexically constrained generation which outperformed many supervised and unsupervised approaches in Lu et al. [2021]. We also report the results of TSMH [Zhang et al., 2020a] as another recent baseline which uses Monte-Carlo Tree Search [Coulom, 2006].

Evaluation. We use the set of constraint words from the COMMONGEN corpus [Lin et al., 2020b], but adopt the *canonical* setting that the generated text must contain the exact constraint words (e.g., `write`) instead of their variants (e.g., `wrote`) [Hokamp and Liu, 2017; Sha, 2020]. Following previous works [Hokamp and Liu, 2017; Sha, 2020; Zhang et al., 2020a], we report a measure of constraint words coverage as well as language fluency by evaluating the perplexity of the text . We also ask crowdworkers to rate the text fluency on a 3-point Likert scale on 200 test examples. The average ordinal Krippendorff alpha is 0.29, indicating a fair inner-annotator agreement.

Results

Table 2.3 shows the evaluation results for the lexically constrained decoding task. COLD, a *general* constrained decoding method, is comparable to the state-of-the-art method NEUROLOGIC designed specifically for dealing with lexical constraints. In particular, COLD achieves a **higher coverage** of given keywords, at the expense of generating slightly less fluent language. COLD is also substantially better than lexically constrained decoding method TSMH in terms of both coverage and fluency.

2.4.4 Additional Analysis

Ablation studies. We ablate two important ingredients of our approach, namely the constraints and the top-k filtering. Due to space limit, we report the results of constraints and defer the results of top-k filtering to the appendix. Table 2.5 shows the human evaluation results for ablations of the constraints used on the abductive reasoning task (Eq. 2.7). The n-gram similarity constraint f_{sim} provides the largest contribution to the overall coherence. The reverse LM fluency constraint $f_{\text{LM}}^{\leftarrow}$ also to some extent helps with the right-side coherence by conditioning on the right-side content $\mathbf{x}_r$. Removing the future-token prediction constraint similarly causes inferior scores in terms of right-side and overall coherence, as expected. Removing the individual constraints leads to better grammaticality due to less competition among different constraints, at the cost of coherence. Our uniform treatment of all constraints as energy terms makes it straightforward to balance the different constraints by controlling the constraint weights.

Models	Gra-mmar	Left-coher. (x-y)	Right-coher. (y-z)	Overall-coher. (x-y-z)
COLD (Full)	4.17	3.96	**2.88**	**2.83**
COLD $-f_{\text{sim}}$	4.54	3.82	2.73	2.69
COLD $-f_{\text{LM}}^{\leftarrow}$	4.35	3.97	2.84	2.80
COLD $-f_{\text{pred}}$	**4.61**	**4.07**	2.75	2.77

Table 2.4: Ablation for the effect of different constraints in Eq.(2.7). We do human evaluation on 125 test examples. The best overall coherence is achieved when all the constraints are present.

Efficiency of COLD. We report the average runtime of generating one sample on the Counterfactual Story Rewriting data. The table below shows the results (on an NVIDIA Quadro GV100 GPU, batch size=32). We compare with Mix-and-Match [Mireshghallah et al., 2022], a recent energy-based decoding method with discrete MCMC sampling (Metropolis-Hastings, in particular). COLD, which uses gradient-based sampling, is faster than the gradient-free Mix-and-Match: COLD is 30% faster with base LMs of similar sizes (GPT2-M and BERTLarge), and has roughly the same time cost when using a much larger

Method	**Runtime (s)**
COLD (GPT2-XL, 1.5B)	33.6
COLD (GPT2-M, 355M)	22.7
Mix-and-Match (BERTLarge, 340M)	33.5

Table 2.5: COLD is more efficient than gradient-free Mix-and-Match [Mireshghallah et al., 2022]. The runtime shown is seconds per sample on Counterfactual Story Rewriting.

LM (GPT2-XL).

2.5 Related Work

Previous works proposed beam search variants for lexically constrained decoding [Hokamp and Liu, 2017; Pascual et al., 2020; Lu et al., 2021] which enforce constraints in a discrete space. Recent works consider constraint satisfaction by adjusting vocabulary distributions using an additional discriminator or LM [Dathathri et al., 2019; Krause et al., 2021; Yang and Klein, 2021]. Differing from those approaches that determine the generation token by token auto-regressively, Qin et al. [2020a] optimize the whole (soft) token sequence via gradient propagation, which facilitates sequence-level semantic constraints (e.g., right-coherence, minimal-edits). COLD also samples complete sequences, while offering a principled and unified formulation based on energy-based modeling. Kumar et al. [2021] extend [Hoang et al., 2017] by imposing constraints with a Lagrangian method and optimizing for a single output with gradient descent. In contrast, our approach based on energy-based sampling (§2.3.1) allows for generating samples for other utilities (e.g., rank-and-select §2.3.4, estimating expectations). We also introduce components for more fluent generations such as the novel discretization procedure. Also, on the empirical side, we explore a different class of problems and tackle them in the absence of labeled data. The recent CGMH [Miao et al., 2019a] and TSMH [Zhang et al., 2020a], followed by [Mireshghallah et al., 2022; Goyal et al., 2021], perform constrained decoding with extended Gibbs sampling or Metropolis-Hastings sampling in the discrete text space. Our energy-based formulation with gradient-based Langevin dynamics sampling produces substantially better results than the discrete TSMH (§2.4.3). Sha [2020] uses gradient information to guide generation, which, however, is specifically designed for lexically constrained generation.

Energy-based models (EBMs) have been used for incorporating additional information to train text generation models [Deng et al., 2020; Khalifa et al., 2020; Parshakova et al., 2019; Hu et al., 2018]. In contrast, we focus on the constrained decoding (*inference*) that can be directly applied to pretrained LMs without fine-tuning. Langevin dynamics is widely used on EBMs of modalities with continuous values, like images [Song and Ermon, 2019; Du and Mordatch, 2019; Zhao et al., 2021], 3D shapes [Xie et al., 2021], latent features [Pang et al., 2020], and audio sequences [Jayaram and Thickstun, 2021]. To our knowledge, we are the first to apply Langevin dynamics for (constrained) discrete text generation (with a continuous

approximation) for efficient sampling.

2.6 Conclusion

We introduce COLD decoding, an energy-based constrained text generation framework that can express various soft/hard constraints through an energy function, and sample using Langevin dynamics. COLD can be applied directly to off-the-shelf LMs without task-specific fine-tuning. We showcase its flexibility and strong performance on three distinct applications of constrained text generation.

2.7 Appendix

2.7.1 Experimental Configurations

Configurations of Abductive Reasoning. For the energy function in Eq.(2.7), we select the constraint weights on the dev set. The overall weight of the fluency constraints is set to 0.5, wherein the $f_{\text{LM}}^{\rightarrow}$ and $f_{\text{LM}}^{\leftarrow}$ constraints are balanced with a 6:4 ratio, leading to $\lambda_a^{lr} = 0.3$ and $\lambda_a^{rl} = 0.2$. The remaining weight 0.5 is assigned to the constraints (b) and (c), with a ratio of 1:0.05, leading to $\lambda_c^{lr} = 0.48$ and $\lambda_a^{rl} = 0.02$. Throughout the experiments, we set the number of Langevin dynamics steps to $N = 2000$, with a step size $\eta = 0.1$ (Eq. 2.2). The text decoded by COLD is set to have length 10 and is completed by the base LM as described in §2.3.4. We set the $k = 2$ for top-k filtering. For each $(\mathbf{x}_l, \mathbf{x}_r)$, we generate 16 samples and pick the best one by first ranking by the perplexity of the joint sequence $\mathbf{x}_l\mathbf{y}\mathbf{x}_r$ for overall coherence, and then from the top 5 candidates selecting the best one in terms of the perplexity of $\mathbf{y}\mathbf{x}_r$ for enhanced coherence with the right context.

Configurations of Counterfactual Story Rewriting. The constraint weights in the energy function in Eq. (2.8) are selected on the dev set. The weights of the constraints (a) and (b) are set to $\lambda_a^{lr} + \lambda_a^{rl} = 0.8$ and $\lambda_b = 0.2$, respectively. For the LM and reverse LM fluency constraints in (a), we use a ratio of 8:2, leading to $\lambda_a^{lr} = 0.64$ and $\lambda_a^{rl} = 0.16$. We largely follow the algorithm configurations in §2.4.1 except that the text length is set to 20, $k = 5$ for top-k filtering, and we generate 32 samples for each test example and pick the best one ranked by the perplexity of $\mathbf{x}'_l\mathbf{y}$.

Configurations of Lexically Constrained Decoding. The weights of the constraints in energy function

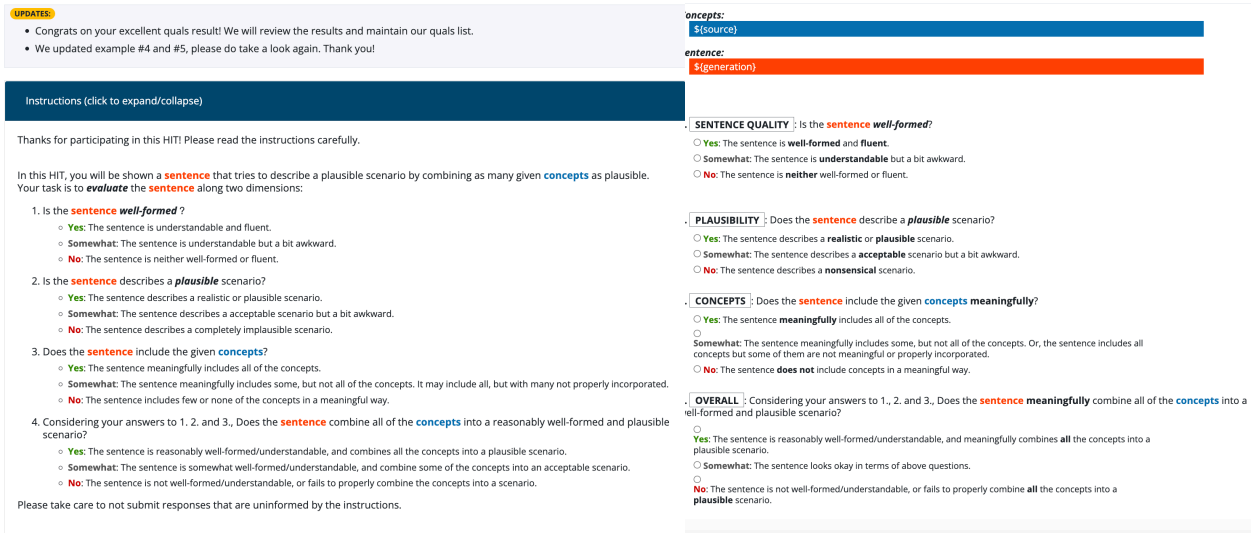


Figure 2.4: Screenshot of the mechanical turk interface used to gather human judgments for Lexically Constrained Generation.

Eq. (2.9) are the same as those in the abductive reasoning task (§2.4.1) except for the ratio of the n-gram similarity constraint, which is increased to 1:0.1 between constraints (b) and (c), leading to $\lambda_b = 0.05$ and $\lambda_c = 0.45$. We set the $k = 5$ for top-k filtering. All other configurations are the same as those in §2.4.1.

Right-to-left language model. The right-to-left LM is publicly released by West et al. [2021]. Specifically, the LM was trained following GPT-2 using the OpenWebText training corpus (see section 2.4 in West et al. [2021]).

Computing. All experiments were conducted using a server with 8 NVIDIA V100 GPUs.

2.7.2 Human Evaluation Details

Instructions of Human Evaluation We conduct human evaluation for 3 tasks: 1)Lexically Constrained Generation 2)Abductive Reasoning 3)Counterfactual Reasoning. We sampled 200 prompts randomly from the corpus for each human evaluation. We shuffle HITs to eliminate systematic bias of rater availability by time. Figures show the screenshot of instructions for our human evaluation.

Human Evaluation Payment Mean hourly pay was determined using a javascript timing tool to be $15/hr.

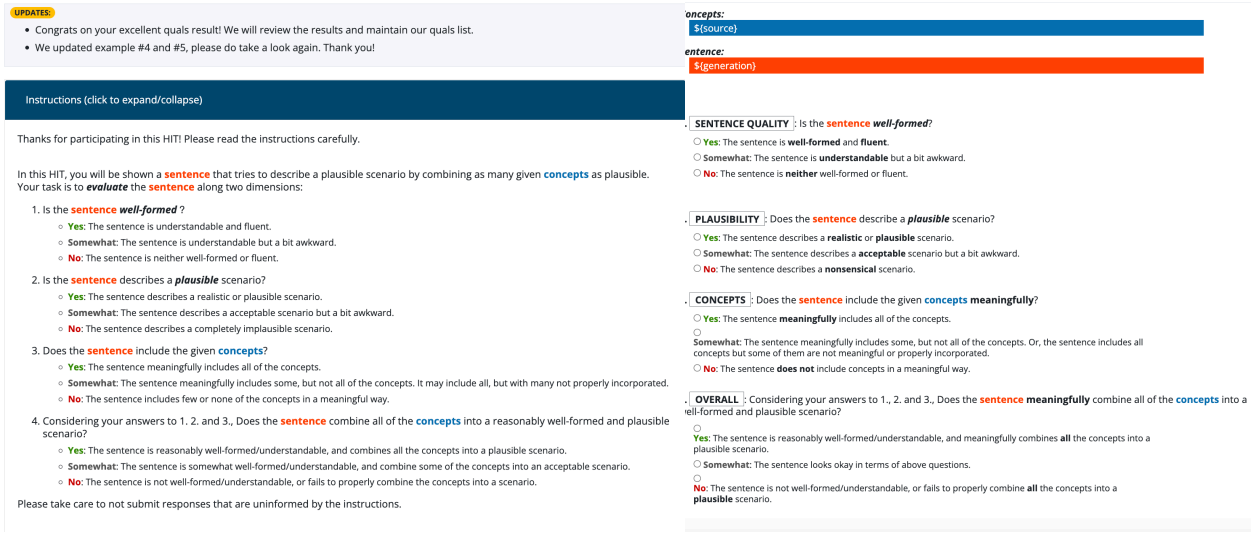


Figure 2.5: Screenshot of the mechanical turk interface used to gather human judgments for Abductive Reasoning.

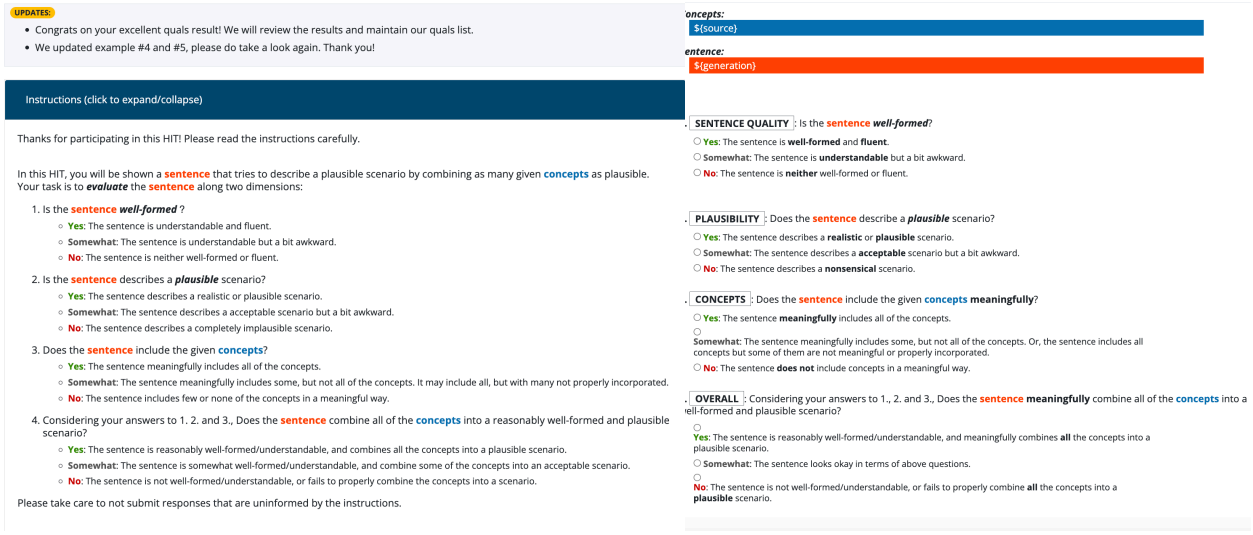


Figure 2.6: Screenshot of the mechanical turk interface used to gather human judgments for Counterfactual Reasoning.

top-k	Grammar	Left-coher. (x-y)	Right-coher. (y-z)	Overall-coher. (x-y-z)
2	**4.38**	**3.99**	2.88	2.92
5	4.27	3.71	3.04	2.87
10	4.09	3.84	**3.09**	**2.94**
50	3.95	3.62	3.07	2.87
100	3.80	3.54	3.03	2.84

Table 2.6: Ablation for the effect of k in top-k filtering mechanism (§2.3.3). We use the same setting as Table 2.5.

2.7.3 Ablation Study: Top-k Filtering

We investigate the role of top-k filtering mechanism (§2.3.3). Specifically, we investigate its effect on the output performance for different k values in Table 2.6. We can see that the grammar score tends to decrease as k increases. This is expected since a larger k indicates more flexibility for the generation to satisfy other constraints, often at the expense of fluency. The left coherence shows a similar relationship with the k value since it is also enforced by the left-to-right LM through the soft fluency constraint (Eq.2.3). In contrast, the right and overall coherence generally benefits from a larger k due to the increased flexibility for choosing the right words. Interestingly, with a large k value (50, 100), the right/overall coherence no longer improves, probably due to the inferior fluency that has affected the meaning and coherence of the generation.

2.7.4 Generated Samples

Tables 2.7, 2.8, and 2.9 show generated samples for the abductive reasoning, counterfactual reasoning, and lexically constrained decoding tasks, respectively.

Begin. $\mathbf{x}_l$	I bought a great pair of red shoe at the shoe store.
End. $\mathbf{x}_r$	I ended up getting a white pair with no heels.
LEFT-ONLY	I was going to wear them to the beach, but I didn't want to be the only one.
DELOREAN	I was going to buy a pair of black shoes, but I decided to go with red shoes because I like red shoes.
COLD	I was going to buy heels but they were out of stock.
Begin. $\mathbf{x}_l$	Arnold was scared of cats.
End. $\mathbf{x}_r$	Arnold dumped his girlfriend.
LEFT-ONLY	He was afraid of the dark.
DELOREAN	He was afraid of the dark.
COLD	He had girlfriend who was a cat lover.

Table 2.7: Examples for abductive reasoning.

Orig. context $\mathbf{x}_l$	Jon decided to go to the pawn store. He found a bornite-coated chalcopyrite crystal.
Orig. ending $\mathbf{x}_r$	He bought it for three thousand dollars.
Counterfactual $\mathbf{x}'_l$	He sold some antiques he had found.
LEFT-ONLY	He bought a few books.
DELOREAN	He bought it for three thousand dollars.
COLD	He bought a thousand dollars' worth of gold.
Orig. context $\mathbf{x}_l$	Peyton and Tom played football often. Tom always won for many Year's.
Orig. ending $\mathbf{x}_r$	Peyton never gave up and kept practicing.
Counterfactual $\mathbf{x}'_l$	Peyton always won for many years.
LEFT-ONLY	Tom was a great quarterback.
DELOREAN	Tom was a great quarterback.
COLD	Tom never gave up and never gave in.

Table 2.8: Examples for counterfactual reasoning.

Keywords $\mathbf{x}_l$	hand, sink, soap, wash
TSMH	They **wash** with their **hands** they **wash** at the **sinks soaps** they **wash**
NEUROLOGIC	I hand **wash** my clothes in the **sink**, **soap** and water.
COLD	The **sink soap** is a **hand wash soap** made from natural ingredients.
Keywords $\mathbf{x}_l$	cream, leg, put, shave
TSMH	I **creamed** my bare **legs** and **put**.
NEUROLOGIC	I **put shave cream** on my **leg**.
COLD	The first time I ever **put** a **leg** in **shave cream** was when I was a kid.

Table 2.9: Examples for lexically constrained generation.

Part II

Counterfactual and Nonmonotonic Reasoning in Natural Language

Chapter 3

TimeTravel: Counterfactual Story Reasoning and Generation

The chapter discusses work originally published in [Qin et al., 2019a].

Counterfactual reasoning requires predicting how alternative events, contrary to what actually happened, might have resulted in different outcomes. Despite being considered a necessary component of AI-complete systems, few resources have been developed for evaluating counterfactual reasoning in narratives.

In this work, we propose *Counterfactual Story Rewriting*: given an original story and an intervening counterfactual event, the task is to minimally revise the story to make it compatible with the given counterfactual event. Solving this task will require deep understanding of causal narrative chains and counterfactual invariance, and integration of such story reasoning capabilities into conditional language generation models.

We present TIMETRAVEL, a new dataset of 29,849 counterfactual rewritings, each with the original story, a counterfactual event, and human-generated revision of the original story compatible with the counterfactual event. Additionally, we include 80,115 counterfactual "branches" without a rewritten storyline to support future work on semi- or un-supervised approaches to counterfactual story rewriting.

Finally, we evaluate the counterfactual rewriting capacities of several competitive baselines based on pretrained language models, and assess whether common overlap and model-based automatic metrics for text generation correlate well with human scores for counterfactual rewriting.

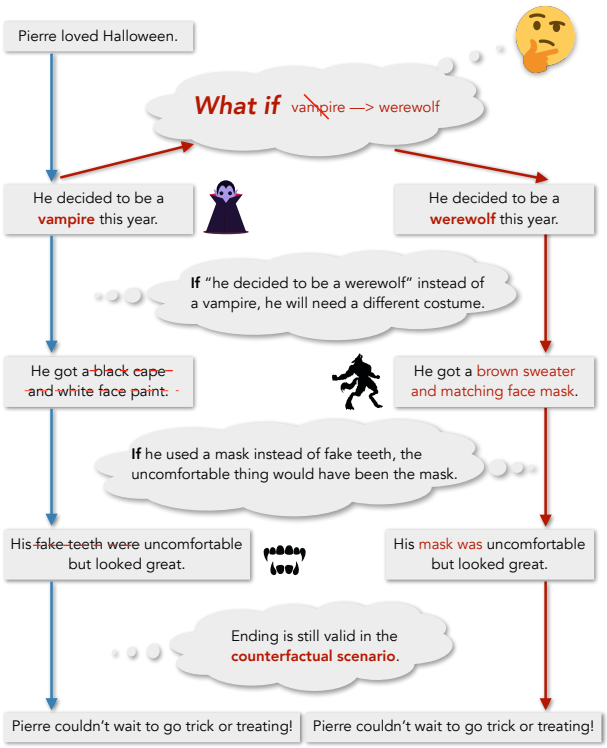


Figure 3.1: Given a short story (left column) and a *counterfactual context* ("He decided to be a werewolf this year"), the task is to revise the original story with minimal edits to be consistent with both the original premise ("Pierre loved Halloween") and the new counterfactual situation. The modified parts in the new story (right column) are highlighted in red.

3.1 Introduction

A desired property of AI systems is counterfactual reasoning: the ability to predict causal changes in future events given a counterfactual condition applied to the original chain of events [Goodman, 1947; Bottou et al., 2013]. For example, given an original story shown in the left chain in Figure 3.1, where "Pierre loved Halloween. He decided to be a *vampire* this year. He got a *black cape and white face paint*..." and a counterfactual condition, "what if Pierre decided to be a *werewolf* instead of a *vampire*?", an intelligent system should be able to revise the subsequent events in the story appropriately, for example, that a *brown sweater* would be more appropriate than a *black cape*.

This notion of counterfactuals has become increasingly relevant in several recent benchmarks such as ROC story cloze [Mostafazadeh et al., 2016b], COPA [Roemmele et al., 2011b], and HellaSwag [Zellers et al., 2019a], where the negative responses in multiple-choice problems implicitly construct counterfactual

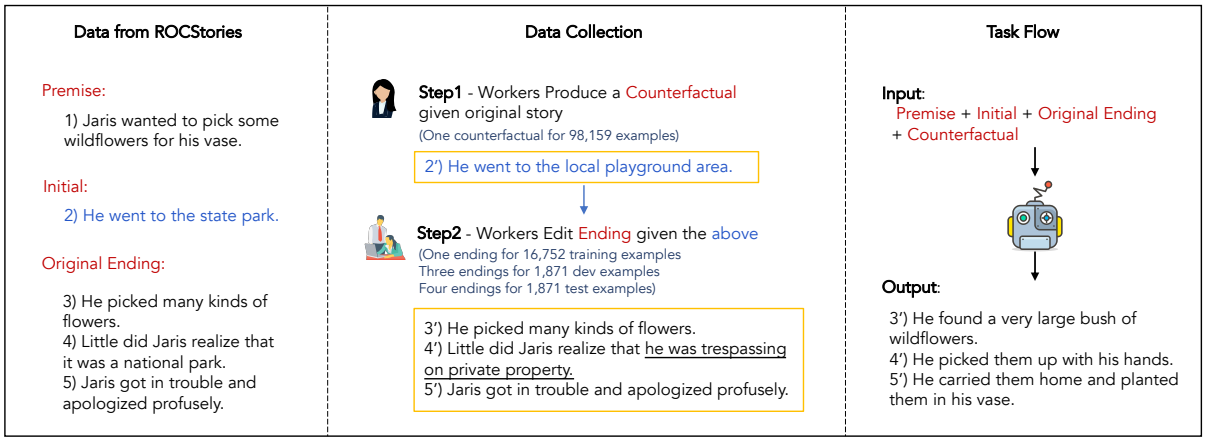


Figure 3.2: Data annotation process for the TIMETRAVEL dataset. Given a story from the ROCStories corpus, crowdworkers write a counterfactual sentence w.r.t the second sentence of the story. The counterfactual sentence and the original story are then presented to other workers to rewrite the story ending. Models for the task are expected to generate a rewritten ending given the original story and counterfactual sentence.

narratives. However, no existing benchmark to date has been designed to explicitly evaluate counterfactual narrative reasoning and revision as its principal focus, where a system is evaluated on its ability to make modifications to future events based on a counterfactual condition, as illustrated in Figure 3.1.

In this work, we introduce *Counterfactual Story Rewriting* as a new challenge to story understanding and generation. Given an original story and a counterfactual condition, the task is to re-write the story to regain narrative consistency through counterfactual reasoning. An important challenge in counterfactual reasoning is *causal invariance*, namely, the aspects of future events that are invariant under the counterfactual conditions. This is necessary to accurately reason about the new consequences with minimal edits to the original sequence of events, instead of being confounded by spurious correlations [Woodward, 2002; Bottou, 2019]. Therefore, a key measure of the task besides consistency is that the rewriting must perform *minimal edits* to the original story. This challenges the system to reason about causal invariance, which in turn, challenges the system to reason more carefully about the causal chains of how the story unfolds.

We introduce TIMETRAVEL, a new dataset with 29,849 counterfactual revisions to support research on counterfactual narrative reasoning and revision. In addition, our dataset provides 80,115 counterfactual *branches* without rewritten storylines to support potential future work on semi- or un-supervised approaches. Figure 4.2 illustrates (1) the structure of the original stories, (2) the counterfactual data construction process, and (3) the final task definition.

We establish baseline performances of state-of-the-art neural language models on this task, such as GPT [Radford et al., 2018] and GPT-2 [Radford et al., 2019b], evaluated in zero-shot, unsupervised, and supervised learning settings. Empirical results indicate that while these models are able to capture certain instances of counterfactual reasoning, they generally struggle with rewriting endings with full consistency. Our results suggest that current neural language models operate based primarily on frequent patterns in language without true understanding of the causal chains in narratives, thus requiring more focused future research to integrate reasoning capabilities in neural language models. [1]

3.2 Background

Counterfactual reasoning is the ability to consider *alternative possibilities* that diverge from current observed narratives. Due to their prevalence in common reasoning situations, counterfactuals have been studied in a wide range of disciplines, including psychology [Epstude and Roese, 2008], cognitive science [Byrne, 2002], as well as natural language processing [Hobbs, 2005; Lawrence and Riezler, 2018a; Son et al., 2017b].

Meanwhile, despite the progress made in NLU tasks by adapting pretrained language representations such as BERT [Devlin et al., 2018a] or GPT [Radford et al., 2018], models still have trouble discriminating between reasonable and unreasonable counterfactuals, as shown in [Zellers et al., 2019a]. Moreover, success in tasks linked to discrimination of reasonable alternatives often results in models learning to exploit latent artifacts of the dataset [Niven and Kao, 2019; Zellers et al., 2018], rather than learning to robustly reason about counterfactuals. In response to this, we hypothesize that learning to *generate the result* of counterfactual prompts will encourage models to learn to understand the underlying dynamics of a given situation, whereas discrimination between two alternatives is more likely to take advantage of dataset biases.

This goal shares many similarities with script learning [Pichotta and Mooney, 2014; Chambers, 2013], which attempts to canonicalize stereotypical event sequences for learning causal structure of narratives. However, because it is often difficult to capture the richness of causal dependencies with templatized structures [Sap et al., 2019], we instead study counterfactual reasoning in unstructured text directly and also require the model to *generate* the consequences of the counterfactual reasoning.

[1] Code and data are available at `https://github.com/qkaren/Counterfactual-StoryRW`.

The "counterfactual event" in our task can be viewed as a causal intervention [Pearl, 2000] in the latent chain of events of the story. Such interventions demand changes to the written narrative in order to abide by the shared background knowledge that human readers have about how the world works. This neatly embeds the problem of causal reasoning in a space that laymen with no knowledge of formalized causality can understand. It also allows us to evaluate the capabilities and limitations of the recent advances in neural language models in the context of counterfactual reasoning.

Similar issues arise in the area of controllable language generation [e.g., Hu et al., 2017], which involves preserving the content of text while changing it along a single or multiple dimensions, such as theme [Koncel-Kedziorski et al., 2016b], style [Lample et al., 2019b], and sentiment [Shen et al., 2017]. Reasoning in these tasks is limited to discrete axes (e.g., sentiment), which are often categorized with a closed label set ({positive, negative}). Because of controllability motivations, these axes and labels are generally known *a priori*. In contrast, counterfactual rewriting focuses on the causes and effects of a story, dimensions that can require more complex and diverse, yet potentially subtle, changes to accommodate the counterfactual event. Additionally, we put no restrictions on the nature of counterfactual events, yielding no clear set of discrete axes along which the story can change and no closed set of labels for them.

3.3 Counterfactual Story Rewriting

Premise	Alec's daughter wanted more blocks to play with.
Initial	Alec figured that blocks would develop her scientific mind.
Original Ending	Alec bought blocks with letters on them. Alec's daughter made words with them rather than structures. Alec was happy to see his daughter developing her verbal ability.
Counterfactual	Alec couldn't afford to buy new blocks for his daughter.
Edited Ending	Alec decided to make blocks with letters on them instead. Alec's daughter made words with the blocks. Alec was happy to see his daughter developing her verbal ability.
Premise	Ana had just had a baby girl.
Initial	She wanted her girl to have pierced ears.
Original Ending	She took her baby to the studio and had her ears pierced. Then she fastened tiny diamond studs into the piercings. Ana loved the earrings.
Counterfactual	She didn't like the idea of having her ears pierced.
Edited Ending	She decided not to take her baby to the studio to get her ears pierced. So she took tiny diamond stickers and stuck them to her ear. Ana loved the fake earrings.

Table 3.1: Examples from TIMETRAVEL

3.3.1 Task

We now formally introduce the task and establish the notation used in the paper. Each example consists of a five-sentence story $S = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_5)$ with a general structure where the first sentence $\boldsymbol{s}_1$ sets up the *premise*, the second sentence $\boldsymbol{s}_2$ provides more information of the *initial context*, and the last three sentences $\boldsymbol{s}_{3:5}$ are the *original ending* of story. We are further given an additional sentence $\boldsymbol{s}'_2$, which is counterfactual to the initial context $\boldsymbol{s}_2$. That is, $\boldsymbol{s}'_2$ states something contrary to that in $\boldsymbol{s}_2$, which in turn can make the original ending $\boldsymbol{s}_{3:5}$ no longer valid. Thus, the goal of the task is to rewrite the ending, such that the *edited ending* $\boldsymbol{s}'_{3:5}$ minimally modifies the original one and regains narrative coherency to the new counterfactual context.

The minimum edit goal differentiates our task from previous story ending studies, which have mostly focused on consistency in a given context. To achieve consistency with minimal edits, a model must understand the key mechanisms that drive the story's narrative so that it can filter out spurious correlations and capture counterfactual invariance. We thus consider the new task as a suitable testbed for studying counterfactual reasoning in combination with language generation.

3.3.2 Dataset: TIMETRAVEL

Our dataset is built on top of the ROCStories corpus [Mostafazadeh et al., 2016b], which contains 98,159 five-sentences stories in the training set, along with 3,742 stories in the evaluation sets. Each story was written by crowdworkers. To collect counterfactual events and new story continuations for TIMETRAVEL, we employ workers from Amazon Mechanical Turk (AMT) for a two-step task, which we describe in detail below.

3.3.3 Data Collection

Counterfactual Event Collection We present workers with an original five-sentence story $S = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_5)$ and ask them to produce a counterfactual event $\boldsymbol{s}'_2$ based on $\boldsymbol{s}_2$. Workers are instructed to produce counterfactual sentences $\boldsymbol{s}'_2$ that are:

(1) Topically related to the original context sentence $\boldsymbol{s}_2$, rather than a completely new sentence.

(2) Relevant to the original premise sentence, $\boldsymbol{s}_1$, allowing for a coherent story continuation.

(3) Influential to the subsequent storyline, such that at least one of the original ending's sentences, $\{\boldsymbol{s}_3, \boldsymbol{s}_4,$

	Train	Valid	Test
ROCStories data:			
# Stories	98,159	1,871	1,871
TIMETRAVEL*:*			
# Counterfactual Context	96,867	5,613	7,484
# Edited Ending	16,752	5,613	7,484

Table 3.2: Dataset statistics

$\boldsymbol{s}_5\}$ is no longer appropriate given $\boldsymbol{s}_1$ and $\boldsymbol{s}_2'$, necessitating a rewritten story ending.

Continuation Rewriting Once a counterfactual sentence $\boldsymbol{s}_2'$ is provided, we present it to a new set of workers with the original story $S = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_5)$. Now that $\boldsymbol{s}_2'$ invalidates the original storyline, workers are instructed to make minimal edits to $\boldsymbol{s}_{3:5}$, such that the narrative is coherent again. Before beginning, workers are instructed to validate whether the counterfactual event satisfies the requirements from the previous stage of the pipeline. If not, we ask them to rewrite the counterfactual again, and the continuation rewriting step is reassigned to a new worker.

Summary We provide examples from the TIMETRAVEL dataset in Table 3.1 and summarize its scale in Table 6.3. Overall, we collect 16,752 training examples of a counterfactual context and a rewritten ending. We also collect an additional 80,115 counterfactual contexts for the training set with no rewritten ending to support future work in unsupervised learning on this task. For the development and test sets, we gather multiple counterfactual contexts and rewritten endings for *each* example (3 new endings for development and 4 for test). Information regarding quality control and cost are provided in Appendix 3.8.1.

3.4 Learning a Counterfactual Rewriter

Recent work in constructing large-scale generative language models based on transformers [Radford et al., 2018, 2019b] has led to considerable improvements in natural language generation tasks. Due to their current prominence, we use them as baselines to study the extent to which the current neural text generation systems can perform and fail counterfactual narrative reasoning and revision. We focus on the family of GPT models, including GPT [Radford et al., 2018] and the latest small- (GPT2-S) and medium-sized (GPT2-M)

transformer models from Radford et al. [2019b]. For each of the three pretrained language models, we fine-tune with multiple objectives, leading to 14 different model variants for the task, which we describe in more detail below.

3.4.1 Unsupervised Training

Constructing large-scale counterfactual revision dataset is costly. Therefore, an ideal system must learn to reason without direct supervision. Toward this goal, we examine how unsupervised approaches to counterfactual story rewriting perform on our evaluation task. We devise the following unsupervised settings for models to learn to generate counterfactual story endings.

Zero-shot (ZS) In our simplest setting, we evaluate the counterfactual reasoning abilities already learned by these models due to pretraining on large corpora: the BooksCorpus dataset [Zhu et al., 2015] for GPT and the WebText corpus for GPT-2 [Radford et al., 2019b]. In this setting, models are not trained on any portion of the training data from TIMETRAVEL and must instead produce counterfactual rewritten stories for the evaluation set using only the representations learned from pretraining. At test time, the model receives the premise and the counterfactual context $(\boldsymbol{s}_1, \boldsymbol{s}'_2)$ as input and generates the tokens that constitute the rewritten counterfactual outcome.

Fine-tuning (FT) Because the domains on which both the GPT and GPT2 models were trained are broad and more complex than the domain of ROCStories, we investigate whether adapting the language model to the data distribution of ROCStories is helpful for learning to reason about counterfactuals. In this setting, the model is further fine-tuned to maximize the log-likelihood of the stories in the ROCStories corpus:

$$\mathcal{L}^{ft}(\boldsymbol{\theta}) = \log p_\theta(S), \tag{3.1}$$

where p_θ is the language model with parameters $\boldsymbol{\theta}$, and S is the original story as defined in Section 3.3.1. This fine-tuning step encourages the model to generate text with the same consistent style of the stories. Similar to the zero-shot setting, the premise and the counterfactual sentence $(\boldsymbol{s}_1, \boldsymbol{s}'_2)$ are provided as input to the model.

Fine-tuning + Counterfactual (FT + CF) The above training loss, however, does not make use of the additional 81,407 counterfactual training sentences for fine-tuning. To inform the model with a larger set of possible counterfactual narratives in the training data, we propose an additional loss function that fits the model to the counterfactual sentences given the premise sentence:

$$\mathcal{L}^{cf}(\boldsymbol{\theta}) = \log p_\theta(\boldsymbol{s}_2'|\boldsymbol{s}_1), \tag{3.2}$$

where $p_\theta(\boldsymbol{s}_2'|\boldsymbol{s}_1)$ denotes that the language model first reads the premise $\boldsymbol{s}_1$ and maximizes the log-likelihood of counterfactual sentence $\boldsymbol{s}_2'$. The model is fine-tuned with both objectives in Eqs equation 3.1 and equation 3.2:

$$\mathcal{L}^{ft+cf}(\boldsymbol{\theta}) = \mathcal{L}^{ft} + \mathcal{L}^{cf}, \tag{3.3}$$

and receives inputs in the same format as the zero-shot and fine-tuned models at test time.

Reconstruction + Counterfactual (RC + CF) One issue with the above training procedures is that models are not explicitly trained to retain as much text of the original outcome $\boldsymbol{x}_{3:5}$ as possible (i.e., minimum edits). If these models are to learn to "rewrite" the original story ending given the counterfactual sentence, rather than learning to produce a completely new plot, they must be able to condition on the original ending during generation. Motivated by this requirement and following the goal of developing unsupervised methods for counterfactual rewriting, we design a reconstruction objective for learning a noisy reproduction of the original ending. Specifically, we provide the model with both the original story and a masked context as input $(S, [s], \boldsymbol{s}_1, [mask])$ and train the model to reconstruct the original ending $\boldsymbol{s}_{3:5}$:

$$\mathcal{L}^{rc}(\boldsymbol{\theta}) = \log p_\theta(\boldsymbol{s}_{3:5}|S, [s], \boldsymbol{s}_1, [mask]), \tag{3.4}$$

where `[s]` denotes a separator token and `[mask]` is a special mask token. In this setting, the model first reads the original story S followed by the separation token `[s]`, and then reads the premise $\boldsymbol{x}_1$ again, followed by the mask token `[mask]`, which serves as a placeholder sentence for the counterfactual sentence. This objective encourages the model to reproduce the original ending $\boldsymbol{s}_{3:5}$ in the general case where the

Premise	Rob loved to play soccer and was on his middle school team.
Initial	They had just started their season and were playing their first game.
Original Ending	Rob accidentally tripped over another player during the game. The other player was mad, but Rob apologized and kept his composure. Rob ended up playing the best game of his life!
Counterfactual	When he graduated from middle school, he started learning to play the guitar.
Model	**Generated Ending**
GPT-M + ZS	He played in bands for about a year, then got into touring with a band called The Muffs. He wrote the music for a few of The Muffs' songs. He was a very talented singer-songwriter and guitarist.
GPT-M + FT + CF	He took classes on the guitar for years to get better. He took a solo lesson and had a great time. Rob now is a famous guitarist.
GPT-M + Supervised	He accidentally tripped over the guitar during the song. The other person was mad, but Rob apologized and kept his composure. Rob ended up playing the best song of his life.
Premise	I called Carly to ask her on a date.
Initial	I got her voicemail.
Original Ending	I texted her that we should talk soon. She texted me back, asking why we should talk soon. I told her I liked her.
Counterfactual	She said that we should just stay friends which made me really sad.
Model	**Generated Ending**
GPT-M + ZS	She said that she went out and did something really stupid and that was all I could think about. I did my best to be nice to Carly by telling her how much I loved her and that I would be her best friend no matter what. There were nights when I felt like crying because I couldn't get out of bed.
GPT-M + FT + CF	I decided to watch a movie instead. After that I decided I should see a movie I had never seen before. I went to the theater and bought a ticket.
GPT-M + Sup	I texted her that we should just stay friends soon. She texted me back, asking why we should just stay friends soon. I told her I liked her.

Table 3.3: Model Outputs

second sentence is not specified, thereby encouraging generations similar to the original ending regardless of the counterfactual provided. At test time, we replace `[mask]` in the input with the counterfactual sentence $\boldsymbol{s}_2'$, and the model must generate the continuation of $(S, [s], \boldsymbol{s}_1, \boldsymbol{s}_2')$. We also use the objective from Eq equation 3.2 above to inform the model with counterfactual information during training.

3.4.2 Supervised Training (Sup)

Our dataset also provides 16,752 training instances that include human annotated rewritten endings for supervised learning. To assess whether being able to train directly on alternative endings is helpful for learning counterfactual narrative understanding, we train models on this portion of data in a supervised

manner. More concretely, the input to the model contains the full information $(S, [s], \boldsymbol{s}_1, \boldsymbol{s}'_2)$, and we train the model to maximize the log-likelihood of ground-truth rewritten endings:

$$\mathcal{L}^s(\boldsymbol{\theta}) = \log p_\theta(\boldsymbol{s}'_{3:5}|S, [s], \boldsymbol{s}_1, \boldsymbol{s}'_2). \tag{3.5}$$

where `[s]` denotes a separator token.

3.4.3 Hyperparameters

We largely follow the same training and inference setups as in Radford et al. [2018] for the GPT model and Radford et al. [2019b] for the GPT2 variants. Experiments are implemented with the text generation toolkit Texar [Hu et al., 2019]. We provide more details in Appendix 3.8.2.

3.5 Human Study of Rewritten Sentences

To assess the quality of rewritten endings, we conduct two sets of human evaluation. To give a sense of the model generation, Table 3.3 presents example outputs by a subset of representative models on two test cases.

3.5.1 Rewritten Sentence Scoring

Setup In this setting, workers from Amazon Mechanical Turk were presented 100 outputs from 14 different models. For each example, two workers were presented the original premise sentence, the original ending, the counterfactual sentence, and the rewritten ending, and asked to answer the following three questions on a 3-point Likert scale:

(1) Does the rewritten ending keep in mind details of the original premise sentence?

(2) Is the plot of the rewritten ending relevant to the plot of the original ending?

(3) Does the rewritten ending respect the changes induced by the counterfactual sentence?

In addition to evaluating the 14 models, we also provided gold human annotated counterfactual endings for the same 100 test examples to compute an expected upper bound for how models should perform. We

Model	**Pre (1)**	**Plot (2)**	**CF (3)**
GPT + ZS	1.945	1.290	1.555
GPT2-S + ZS	1.945	1.335	1.475
GPT2-M + ZS	2.435	1.615	2.045
GPT + FT	2.485	1.750	2.005
GPT2-S + FT	2.365	1.645	1.895
GPT2-M + FT	2.580	1.790	**2.070**
GPT + FT + CF	2.310	1.595	1.925
GPT2-S + FT + CF	2.310	1.640	1.850
GPT2-M + FT + CF	2.395	1.650	1.945
GPT2-S + RC + CF	2.240	2.090	1.500
GPT2-M + RC + CF	**2.780**	2.595	1.660
GPT + Sup	2.630	**2.690**	1.460
GPT2-S + Sup	2.705	2.650	1.625
GPT2-M + Sup	2.750	2.620	1.820
Human	2.830	2.545	2.520

Table 3.4: Likert scale scores for different models. The top performing model for each question is **bolded**.

present the results from this study in Table 3.4 and share key observations below. [2]

Model Size and Pretraining Data We observe that models with more parameters are better at the counterfactual rewriting task than smaller models. The GPT2-M variants consistently outperform the GPT and GPT2-S models, regardless of the objective on which the model was trained. Interestingly, however, the GPT model appears to generally outperform the GPT2-S model on the counterfactual question **(3)**, indicating that the domain on which models are pretrained does affect how adaptable their representations are to the story rewriting task.

Domain Adaptation Another pattern we notice is that fine-tuning on the ROCStories data (FT) is always helpful for increasing performance on counterfactual relevance (**CF (3)** in Table 3.4), indicating adapting to the ROCStories-style language distribution helps the model learn to produce relevant rewrites for counterfactuals, especially for models with fewer parameters. The **Plot (2)** question in Table 3.4 indicates why this might be the case, as the zero-shot models tend to produce more creative rewritings that are not at all tied to the original story. Interestingly, however, fine-tuning with the larger set of counterfactuals (CF loss) does not seem to help in rewriting endings that relate to the counterfactuals well.

[2]The average Krippendorff alpha for all three questions is 0.42 ("moderate"). [Ageeva et al., 2015])

Supervised vs. Unsupervised Learning A surprising observation is that using the dataset of labeled rewritten endings for training does not seem to help the language models learn to rewrite endings better. While the supervised models are generally able to adhere to the plot better than unsupervised methods, their new endings do not score well on question **(3)**, indicating that they may be copying the original ending or learning to paraphrase the original story ending without acknowledging the counterfactual sentence. This points to the fact that this task cannot be trivially solved by adding more paired data, since adding more data merely simplifies to having more stories in the dataset, without necessarily learning to handle counterfactuals more effectively.

3.5.2 Pairwise Model Preference

Setup We conduct a pairwise comparison between the best model (GPT2-M + Sup) with other models along the same three dimensions as in the first evaluation setting (section 3.5.1). Specifically, crowdworkers were presented outputs of a pair of systems, and asked to choose which one is better, or "equally good" or "equally bad", in terms of each of the three criteria. As in section 3.5.1, we evaluate 100 outputs of each model.

Results In Table 5.2, we present the human preference results, showing that the best model outperforms the comparison baselines in terms of consistency with premise, while being less consistently better with regards to the other two questions. Interestingly, a model that performs better on one of the evaluated dimensions often performs worse for another question, indicating plenty of room for future work in counterfactual reasoning for story rewriting.

3.6 Challenges for Automatic Metrics

To provide further insight into the performance of candidate models, we explore how different automatic metrics evaluate the produced generations.

3.6.1 Metrics

Overlap Metrics The most common metrics used in evaluating text generation are based on textual overlap between a candidate generated sequence and set of reference sequences provided by the dataset. **BLEU** [Papineni et al., 2002] is perhaps the most widely used metric in text generation, which computes the number of overlapping n-grams between the generated and reference sequences. Another commonly used metric in text generation (though originally designed for extractive summarization) is **ROUGE-L** [Lin, 2004], which measures the length of the longest common subsequence (LCS) between a candidate generation and reference. We report the performance of all models on both of these metrics.

Model-based Metrics Although BLEU and ROUGE are widely used in text generation, they use exact string matching, and thus fail to robustly match paraphrases and capture semantically-critical ordering changes. Recently, there has been a growing body of work in producing model-based metrics [Lowe et al., 2017] that use trained models and embeddings to score a sequence.

Kusner et al. [2015] proposed Word Mover's Distance, which defines the distance between two texts as the minimal cost of transforming one sequence's word embeddings to the other's. The measure finds a matching between the two texts that minimizes the total Euclidean distance between the matched word embeddings. Following Kilickaya et al. [2017], we take the negative exponential of this distance to get **Word Mover's Similarity (WMS)**. More recently, Clark et al. [2019] proposed **Sentence + Word Mover's Similarity (S+WMS)** to extend WMS for longer multi-sentence texts by using sentence representations in the minimum distance calculation in addition to word embeddings.[3]

Other recent methods use contextualized embeddings [Devlin et al., 2018a] to compute similarity between sequences. We use **BERTScore** [Zhang et al., 2019a], which computes cosine similarity between two sentences using BERT encodings. Zhang et al. show that BERTScore correlates better with human judgments than existing metrics such as BLEU, ROUGE, and other learning-based metrics. To adapt BERTScore to our task, we finetune BERT on ROCStories using the same training framework from Devlin et al. [2018a] and compute **BERT-FT** the same way as before.

[3]We follow previous work and use GloVe embeddings [Pennington et al., 2014] to represent words and the averaged word embeddings to represent sentences.

3.6.2 Human Correlation with Metrics

Recent work in text generation [Wiseman et al., 2017] and dialogue [Liu et al., 2016a] have explored the limitations of automatic metrics for text production tasks. Due to the highly semantic nature of the counterfactual rewriting task and the need to recognize subtle changes in event descriptions, we anticipate that automatic metrics would have difficulty assessing rewritten endings. To test the correlation between available evaluation metrics for long-form generation and human opinions of quality of counterfactual generations, we compute the Pearson Correlation between automatic scores and human scores for 800 validation set data points, 300 taken from the gold annotations and 100 generated from each of the 5 GPT2-M variants.[4] For each example, we use the same questions and Likert scale evaluation as in §3.5 and report the results in Table 3.6.

As expected, the automatic metrics are decently correlated with human scores for adherence to the premise sentence and plot. However, these same metrics correlate negatively with question **(3)** – adherence to the counterfactual sentence – indicating poor measurement of counterfactual understanding if they were to be reported in their typical manner (i.e., higher score indicating superior performance). Only the BERTScore metrics appear to positively correlate with human scores for counterfactual understanding, making them usable for evaluating generations across properties related to all three questions. However, the correlation is weak, and the results in Table 3.7 indicate that the BERTScore metrics are difficult to distinguish between models.

3.7 Conclusion

We introduced a new task of *Counterfactual Story Rewriting* that challenges current language understanding and generation systems with counterfactual reasoning. Our new dataset, TIMETRAVEL, provides nearly 30k counterfactual revisions to simple commonsense stories together with over 100k counterfactual sentences. We establish baseline performances of state-of-the-art neural language models with over 14 model variants with zero-shot, unsupervised and supervised settings. The empirical results demonstrate that while neural language models show promises, they generally have difficulties in rewriting the consequences of the counterfactual condition with full consistency, suggesting more focused research on integrating true reasoning

[4] We include both human annotations and model-generated outputs in this computation to encourage diversity of source.

capabilities to neural language models.

3.8 Appendix

3.8.1 Crowdsourcing Details

Quality Control Since this is a creative annotation task for crowdworkers, rather than a tagging or selection task, we need two groups of crowdworkers for two separate steps: 1) workers to create a counterfactual alternatives for the storylines, 2) workers to create a new story ending that is coherent and logically consistent with the previous context that only changes the original story arc to regain narrative consistency. Crowdworkers with more than 5000 HITs and at least a 99% acceptance rate can take our qualification test, in which we require each crowdworker to do 3 HITs before being approved for the full task. We encourage workers to submit feedback to help us improve our instructions.

Cost We pay \$0.24 to crowdworkers per instance for Step 1 and \$0.36 per instance for Step 2.

3.8.2 Training Hyperparameters

GPT2 Text is encoded with BPE using a vocabulary size of 50,257. We set the maximum sequence length to 128 tokens, which we found is large enough to contain complete stories. We use Adam optimization with an initial learning rate of 10^{-5} and a minibatch size of 2. We train the models for 10K iterations using early stopping to select the model that does the best on the validation set. At inference time, we generate using the same procedure outlined in Radford et al. [2019b]: top-k sampling with temperature set to 0.7 and k set to 40.

GPT All models follow the setting of GPT [Radford et al., 2018] that used a 12-layer decoder-only transformer with masked self-attention heads. Text is encoded with BPE using a vocabulary size of 40,000. As above, we set the maximum sequence length to 128 tokens. We use Adam optimization with an initial learning rate of 6.25^{-5}. We train the models for 10K iterations using early stopping to select the model that does the best on the validation set. We use the same generation procedure as for GPT2.

COUNTERFACTUAL - Human Judges Preferred				
Best model		Neutral	Comparator	
M+Sup	20.0	7.0	**29.5**	M+FT+CF
M+Sup	19.0	3.0	**38.5**	M+FT
M+Sup	**23.5**	14.0	4.5	M+Recon+ CF
M+Sup	26.5	5.0	**33.5**	M+ zero-shot
M+Sup	**14.0**	18.5	6.0	S+Sup
M+Sup	**18.5**	20.0	8.0	GPT + Sup
M+Sup	10.0	15.0	**52.0**	Human

PLOT - Human Judges Preferred				
Best model		Neutral	Comparator	
M+Sup	**57.5**	14.5	13.5	M+FT+CF
M+Sup	**58.5**	16.5	12.5	M+FT
M+Sup	11.5	60.0	**16.5**	M+Recon+CF
M+Sup	**63.0**	14.5	11.0	M+zero-shot
M+Sup	11.5	62.5	**12.5**	S+Sup
M+Sup	14.5	61.0	**15.0**	GPT+Sup
M+Sup	22.0	47.5	**25.0**	Human

PREMISE - Human Judges Preferred				
Best model		Neutral	Comparator	
M+Sup	**35.5**	31.0	16.5	M+FT+CF
M+Sup	**32.5**	39.5	14.0	M+FT
M+Sup	**10.5**	65.0	9.0	M+Recon+CF
M+Sup	**46.5**	29.5	13.0	M+zero-shot
M+Sup	**8.5**	71.0	7.5	S+Sup
M+Sup	**12.0**	68.0	7.5	GPT+Sup
M+Sup	12.5	59.0	**22.5**	Human

Table 3.5: Pairwise human comparison between the best model (GPT2-M + Sup) and comparison models on all three questions. "Neutral" means both are "equally good". Percentage of "equally bad" are omitted.

Metric	**(1) Prem**	**(2) Plot**	**(3) CF**
BLEU-4	**.2623**	**.6792**	**-.1804**
ROUGE-L	**.3187**	**.7484**	**-.1423**
WMS	**.2713**	**.5809**	-.0343
S+WMS	**.2789**	**.6075**	-.0538
BERT	**.2124**	**.1929**	**.1067**
BERT-FT	**.2408**	**.1847**	**.0995**

Table 3.6: Pearson correlation between automatic metrics and human scores. **Bolded** numbers are statistically significant at $p < 0.05$.

	BLEU-4	ROUGE-L	BERT	BERT-FT	WMS	W+SMS
Training: Pretrained Only				*Input:* $\boldsymbol{s}_1\boldsymbol{s}'_2$		
GPT + zero-shot	1.25	18.26	59.50	58.28	0.30	0.97
GPT2-S + zero-shot	1.28	20.27	59.62	58.11	0.33	1.09
GPT2-M + zero-shot	1.51	19.41	60.17	58.59	0.34	1.12
Training: Unsupervised + Generative				*Input:* $\boldsymbol{s}_1\boldsymbol{s}'_2$		
GPT + FT	4.20	24.55	64.38	62.60	0.56	1.48
GPT2-S + FT	3.78	24.18	64.25	62.60	0.54	1.40
GPT2-M + FT	4.09	24.08	62.23	62.49	0.53	1.42
GPT + FT + CF	3.82	24.21	64.48	62.66	0.57	1.45
GPT2-S + FT + CF	3.96	24.06	64.50	62.71	0.53	1.44
GPT2-M + FT + CF	4.00	24.38	64.31	62.59	0.48	1.33
Training: Unsupervised + Discriminative				*Input:* $\boldsymbol{s}_1\boldsymbol{s}_2\boldsymbol{y}[S]\boldsymbol{s}_1[MASK]$		
GPT2-S + Recon + CF	47.08	51.19	**63.82**	62.36	5.53	8.08
GPT2-M + Recon + CF	**76.57**	**71.35**	64.15	62.49	**18.29**	**20.87**
Training: Supervised + Discriminative				*Input:* $\boldsymbol{s}_1\boldsymbol{s}_2\boldsymbol{y}[S]\boldsymbol{s}_1\boldsymbol{s}'_2$		
GPT + Sup	80.09	75.03	64.15	62.36	20.93	23.37
GPT2-S + Sup	79.03	73.31	64.14	62.40	20.57	22.97
GPT2-M + Sup	76.63	74.42	64.06	**62.33**	19.62	22.01
Human	65.12	68.58	63.58	61.82	16.95	19.16

Table 3.7: Results on automatic metrics for the cross-product of the models and loss functions proposed in Section 5.3. **Bolded** results are closest to the human score.

Chapter 4

DeLorean: Unsupervised Backprop-based Decoding for Counterfactual and Abductive Commonsense Reasoning

The chapter discusses work originally published in [Qin et al., 2020a].

Abductive and counterfactual reasoning, core abilities of everyday human cognition, require reasoning about what might have happened at time t, while conditioning on multiple contexts from the relative past and future. However, simultaneous incorporation of past and future contexts using generative language models (LMs) can be challenging, as they are trained either to condition only on the past context or to perform narrowly scoped text-infilling. In this work, we propose DELOREAN, a new unsupervised decoding algorithm that can flexibly incorporate both the past and future contexts using only off-the-shelf, left-to-right language models and no supervision. The key intuition of our algorithm is incorporating the future through *back-propagation*, during which, we only update the internal representation of the output while fixing the model parameters. By alternating between forward and backward propagation, DELOREAN can decode the output representation that reflects both the left and right contexts. We demonstrate that our approach is general and applicable to two nonmonotonic reasoning tasks: abductive text generation and counterfactual story revision, where DELOREAN outperforms a range of unsupervised and some supervised methods, based on automatic and human evaluation.[1]

[1]Code is available at `https://github.com/qkaren/unsup_gen_for_cms_reasoning`

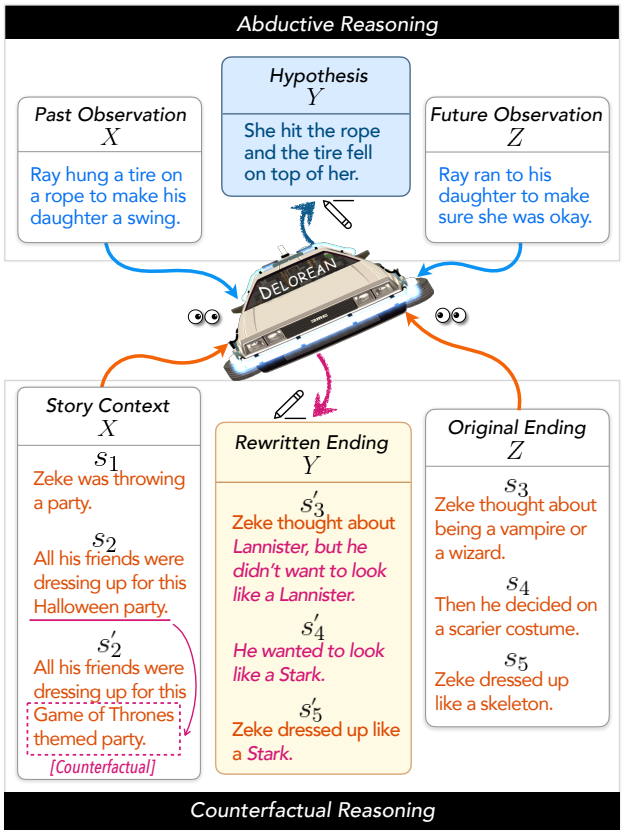


Figure 4.1: DELOREAN, our proposed method, with generated reasoning results. **Top**: the goal in abductive reasoning is to generate a hypothesis (Y) of what happened between the observed past (X) and future (Z) contexts. **Bottom**: In counterfactual reasoning, given a story context altered by a counterfactual condition, X, and the original ending Z, the goal is to generate a new ending Y which is coherent with X while remaining similar to Z. The story from TIMETRAVEL [Qin et al., 2019a] consists of five sentences. Our approach alternates forward (left-to-right) and backward (right-to-left) passes that iteratively refine the generated texts w.r.t context from each side.

4.1 Introduction

Everyday causal reasoning requires reasoning about the likely explanations to partially observable past and future (*abductive* reasoning [Peirce, 1960]) and reasoning about the alternative future based on counterfactual past (*counterfactual* reasoning). Such *nonmonotonic* reasoning requires inferring plausible but potentially defeasible conclusions from incomplete or hypothetical observations [Reiter, 1988]. While humans are remarkably good at this type of causal reasoning, developing AI systems capable of nonmonotonic reasoning for a wide range of situations describable in natural language has been a major open research question.

More concretely, with abductive reasoning, the goal is to find the most plausible explanation for in-

complete observations [Peirce, 1960]. In the top part of Figure 4.1, given the first observation that Ray is "making his daughter a swing" and the later observation that he "ran to [her] to make sure she was okay," we can hypothesize that she somehow got hurt by the swing.

In contrast, counterfactual reasoning concerns the causal changes to future events given a change in the past condition [i.e., "counterfactual condition"; Goodman, 1947]. For example, the bottom part of Figure 4.1 shows the *original* five sentence story (S_1, ..., S_5) and an alternative *counterfactual condition* given in S_2'—that instead of being a generic "Halloween party", the new counterfactual condition is that it is going to be a "Game of Thrones themed party"! Given these, the problem we want to solve is to update the future events (S_3', ..., S_5'), so that instead of "Zeke dressed up as skeleton", we have "Zeke dressed up like a Stark".[2]

Recently, two tasks and corresponding benchmarks have been introduced to tackle language-based nonmonotonic reasoning: the $\mathcal{ART}$ dataset for abductive NLG [Bhagavatula et al., 2019b], and the TIMETRAVEL dataset for counterfactual story rewriting [Qin et al., 2019a]. Both tasks are framed as conditional generation, with multiple contexts to condition on. The currently dominant paradigm for conditional text generation tasks is fine-tuning pre-trained language models (LMs), such as GPT2 [Radford et al., 2019a], on large-scale training data for supervision. However, despite the large number of training examples, supervised approaches still perform considerably worse than humans and are subject to developing superficial strategies such as repeating the observations as is or memorizing prevalent surface patters specific in the dataset [Qin et al., 2019a]. Furthermore, having to require large-scale training data for each domain and task would be utterly inefficient for broad-coverage nonmonotonic reasoning in language.

In this paper, we investigate an alternative path toward language-based nonmonotonic reasoning using pre-trained language models as is. Intuitively, both the abductive and counterfactual reasoning requires learning coherent patterns in narrative, which should be already available in large-scale pretrained language models. However, the key challenge is that most generative language models are trained to condition only on the left context, or to perform narrowly scoped text-infilling.

This paper presents DELOREAN: DEcoding for nonmonotonic LOgical REAsoNing, an unsupervised decoding algorithm that only assumes off-the-shelf left-to-right language models with no supervision. The key intuition of our algorithm is incorporating the future through back-propagation, during which, we only

[2] "Lannister" in S_3' and "Stark" in S_4' and S_5' refer to character names in the TV show, "Game of the Thrones." All the output text shown in Figure 4.1 is the actual system output from DELOREAN.

update the internal representation of the output while fixing the model parameters. More specifically, DELOREAN alternates between the forward and backward passes, where the forward pass performs left-to-right inference given the left context (roughly maximizing $P(Y|X)$ in Figure 4.1), while the backward pass instills the right constraint through right-to-left backpropagation with a task-specific loss (roughly maximizing $P(Z|XY)$). The forward and backward outputs are mixed into a single vector, from which tokens are sampled to generate the desired output. To choose the best output across iterations, we employ an unsupervised ranking step based on BERT's next sentence prediction task to measure coherence [Devlin et al., 2018a].

On both tasks, DELOREAN outperforms all other unsupervised methods in terms of both automatic metrics and human evaluation, demonstrating that nonmonotonic reasoning through conditional decoding is a promising research direction. Moreover, outputs produced by our model are judged as more coherent than those from the supervised models. In sum, our study shows that backpropagation-based decoding may enable additional future applications of unsupervised generation and reasoning.

4.2 Background

Most NLP benchmarks have focused on reasoning about information that is *entailed* from the premise. For instance, natural language inference [NLI; Bowman et al., 2015] focuses primarily on whether a hypothesis is entailed from a given premise, which means the information stated in the hypothesis is a subset of the information provided in the premise. However, it has been noted that human reasoning is often the other way, where hypotheses often contain new information that was not available in the premise, but plausibly true (but possibly defeasible with new additional context) [Johnson-Laird, 2006; Mercier and Sperber, 2017]. This type of reasoning corresponds to nonmonotonic reasoning [Kraus et al., 1990], as it contradicts the monotonicity property according to which valid arguments cannot be made invalid by adding premises. We study two tasks of that nature: abductive reasoning (§4.2.1) and counterfactual reasoning (§4.2.2).

4.2.1 Abductive Reasoning

Abductive reasoning aims at finding the most likely explanation to partial observations [Peirce, 1960]. It has a central role in the human ability to "read between the lines," and is crucial for language acquisition [Andersen, 1973], understanding sentences in discourse [Hobbs et al., 1993], and many more. Despite the

importance, however, relatively little focus has been given to it in NLP research.

Recently, 122019bBhagavatula et al.Bhagavatula, Le Bras, Malaviya, Sakaguchi, Holtzman, Rashkin, Downey, Yih, and Choi) propose the abductive reasoning task. Given two observations, the goal is to determine the most likely explanation of what happened in-between. The dataset introduced for the task, $\mathcal{ART}$, consists of 20k observations derived from the first and last sentence of stories in the ROCStories dataset [Mostafazadeh et al., 2016b]. We focus on the abductive NLG setup introduced in the paper, which is framed as a conditional generation task where a plausible explanation to the observations must be generated using language. The authors reported the performance of several pre-trained LM-based baselines and showed promises and limitations of such approaches.

4.2.2 Counterfactual Reasoning

Counterfactual reasoning aims at inferring alternative past events that could have happened given a certain change in conditions [Goodman, 1947; Starr, 2019]. While counterfactual reasoning plays an important role in AI systems [Isard, 1974; Ginsberg, 1986], it requires causal reasoning abilities, which are arguably absent from current association-based AI [Pearl and Mackenzie, 2018]. While there has been work on counterfactual reasoning in NLP, including recognizing counterfactuals in text [Son et al., 2017a], and improving the performance of NLP tasks using *counterfactual learning* [Lawrence et al., 2017; Lawrence and Riezler, 2018b], it remains a major research challenge.

Recently, 1562019aQin et al.Qin, Bosselut, Holtzman, Bhagavatula, Clark, and Choi) introduce the task of counterfactual story generation. Given a 5-sentence original story, and an alternative context in which the second sentence of the story was altered by a counterfactual, the task is to generate a new 3-sentence story ending that addresses the alternative beginning while minimally editing the original ending. The associated TIMETRAVEL dataset is based on fictional narratives from ROCStories, for which counterfactual contexts and alternative endings are crowdsourced, yielding 29,849 problem instances. 1562019aQin et al.Qin, Bosselut, Holtzman, Bhagavatula, Clark, and Choi) report several baseline performances, and find that models based on pre-trained LMs produce output that recognize the counterfactual, but generated endings which deviated considerably from the original storyline. In contrast, in the supervised setup, models optimize the easier of the two goals and generate endings that are overly similar to the original endings.

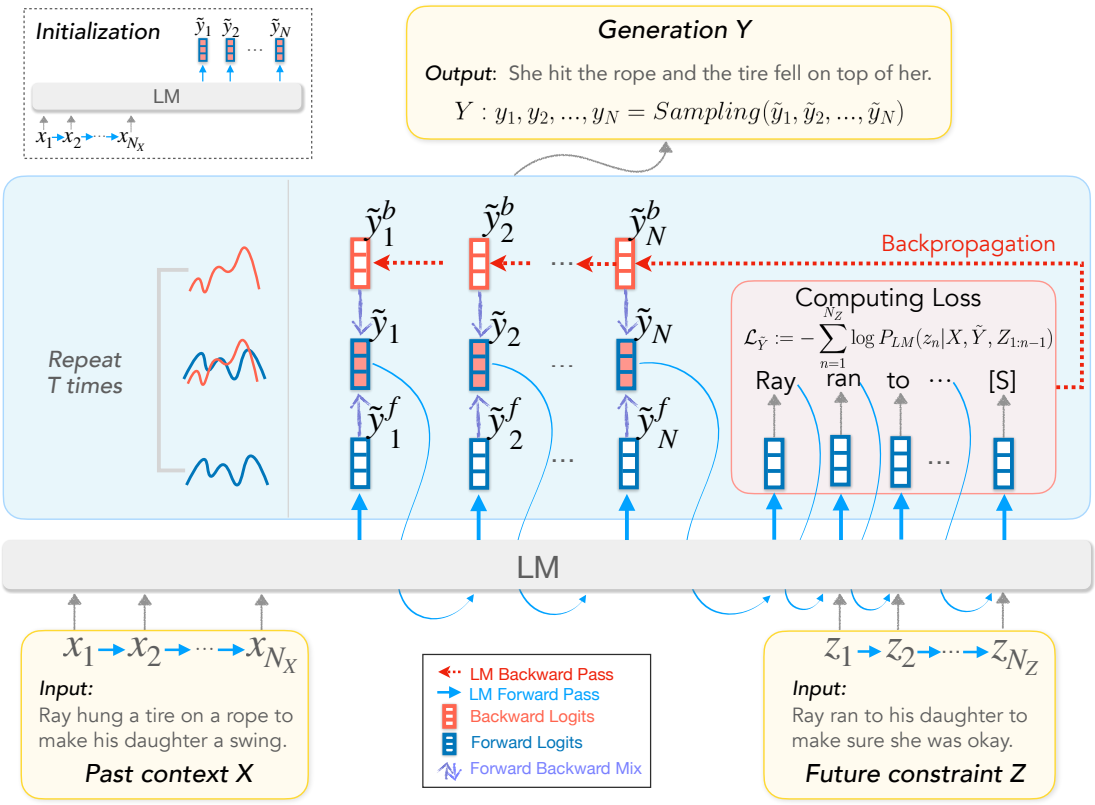


Figure 4.2: Illustration of the DELOREAN decoding procedure, using abductive reasoning as an example. At initialization (upper-left box), the language model (LM) initializes the logits $\tilde{Y} = \{\tilde{y}_1, \ldots, \tilde{y}_N\}$ of the hypothesis by reading the past context X and generating a continuation with regular decoding. At each forward-backward iteration, we compute the task-specific **loss** $\mathcal{L}_{\tilde{Y}}$ of the logits based on the future constraint Z (red box). The **backward pass** then performs back-propagation and produces the backward logits $\tilde{Y}^b = \{\tilde{y}_1^b, \ldots, \tilde{y}_N^b\}$. In the subsequent **forward pass**, for each step n, we compute the forward logits $\tilde{y}_n^f$ conditioning on the preceding logits $\tilde{y}_{n-1}$, and then mix it with the respective backward logits to produce the new logits $\tilde{y}_n$ at step n.

4.3 The DELOREAN Approach

Humans make inferences based on available information and refine them when new information arrives. Since currently available pre-trained LMs generate text by sequentially predicting the next token from left to right, they are incapable of conditioning on future constraints. Therefore, we propose DELOREAN: an unsupervised backprop-based decoding algorithm, which is summarized in Algorithm 2, illustrated in Figure 4.2, and detailed below. DELOREAN intermittently refines the predictions to cohere with either the context or the constraints (Section 4.3.1). The candidate generations are then ranked by coherence (Section 4.3.2).

Algorithm 2 COLD Decoding

Input: Pre-trained language model (LM)
Context X
Future constraint Z

Initialize logits $\tilde{Y}^{(0)}$
Initialize Y_s, list of candidate generations
for $t \leftarrow 1$ to T **do**
  // Backward pass
  for $n \leftarrow N$ to 1 **do**
    Compute backward logits $\tilde{\boldsymbol{y}}_n^{\text{b}}$, Eq.equation 4.1
  end for
  // Forward pass
  for $n \leftarrow 1$ to N **do**
    Compute forward logits $\tilde{\boldsymbol{y}}_n^{\text{f}}$, Eq.equation 4.2
    Mix forward and backward logits, Eq.equation 4.3
  end for
  Sample candidate Y from logits $\tilde{Y}$ and add to Y_s
end for
Rank Y_s by coherence

Output: The most coherent generated text Y from Y_s

4.3.1 Decoding Strategy

Given context text X, the goal is to generate continuation text $Y = (y_1, \ldots, y_N)$, such that Y satisfies certain constraints according to the reasoning tasks, usually defined based on another context Z (see Figure 4.1; we discuss the task-specific constraints in the respective task sections).

The proposed approach interleaves two procedures, namely, *forward* and *backward*, that produce and iteratively refine the generation, for a predefined number of iterations T. In particular, the *forward* pass ensures the generated text is a fluent continuation of the context X, while the *backward* pass informs the model about the constraint and steers the generation to satisfy it.

As detailed below, the backward pass uses gradient descent to update the generation Y. However, Y is a discrete text that is not differentiable. Instead, throughout the algorithm, we maintain a soft representation of the sequence $\tilde{Y} = (\tilde{\boldsymbol{y}}_1, \ldots, \tilde{\boldsymbol{y}}_N)$, where $\tilde{\boldsymbol{y}}_n \in \mathbb{R}^V$ represents the logits of the n-th token and V is the vocabulary size. After the logits are refined over multiple iterations of the forward and backward passes, we generate discrete text at each step by sampling from $y_n \sim \text{softmax}(\tilde{\boldsymbol{y}}_n/\tau)$, where $\tau > 0$ is the temperature.

We start by initializing the logits before the first iteration, $\tilde{Y}^{(0)} = (\tilde{y}_1^{(0)} \ldots \tilde{y}_N^{(0)})$, by feeding the context X into the LM and greedily decoding N continuation tokens.

Backward The backward pass uses gradient backpropagation to update the generation with respect to the constraint. Specifically, we express the task-specific constraint as a loss function $\mathcal{L}(X, \tilde{Y}^{(t-1)}, Z)$ that evaluates how well the generation Y (approximated with the soft representation $\tilde{Y}$) obeys the constraint (see the subsequent sections for concrete instantiations of the loss). The goal of this pass is thus to minimize the loss w.r.t the generation. Specifically, at iteration t, for each step n in the generation, we update its logits with:

$$\tilde{\boldsymbol{y}}_n^{(t),\text{b}} = \tilde{\boldsymbol{y}}_n^{(t-1)} - \lambda \cdot \nabla_{\tilde{\boldsymbol{y}}_n} \mathcal{L}(X, \tilde{Y}^{(t-1)}, Z), \tag{4.1}$$

where $\nabla_{\tilde{\boldsymbol{y}}_n} \mathcal{L}(X, \tilde{Y}^{(t-1)}, Z)$ is the gradient of the constraint-informed loss $\mathcal{L}$ w.r.t the n-th logits, and $\lambda \in \mathbb{R}$ is the step size. In practice, we may repeat the gradient updates multiple times in a single pass.

Forward The forward pass ensures that Y is fluent and coherent with the preceding context X. At iteration t, for a particular step n, we compute the forward logits with the LM:

$$\tilde{\boldsymbol{y}}_n^{(t),\text{f}} = \text{LM}(X, \tilde{Y}_{1:n-1}^{(t)}). \tag{4.2}$$

We then *mix* the nth-step forward and backward logits to get the final logits of iteration t:

$$\tilde{\boldsymbol{y}}_n^{(t)} = \gamma \cdot \tilde{\boldsymbol{y}}_n^{(t),\text{f}} + (1 - \gamma) \cdot \tilde{\boldsymbol{y}}_n^{(t),\text{b}}, \tag{4.3}$$

where $0 < \gamma < 1$ is the mixing weight. The resulting logits $\tilde{\boldsymbol{y}}_n^{(t)}$ are then fed to the LM to compute the forward logits at the $(n+1)$th step (Eq.4.2). This way, information from the backward pass is integrated into the left-to-right generation process to produce text that is informed by the constraint.

We pre-define the number of tokens N required by the backward pass, but we allow the forward pass to generate more than N tokens if those are needed to obtain complete sentences. In that case, we set the logits of the extra tokens to the forward logits, without mixing: $\tilde{\boldsymbol{y}}_n^{(t)} = \tilde{\boldsymbol{y}}_n^{(t),\text{f}}$ for $n > N$. We then prune any trailing tokens in the sampled text to get complete sentences.

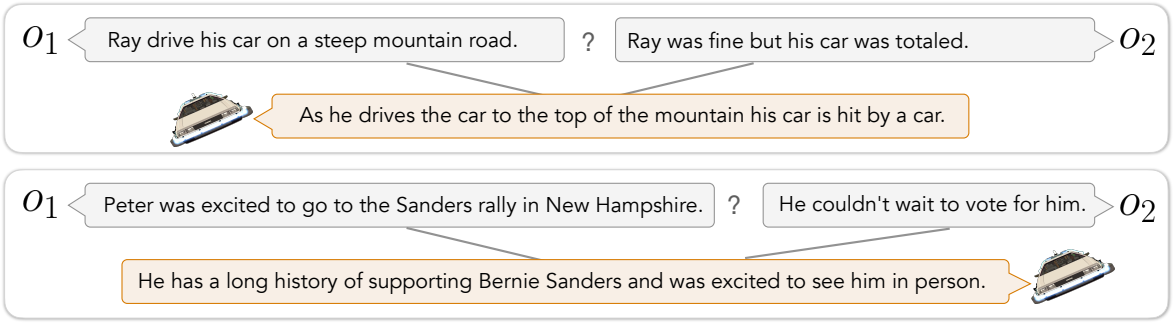


Figure 4.3: Examples of generated hypotheses on three abductive reasoning cases. Given observations O1 and O2, DELOREAN generates a hypothesis explaining the observations.

4.3.2 Ranking

The output of the decoding step is a list of candidate generations for each iteration: $Y_s = \{Y^{(t)}|t = 1, ..., T\}$. We further use an unsupervised approach to rank and pick the best sample as the final output. Specifically, we take advantage of the BERT model, which was pre-trained with a next-sentence prediction (NSP) objective. Given two sentences A and B, we use NSP to compute the likelihood of B following A as a proxy for coherence:

$$c(A, B) = \text{BERT\_NSP}(A, B), \tag{4.4}$$

where $c(\cdot, \cdot)$ denotes the coherence score. This score is used to evaluate the quality of a given candidate continuation Y by measuring (1) its compatibility with the subsequent text of the context X, (2) the internal consistency of Y if it consists of multiple sentences, and (3) the compatibility of Y with its right-side text when it is applicable.

4.4 Task 1: Abductive Reasoning

Each instance in the $\mathcal{ART}$ dataset consists of two observations O_1, O_2 and a hypothesis H that explains the two observations. These inputs naturally map to X, Z and Y in our framework. Formally, the abductive generation task aims to maximize $P(Y|X, Z)$ – i.e. models must consider both left and right contexts (X and Z) jointly.

Model	BLEU-4	ROUGE-L	BERT
Supervised			
Sup	3.46	25.60	49.38
+*COMET-Emb*	4.06	26.06	49.71
Unsupervised			
Zero-Shot$_X$	0.65	14.99	39.36
Zero-Shot$_{ZX}$	0.53	14.23	40.03
Zero-Shot$_X$-Ranked	0.87	16.76	41.58
Zero-Shot$_{ZX}$-Ranked	0.98	17.25	41.93
DELOREAN	**1.38**	**18.94**	**42.86**
Human	8.25	30.40	53.30

Table 4.1: Automatic evaluation results on the abductive task, using the test set of $\mathcal{ART}$.

4.4.1 Task Setup

Constraints We maximize Z given $X\tilde{Y}$ by defining the loss function as the cross-entropy loss of generating Z given $X\tilde{Y}$ with the LM:[3]

$$\mathcal{L}(X, \tilde{Y}, Z) := -\sum_{n=1}^{N_Z} \log P_{\text{LM}}(z_n | X, \tilde{Y}, Z_{1:n-1}), \tag{4.5}$$

where $P_{\text{LM}}(a_j | a_{1:j-1})$ is the likelihood of generating token a_j given the preceding text $a_{1:j-1}$.

Following the earlier study of the task [Bhagavatula et al., 2019b], we also prepend Z to X to "leak" the future information to the LM. That is, we replace X with $Z\langle e\rangle X$ in the above equation, where $\langle e\rangle$ denotes a special end-of-text token. However, the comparisons with respective baselines below show the prepended Z is minor to the performance.

Ranking We rank candidates by the overall coherence after inserting Y in between X and Z:

$$\text{ranking\_score}(Y) = c(XY, Z) + c(X, YZ). \tag{4.6}$$

Hyperparameters We use GPT2-345M [Radford et al., 2019b] as the pre-trained LM for all models. We use the $\mathcal{ART}$ development set to select hyperparameters. We use greedy decoding for our method and top k decoding [Fan et al., 2018] ($k = 40, \tau = 0.7$) for our baselines. Other hyperparameters are outlined in

[3]Note that this is applied to each prefix of $\tilde{Y}$, although some of them are not complete sentences.

Appendix **??**.

4.4.2 Experimental Setup

Baselines We compare our method against baselines from 122019bBhagavatula et al.Bhagavatula, Le Bras, Malaviya, Sakaguchi, Holtzman, Rashkin, Downey, Yih, and Choi). The unsupervised baselines use a pre-trained GPT-2 model to generate Y given a prompt text—either the observation X alone (Zero-Shot$_X$) or $Z\langle e\rangle X$ (Zero-Shot$_{ZX}$). The supervised method (Sup) follows the same input format as Zero-Shot$_{ZX}$, but fine-tunes GPT-2 on the $\mathcal{ART}$ training set. Finally, our knowledge-informed baseline (+*COMET-Emb*) further augments the representation of Sup with knowledge from COMET [Bosselut et al., 2019].

To separately study the contribution of our decoding strategy and ranking component, we also report the performance of ranking the baseline outputs. Specifically, we let each baseline generate 20 candidates and rank them by coherence (Eq. 4.6).[4]

4.4.3 Results

Automatic Evaluation We report the same metrics as 122019bBhagavatula et al.Bhagavatula, Le Bras, Malaviya, Sakaguchi, Holtzman, Rashkin, Downey, Yih, and Choi): BLEU-4 [Papineni et al., 2002], ROUGE-L [Lin, 2004] and BERTSCORE [Zhang et al., 2019b] (with the *bert-base-uncased* model). The results in Table 4.1 show that DELOREAN performs best among the unsupervised systems across all metrics. We also note that our ranking step improves both the performance of our model and that of the zero-shot baselines.

Human Evaluation We conduct two sets of human evaluations on 100 test examples using crowdworkers from Amazon Mechanical Turk. In the scoring setting, presented in Table 4.2, workers were presented a pair of observations (X and Z) and a generated hypothesis Y, and asked to rate the coherence of the hypothesis with respect to the observation X (X-Y), the observation Z (Y-Z), and both (X-Y-Z), on a 4-point Likert scale. In the pairwise comparison setting, presented in Table 4.3, workers were presented the outputs from a pair of systems (DELOREAN and baseline) and asked to choose the better output in terms of the same

[4]We tried ablating the ranking component from our method in preliminary experiments, and found that ranking is essential to obtaining good performance. By adding ranking to our baselines, we assess the contribution of our decoding strategy.

Model	X-Y	Y-Z	X-Y-Z
Supervised			
Sup	0.510	0.375	0.314
+*COMET-Emb*	0.466	0.342	0.286
Unsupervised			
Zero-Shot$_{ZX}$	0.233	0.103	0.108
Zero-Shot$_{X}$-Ranked	0.478	0.208	0.195
Zero-Shot$_{ZX}$-Ranked	0.474	0.238	0.236
DELOREAN	**0.522**	**0.325**	**0.297**
Human	0.879	0.823	0.783

Table 4.2: Human calibration results on test set of $\mathcal{ART}$. All scores are normalized to $[0, 1]$.

Overall - Human Judges Preferred				
Our model		Neutral	Comparator	
DELOREAN	21%	43%	**36%**	Sup
DELOREAN	25%	44%	**31%**	+*COMET-Emb*
DELOREAN	**23%**	62%	15%	Zero-Shot$_{X}$-Ranked
DELOREAN	**27%**	50%	23%	Zero-Shot$_{XZ}$-Ranked
DELOREAN	3%	11%	**86%**	Human

Table 4.3: Human pairwise comparison results on the test set of $\mathcal{ART}$, between COLD and each of the baselines, by jointly considering all 3 criteria from Table 4.2. "Neutral" means "equally good/bad".

coherence criteria. Each example was labeled by 3 workers.[5]

In both evaluation setups, our method substantially outperform the unsupervised baselines, achieving a relative improvement of $36\% - 215\%$ with respect to Y-Z coherence. Our method also outperform the supervised methods with respect to X-Y coherence (Table 4.2), and achieve competitive performance in the pairwise comparison (Table 4.3). Again, the ranking component contributes to increasing performance for the zero-shot baselines. Finally, the large performance gap between the methods and human-written explanations stresses the difficulty of this reasoning task and warrants future research.

Qualitative Analysis Figure 4.3 presents two example outputs produced by DELOREAN. We can see our approach generates reasonable hypotheses by taking into account both the past and future contexts. For instance, in the first example, the future observation (O2) "car was totaled" indicates that Ray had a car accident, which is correctly captured in the generated hypothesis "car is hit by a car".

[5]The average inter-rater agreement measured by Fleiss' $\kappa = 0.44$ ("moderate agreement") [Fleiss, 1971].

	BLEU_4	ROUGE_L	BERT
Supervised + Discriminative			
Sup+Disc	75.71	72.72	62.39
Unsupervised+ Discriminative			
Recon+CF	**75.92**	**70.93**	62.49
Unsupervised			
FT	4.06	24.09	62.55
FT+CF	4.02	24.35	62.63
Pretrained-only			
Zero-Shot$_{s_1 s_2'}$	1.74	21.41	59.31
Zero-Shot$_{s_1 s_2'}$-Ranked	2.26	25.81	60.07
DELOREAN	21.35	40.73	**63.36**
Human	64.93	67.64	61.87

Table 4.4: Automatic evaluation results of counterfactual story rewriting, on the test set of TIMETRAVEL.

4.5 Task 2: Counterfactual Reasoning

Given an original story ending Z of story context X^{ori}, and a counterfactual condition X that changes X^{ori} to invalidate Z (see Fig. 4.1), the task is to generate a new story ending Y that minimally edits the original ending Z to regain coherence with the counterfactual condition X [Qin et al., 2019a].

4.5.1 Task Setup

Constraints The constraint we enforce is that Y is close to Z (i.e., minimal edits). We impose this constraint by minimizing their KL divergence:

$$\mathcal{L}(X, \tilde{Y}, Z) := \text{KL}\left(Z \| \text{softmax}(\tilde{Y}/\tau)\right), \tag{4.7}$$

where, with a slight abuse of notation, Z is the one-hot distribution of the tokens in the original ending. That is, we encourage the generated logits to recover the original ending.

Ranking We rank the candidates based on both their coherence with the context, as well as the internal coherence between the multiple sentences of each candidate (rewritten ending, consists of 3 sentences).

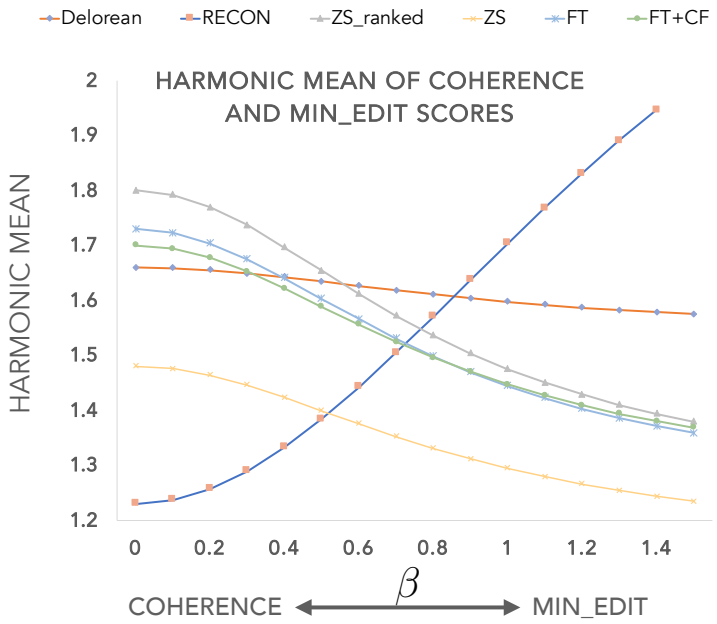


Figure 4.4: Human calibration results for counterfactual generation in terms of weighted harmonic mean of coherence and min-edit, $H_\beta = \frac{(1+\beta^2)\cdot\text{coherence}\cdot\text{min\_edit}}{\beta^2\cdot\text{coherence}+\text{min\_edit}}$, as a function of the scaling factor β. Low β values assign more weight to coherence, and high β values emphasize more on min-edit.

More concretely, given a candidate Y, we compute the aggregated coherence score:

$$\text{ranking\_score}(Y) = c(X, Y) + \sum_{s=1}^{S-1} c(Y[s], Y[s+1]), \tag{4.8}$$

where each candidate has S sentences (here, $S = 3$) and $Y[s]$ denotes the sth sentence.

Hyperparameters We largely follow the same setting as in the abductive reasoning task, but tune hyperparameters on the TIMETRAVEL development set. Deviations from these settings are outlined in Appendix **??**.

4.5.2 Experimental Setup

Baselines We compare our method with baselines from 1562019aQin et al.Qin, Bosselut, Holtzman, Bhagavatula, Clark, and Choi). The zero-shot baseline uses the pre-trained GPT-2 model to generate Y as a continuation to the counterfactual condition X. It is the most apt comparison to our method which also doesn't require additional supervision. We also experiment with two baselines that fine-tune GPT-2 on the original story $X^{ori}Z$ to fit the model to the story domain, either with an LM objective (FT) or a tailored

Coherence - Human Judges Preferred				
Our model		Neutral	Comparator	
DELOREAN	**25%**	58%	17%	Sup+Disc
DELOREAN	**23%**	70%	7%	Recon+CF
DELOREAN	22%	48%	**30%**	FT
DELOREAN	18%	60%	**22%**	Zero-Shot$_{s_1 s_2'}$
DELOREAN	27%	42%	**31%**	Zero-Shot$_{s_1 s_2'}$-Ranked
DELOREAN	10%	29%	**61%**	Human

Min-Edits - Human Judges Preferred				
Our model		Neutral	Comparator	
DELOREAN	4%	17%	**79%**	Sup+Disc
DELOREAN	1%	14%	**85%**	Recon+CF
DELOREAN	**21%**	76%	3%	FT
DELOREAN	**28%**	71%	1%	Zero-Shot$_{s_1 s_2'}$
DELOREAN	**37%**	56%	7%	Zero-Shot$_{s_1 s_2'}$-Ranked
M+Sup	8%	22%	**70%**	Human

Table 4.5: Human pairwise comparison results on the counterfactual task, between our best model and each baseline with respect to coherence and min-edits.

conditional objective that encourages minimal edits of Z (`Recon+CF`).[6] Finally, we report the performance of a supervised baseline (Sup), in which GPT-2 is fine-tuned to produce the gold Y from $X^{ori}Z$ and X.

4.5.3 Results

Automatic Evaluation Following 1562019aQin et al.Qin, Bosselut, Holtzman, Bhagavatula, Clark, and Choi), we report BERTSCORE [Zhang et al., 2019b], which was shown to best correlate with human judges' notion of counterfactual coherence, and BLEU-4 and ROUGE-L, which better measure minimum-edits. We find that the discriminative baselines achieve the highest degree of plot fidelity. Meanwhile, DELOREAN achieves the highest BERTSCORE for counterfactual coherence.

Human Evaluation We repeat the human evaluation setup from Section 4.4.3. Presented with the original story, the counterfactual condition X, and the generated ending Y, workers were asked to judge (1) the *coherence* of Y with respect to the X; and (2) to what extent the generated ending *minimally-edits* the original ending.[7] In order to judge both criteria, we report the weighted harmonic mean H_β of these scores

[6] See 1562019aQin et al.Qin, Bosselut, Holtzman, Bhagavatula, Clark, and Choi) for more details.

[7] Fair inter-rater agreement with Fleiss' $\kappa = 0.34$

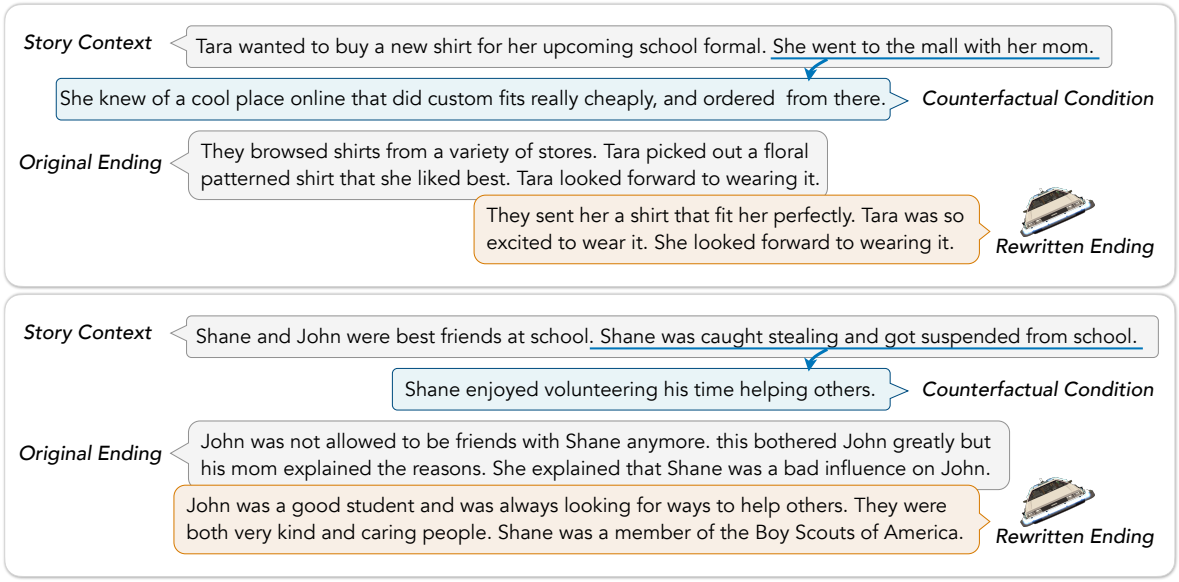


Figure 4.5: Examples of generated story endings on three counterfactual reasoning cases. Given a story context, a counterfactual condition, and a original ending, DELOREAN generates a rewritten ending which is coherent with the counterfactual condition and is similar to the original ending.

across a range of weights β (Figure 4.4).

Our results show that DELOREAN is the only model that maintains a consistent balance between *coherence* (1.66) and *minimal edits* (1.54). While the ranking-augmented zero-shot model produces the most coherent endings (*coherence* = 1.8), it deviates from the original ending. As β is increased (i.e., increasing importance of *minimal edits*), its weighted performance drops considerably, indicating it cannot generate new endings that follow the original plot of the story (*min-edit* = 1.25). Conversely, `Recon+CF` generates stories that are faithful to the original endings, but are far less coherent with the counterfactual condition (*coherence* = 1.23). Through human annotation, we found that `Recon+CF` copies the original ending word-for-word in a 84% of cases.

The pairwise comparison results in Table 5.2 parallel these observations. DELOREAN significantly outperforms the discriminative approaches (`Recon+CF` and `Sup+Disc`) in coherence, while falling short of the Zero-shot re-ranked baselines. In minimal edits, this pattern is flipped with our approach outperforming Zero-shot baselines considerably and losing to the discriminative baselines.

Qualitative Analysis Figure 4.5 provides two example results for counterfactual story rewriting by DELOREAN. The approach successfully captures the causal relations between events and properly rewrites the

endings with minimal edits. For instance, in the first example, given the counterfactual condition that "Tara ordered a shirt online" (as opposed to the original "went to mall"), the rewritten ending is about "sent shirt" to Tara (as opposed to the original "browsed from stores"). The last sentence of the original ending "She looked forward to wearing it" is correctly preserved as it is coherent with the counterfactual condition.

4.6 Related Work

Unsupervised text generation. Unsupervised approaches are often applied to problems that copy information from a source text into decoded text. Unsupervised paraphrasing requires repeating this information [Miao et al., 2019b; Bao et al., 2019], as does translation, but with a bilingual transformation [Artetxe et al., 2017; Lample et al., 2018]. In summarization there is an additional task to select a subset of the original text [Baziotis et al., 2019; Schumann et al., 2020; West et al., 2019]. In cases where information is mostly copied from the original, auto-encoding objectives can ensure the correct information is captured [Bao et al., 2019; Baziotis et al., 2019; Artetxe et al., 2017]. This work tackles problems where generation is more open-ended. Rather than reproducing information from the prompt, generations should agree with and expand on it, making autoencoding less applicable.

Controllable language generation. Earlier approaches for controllable generation involved preserving the content of text while changing it along discrete dimensions, such as theme, sentiment, or style [Koncel-Kedziorski et al., 2016a; Hu et al., 2017; Ficler and Goldberg, 2017; Shen et al., 2017; Lample et al., 2019a]. Recent works such as Grover [Zellers et al., 2019b] and CTRL model [Keskar et al., 2019] used these ideas to augment transformer language models that can condition on structured metadata such as source, domain, etc. The Plug & Play model [PPLM; Dathathri et al., 2019] controls topic and sentiment in an approach similar to ours that involves forward and backward passes to update token distributions. However, PPLM relies on trained attribute discriminators for supervision, while our method is unsupervised. While these models are restricted to specific dimensions, often with pre-defined values, our model can adjust to any open-ended textual constraint. Perhaps the most similar work in that aspect is the "text infilling" models, which, however, are in a more narrow setting by filling only a relatively short text span [Devlin et al., 2018a; Zhu et al., 2019; Donahue et al., 2020], and more restrictive due to the reliance on an extra right-to-

left language model [Sun et al., 2017] or a pre-specified generation length [Zeldes et al., 2020, which is not publicly available].

Reasoning about narratives. A prominent resource from recent years is the RocStories corpus [Mostafazadeh et al., 2016c], consisting of 98K crowdsourced 5-sentence everyday life stories. It was used for the story cloze task whose goal was to predict the story ending from its first 4 sentences, but gained popularity and became the base of additional benchmarks [Rashkin et al., 2018]. Additional related work includes "script knowledge", i.e. learning about prototypical series of events [Schank and Abelson, 1977; Chambers and Jurafsky, 2008; Pichotta and Mooney, 2014], temporal commonsense [Granroth-Wilding and Clark, 2016; Li et al., 2018], and modeling pre- and post- conditions of events [Roemmele et al., 2011a; Sap et al., 2019; Bosselut et al., 2019]. Qin et al. [2019b] studied conversation modeling that reads and connects the dots of events in related documents. Finally, a recent line of work explores counterfactual questions in reading comprehension [Huang et al., 2019a; Tandon et al., 2019], but instantiates the problem of counterfactual reasoning as a multiple choice task.

4.7 Conclusion

We presented DELOREAN, an unsupervised LM-based approach to generate text conditioned on past context as well as future constraints, through forward and backward passes considering each condition. We demonstrated its effectiveness for abductive and counterfactual reasoning, on which it performed substantially better than unsupervised baselines. Our method is general and can be easily adapted for other generative reasoning tasks.

4.8 Appendix

Abductive Reasoning We set the hypothesis length $N = 15$ in the backward pass and allow the forward pass to generate N*2 tokens for complete sentences. We run $T = 20$ forward-backward iterations, with each backward pass performing 20 gradient updates using a small step size $\lambda = 0.0003$. The mixing weight of forward/backward logits is $\gamma = 0.88$. We use greedy decoding to produce a single candidate at each

iteration T.

Counterfactual Reasoning We use a step size $\lambda = 0.0004$ in backward pass and a mixing weight $\gamma = 0.92$. One difference from the abductive task is that, here we vary the number of forward-backward iterations within $\{5, 10\}$ and the number of backward gradient updates within $\{5, 8, 10, 15\}$. Each configuration produces one candidate at the end of the algorithm. So for each example, we produce 8 candidates for ranking. We found such a generation-ranking protocol gives better performance on the counterfactual task.

Since we need to generate 3 sentences, the number of tokens N is relatively large. For the effectiveness of backpropagation and forward computation, we split the generation into 3 segments, one for each sentence, and perform the forward-backward passes for each segment separately. A sentence that was generated for the ith segment, is then appended to the context when generating the i+1 segment.

Part III

Integration of Knowledge and Logic in Neural Language Model Reasoning

Chapter 5

Conversing by Reading: Contentful Neural Conversation with On-demand Machine Reading

The chapter discusses work originally published in [Qin et al., 2019b].

Although neural conversation models are effective in learning *how* to produce fluent responses, their primary challenge lies in knowing *what* to say to make the conversation *contentful* and non-vacuous. We present a new end-to-end approach to contentful neural conversation that jointly models response generation and on-demand machine reading. The key idea is to provide the conversation model with relevant long-form text *on the fly* as a source of external knowledge. The model performs QA-style reading comprehension on this text in response to each conversational turn, thereby allowing for more focused integration of external knowledge than has been possible in prior approaches. To support further research on knowledge-grounded conversation, we introduce a new large-scale conversation dataset grounded in external web pages (2.8M turns, 7.4M sentences of grounding). Both human evaluation and automated metrics show that our approach results in more contentful responses compared to a variety of previous methods, improving both the informativeness and diversity of generated output.

Figure 5.1: Users discussing a topic defined by a Wikipedia article. In this real-world example from our Reddit dataset, information needed to ground responses is distributed throughout the source document.

5.1 Introduction

While end-to-end neural conversation models [Shang et al., 2015; Sordoni et al., 2015; Vinyals and Le, 2015; Serban et al., 2016; Li et al., 2016a; Gao et al., 2019a, etc.] are effective in learning *how* to be fluent, their responses are often vacuous and uninformative. A primary challenge thus lies in modeling *what* to say to make the conversation contentful. Several recent approaches have attempted to address this difficulty by conditioning the language decoder on external information sources, such as knowledge bases [Agarwal et al., 2018; Liu et al., 2018a], review posts [Ghazvininejad et al., 2018; Moghe et al., 2018], and even images [Das et al., 2017; Mostafazadeh et al., 2017]. However, empirical results suggest that conditioning the decoder on rich and complex contexts, while helpful, does not on its own provide sufficient inductive bias for these systems to learn how to achieve deep and accurate integration between external knowledge and response generation.

We posit that this ongoing challenge demands a more effective mechanism to support on-demand knowledge integration. We draw inspiration from how humans converse about a topic, where people often search and acquire external information as needed to continue a meaningful and informative conversation. Fig-

ure 5.1 illustrates an example human discussion, where information scattered in separate paragraphs must be consolidated to compose grounded and appropriate responses. Thus, the challenge is to connect the dots across different pieces of information in much the same way that *machine reading comprehension (MRC)* systems tie together multiple text segments to provide a unified and factual answer [Seo et al., 2017, etc.].

We introduce a new framework of end-to-end conversation models that jointly learn response generation together with on-demand machine reading. We formulate the reading comprehension task as document-grounded response generation: given a long document that supplements the conversation topic, along with the conversation history, we aim to produce a response that is both conversationally appropriate and informed by the content of the document. The key idea is to project conventional QA-based reading comprehension onto conversation response generation by equating the conversation prompt with the question, the conversation response with the answer, and external knowledge with the context. The MRC framing allows for integration of long external documents that present notably richer and more complex information than relatively small collections of short, independent review posts such as those that have been used in prior work [Ghazvininejad et al., 2018; Moghe et al., 2018].

We also introduce a large dataset to facilitate research on knowledge-grounded conversation (2.8M turns, 7.4M sentences of grounding) that is at least one order of magnitude larger than existing datasets [Dinan et al., 2019; Moghe et al., 2018]. This dataset consists of real-world conversations extracted from Reddit, linked to web documents discussed in the conversations. Empirical results on our new dataset demonstrate that our full model improves over previous grounded response generation systems and various ungrounded baselines, suggesting that deep knowledge integration is an important research direction.[1]

5.2 Task

We propose to use factoid- and entity-rich web documents, e.g., news stories and Wikipedia pages, as external knowledge sources for an open-ended conversational system to ground in.

Formally, we are given a conversation history of turns $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M)$ and a web document $D = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N)$ as the knowledge source, where $\boldsymbol{s}_i$ is the ith sentence in the document. With the pair (X, D), the system

[1]Code for reproducing our models and data is made publicly available at `https://github.com/qkaren/converse_reading_cmr`.

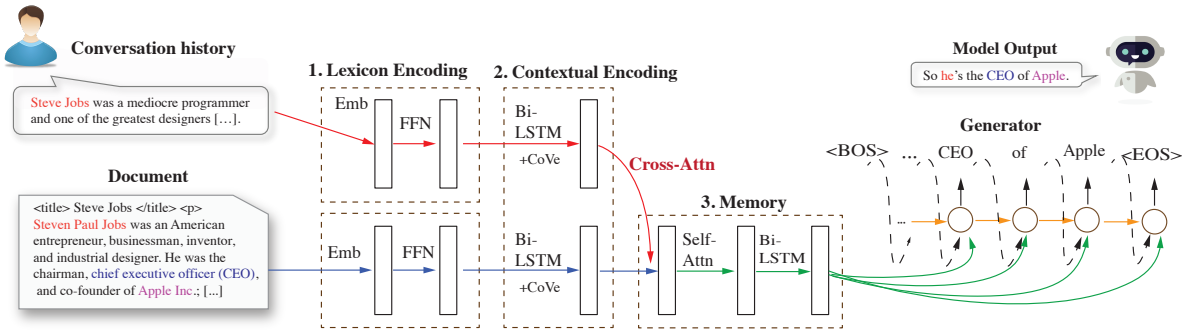


Figure 5.2: Model Architecture for Response Generation with on-demand Machine Reading: The first blocks of the MRC-based encoder serve as a lexicon encoding that maps words to their embeddings and transforms with position-wise FFN, independently for the conversation history and the document. The next block is for contextual encoding, where BiLSTMs are applied to the lexicon embeddings to model the context for both conversation history and document. The last block builds the final encoder memory, by sequentially applying cross-attention in order to integrate the two information sources, conversation history and document, self-attention for salient information retrieval, and a BiLSTM for final information rearrangement. The response generator then attends to the memory and generates a free-form response.

needs to generate a natural language response $\boldsymbol{y}$ that is both conversationally appropriate and reflective of the contents of the web document.

5.3 Approach

Our approach integrates conversation generation with on-demand MRC. Specifically, we use an MRC model to effectively encode the conversation history by treating it as a question in a typical QA task (e.g., SQuAD [Rajpurkar et al., 2016]), and encode the web document as the context. We then replace the output component of the MRC model (which is usually an answer classification module) with an attentional sequence generator that generates a free-form response. We refer to our approach as CMR (Conversation with on-demand Machine Reading). In general, any off-the-shelf MRC model could be applied here for knowledge comprehension. We use Stochastic Answer Networks (SAN)[2] [Liu et al., 2018b], a performant machine reading model that until very recently held state-of-the-art performance on the SQuAD benchmark. We also employ a simple but effective data weighting scheme to further encourage response grounding.

[2] https://github.com/kevinduh/san_mrc

5.3.1 Document and Conversation Reading

We adapt the SAN model to encode both the input document and conversation history and forward the digested information to a response generator. Figure 6.1 depicts the overall MRC architecture. Different blocks capture different concepts of representations in both the input conversation history and web document. The leftmost blocks represent the lexicon encoding that extracts information from X and D at the token level. Each token is first transformed into its corresponding word embedding vector, and then fed into a position-wise feed-forward network (FFN) [Vaswani et al., 2017a] to obtain the final token-level representation. Separate FFNs are used for the conversation history and the web document.

The next block is for contextual encoding. The aforementioned token vectors are concatenated with pre-trained 600-dimensional CoVe vectors [McCann et al., 2017], and then fed to a BiLSTM that is shared for both conversation history and web document. The step-wise outputs of the BiLSTM carry the information of the tokens as well as their left and right context.

The last block builds the memory that summarizes the salient information from both X and D. The block first applies *cross*-attention to integrate information from the conversation history X into the document representation. Each contextual vector of the document D is used to compute attention (similarity) distribution over the contextual vectors of X, which is concatenated with the weighted average vector of X by the resulting distribution. Second, a *self*-attention layer is applied to further ingest and capture the most salient information. The output memory, $M \in \mathbb{R}^{d \times n}$, is obtained by applying another BiLSTM layer for final information rearrangement. Note that d is the hidden size of the memory and n is the length of the document.

5.3.2 Response Generation

Having read and processed both the conversation history and the extra knowledge in the document, the model then produces a free-form response $\mathbf{y} = (y_1, \ldots, y_T)$ instead of generating a span or performing answer classification as in MRC tasks.

We use an attentional recurrent neural network decoder [Luong et al., 2015] to generate response tokens while attending to the memory. At the beginning, the initial hidden state $\boldsymbol{h}_0$ is the weighted sum of the representation of the history X. For each decoding step t with a hidden state $\boldsymbol{h}_t$, we generate a token y_t

based on the distribution:

$$p(y_t) = \text{softmax}((W_1 \boldsymbol{h}_t + \boldsymbol{b})/\tau), \tag{5.1}$$

where $\tau > 0$ is the softmax temperature. The hidden state $\boldsymbol{h}_t$ is defined as follows:

$$\boldsymbol{h}_t = W_2[\boldsymbol{z}_t +\!\!+ f_{attention}(\boldsymbol{z}_t, M)]. \tag{5.2}$$

Here, $[\cdot +\!\!+ \cdot]$ indicates a concatenation of two vectors; $f_{attention}$ is a dot-product attention [Vaswani et al., 2017a]; and $\boldsymbol{z}_t$ is a state generated by GRU$(\boldsymbol{e}_{t-1}, \boldsymbol{h}_{t-1})$ with $\boldsymbol{e}_{t-1}$ being the embedding of the word y_{t-1} generated at the previous $(t-1)$ step. In practice, we use top-k sample decoding to draw y_t from the above distribution $p(y_t)$. Section 6.5 provides more details about the experimental configuration.

5.3.3 Data Weighting Scheme

We further propose a simple data weighting scheme to encourage the generation of grounded responses. The idea is to bias the model training to fit better to those training instances where the ground-truth response is more closely relevant to the document. More specifically, given a training instance $(X, D, \boldsymbol{y})$, we measure the closeness score $c \in \mathbb{R}$ between the document D and the gold response $\boldsymbol{y}$ (e.g., with the NIST [Doddington, 2002] or BLEU [Papineni et al., 2002] metrics). In each training data batch, we normalize the closeness scores of all the instances to have a sum of 1, and weight each of the instances with its corresponding normalized score when evaluating the training loss. This training regime promotes instances with grounded responses and thus encourages the model to better encode and utilize the information in the document.

5.4 Dataset

To create a grounded conversational dataset, we extract conversation threads from Reddit, a popular and large-scale online platform for news and discussion. In 2015 alone, Reddit hosted more than 73M conversations.[3] On Reddit, user submissions are categorized by topics or "subreddits", and a submission typically consists of a submission title associated with a URL pointing to a news or background article, which initiates a discussion about the contents of the article. This article provides framing for the conversation, and this can

[3] https://redditblog.com/2015/12/31/reddit-in-2015/

	Train	Valid	Test
# dialogues	28.4k	1.2k	3.1k
# utterances	2.36M	0.12M	0.34M
# documents	28.4k	1.2k	3.1k
# document sentences	15.18M	0.58M	1.68M
Average length (# words):			
utterances	18.74	18.84	18.48
document sentences	13.72	14.17	14.15

Table 5.1: Our grounded conversational dataset.

naturally be seen as a form of grounding. Another factor that makes Reddit conversations particularly well-suited for our conversation-as-MRC setting is that a significant proportion of these URLs contain named anchors (i.e., '#' in the URL) that point to the relevant passages in the document. This is conceptually quite similar to MRC data [Rajpurkar et al., 2016] where typically only short passages within a larger document are relevant in answering the question.

We reduce spamming and offensive language by manually curating a list of 178 relatively "safe" subreddits and 226 web domains from which the web pages are extracted. To convert the web page of each conversation into a text document, we extracted the text of the page using an html-to-text converter,[4] while retaining important tags such as <title>, <h1> to <h6>, and <p>. This means the entire text of the original web page is preserved, but these main tags retain some high-level structure of the article. For web URLs with named anchors, we preserve that information by indicating the anchor text in the document with tags <anchor> and </anchor>. As the whole documents in the dataset tend to be lengthy, anchors offer important hints to the model about which parts of the documents should likely be focused on in order to produce a good response. We considered it sensible to keep them as they are also available to the human reader.

After filtering short or redacted turns, or which quote earlier turns, we obtained 2.8M conversation instances respectively divided into train, validation, and test (Table 6.3). We used different date ranges for these different sets: years 2011-2016 for train, Jan-Mar 2017 for validation, and the rest of 2017 for test. For the test set, we select conversational turns for which 6 or more responses were available, in order to create a multi-reference test set. Given other filtering criteria such as turn length, this yields a 6-reference test set of size 2208. For each instance, we set aside one of the 6 human responses to assess human performance on

[4] `https://www.crummy.com/software/BeautifulSoup`

this task, and the remaining 5 responses serve as ground truths for evaluating different systems.[5] Table 6.3 provides statistics for our dataset, and Figure 5.1 presents an example from our dataset that also demonstrates the need to combine conversation history and background information from the document to produce an informative response.

To enable reproducibility of our experiments, we crawled web pages using Common Crawl (`http://commoncrawl.org`), a service that crawls web pages and makes its historical crawls available to the public. We also release the code (URL redacted for anonymity) to recreate our dataset from both a popular Reddit dump[6] and Common Crawl, and the latter service ensures that anyone reproducing our data extraction experiments would retrieve exactly the same web pages. We made a preliminary version of this dataset available for a shared task [Galley et al., 2019] at Dialog System Technology Challenges (DSTC) [Yoshino et al., 2019]. Back-and-forth with participants helped us iteratively refine the dataset. The code to recreate this dataset is included.[7]

5.5 Experiments

5.5.1 Systems

We evaluate our systems and several competitive baselines:

SEQ2SEQ [Sutskever et al., 2014] We use a standard LSTM SEQ2SEQ model that only exploit the conversation history for response generation, without any grounding. This is a competitive baseline initialized using pretrained embeddings.

MEMNET: We use a Memory Network designed for grounded response generation [Ghazvininejad et al., 2018]. An end-to-end memory network [Sukhbaatar et al., 2015] encodes conversation history and sentences in the web documents. Responses are generated with a sequence decoder.

CMR-F : To directly measure the effect of incorporating web documents, we compare to a baseline which

[5] While this is already large for a grounded dataset, we could have easily created a much bigger one given how abundant Reddit data is. We focused instead on filtering out spamming and offensive language, in order to strike a good balance between data quality and size.

[6] `http://files.pushshift.io/reddit/`

[7] We do not report on shared task systems here, as these systems do not represent our work and some of these systems have no corresponding publications. Along with the data described here, we provided a standard SEQ2SEQ baseline to the shared task, which we improved for the purpose of this paper (improved BLEU, NIST and METEOR). Our new SEQ2SEQ baseline is described in Section 6.5.

	Appropriateness			Grounding			Diversity			
	NIST	BLEU	METEOR	Precision	Recall	F1	Entropy-4	Distinct-1	Distinct-2	Len
Human	2.650	3.13%	8.31%	2.89%	0.45%	0.78%	10.445	0.167	0.670	18.757
SEQ2SEQ	2.223	1.09%	7.34%	1.20%	0.05%	0.10%	9.745	0.023	0.174	15.942
MEMNET	2.185	1.10%	7.31%	1.25%	0.06%	0.12%	9.821	0.035	0.226	15.524
CMR-F	**2.260**	1.20%	7.37%	1.68%	0.08%	0.15%	9.778	0.035	0.219	15.471
CMR	2.213	**1.43**%	7.33%	2.44%	0.13%	0.25%	9.818	0.046	0.258	15.048
CMR+W	2.238	1.38%	**7.46**%	**3.39**%	**0.20**%	**0.38**%	**9.887**	**0.052**	**0.283**	15.249

Table 5.2: Automatic Evaluation results (higher is better for all metrics). Our best models (CMR+W and CMR) considerably increase the quantitative measures of Grounding, and also slightly improve Diversity. Automatic measures of Quality (e.g., BLEU-4) give mixed results, but this is reflective of the fact that we did not aim to improve response relevance with respect to the context, but instead its level of grounding. The human evaluation results in Table 5.3 indeed suggest that our best system (CMR+W) is better.

omits the document reading component of the full model (Figure 6.1). As with the SEQ2SEQ approach, the resulting model generates responses solely based on conversation history.

CMR: To measure the effect of our data weighting scheme, we compare to a system that has identical architecture to the full model, but is trained without associating weights to training instances.

CMR+W: As described in section 5.3, the full model reads and comprehends both the conversation history and document using an MRC component, and sequentially generates the response. The model is trained with the data weighting scheme to encourage grounded responses.

Human: To get a better sense of the systems' performance relative to an upper bound, we also evaluate human-written responses using different metrics. As described in Section 5.4, for each test instance, we set aside one of the 6 human references for evaluation, so the 'human' is evaluated against the other 5 references for automatic evaluation. To make these results comparable, all the systems are also automatically evaluated against the same 5 references.

5.6 Experiment Details

For all the systems, we set word embedding dimension to 300 and used the pretrained GloVe[8] for initialization. We set hidden dimensions to 512 and dropout rate to 0.4. GRU cells are used for SEQ2SEQ and

[8] https://nlp.stanford.edu/projects/glove/

MEMNET (we also tested LSTM cells and obtained similar results). We used the Adam optimizer for model training, with an initial learning rate of 0.0005. Batch size was set to 32. During training, all responses were truncated to have a maximum length of 30, and maximum query length and document length were set to 30, 500, respectively. we used regular teacher-forcing decoding during training. For inference, we found that top-k random sample decoding [Fan et al., 2018] provides the best results for all the systems. That is, at each decoding step, a token was drawn from the k most likely candidates according to the distribution over the vocabulary. Similar to recent work [Fan et al., 2018; Edunov et al., 2018], we set $k = 20$ (other common k values like 10 gave similar results). We selected key hyperparameter configurations on the validation set.

5.6.1 Evaluation Setup

Table 5.2 shows automatic metrics for quantitative evaluation over three qualities of generated texts. We measure the overall **relevance** of the generated responses given the conversational history by using standard Machine Translation (MT) metrics, comparing generated outputs to ground-truth responses. These metrics include BLEU-4 [Papineni et al., 2002], METEOR [Lavie and Agarwal, 2007]. and NIST [Doddington, 2002]. The latter metric is a variant of BLEU that weights n-gram matches by their information gain by effectively penalizing uninformative n-grams (such as "I don't know"), which makes it a relevant metric for evaluating systems aiming diverse and informative responses. MT metrics may not be particularly adequate for our task [Liu et al., 2016b], given its focus on the informativeness of responses, and for that reason we also use two other types of metrics to measure the level of grounding and diversity.

As a **diversity** metric, we count all n-grams in the system output for the test set, and measure: (1) Entropy-n as the entropy of the n-gram count distribution, a metric proposed in [Zhang et al., 2018b]; (2) Distinct-n as the ratio between the number of n-gram types and the total number of n-grams, a metric introduced in [Li et al., 2016a].

For the **grounding** metrics, we first compute '#match,' the number of non-stopword tokens in the response that are present in the document but not present in the context of the conversation. Excluding words from the conversation history means that, in order to produce a word of the document, the response generation system is very likely to be effectively influenced by that document. We then compute both *precision* as '#match' divided by the total number of non-stop tokens in the response, and *recall* as '#match' divided by

Human judges preferred:				
Our best system		Neutral	Comparator	
CMR+W	*__44.17__%	26.27%	29.56%	SEQ2SEQ
CMR+W	*__40.93__%	25.80%	33.27%	MEMNET
CMR+W	__37.67__%	27.53%	34.80%	CMR
CMR+W	30.37%	16.27%	*__53.37__%	Human

Table 5.3: Human Evaluation results, showing preferences (%) for our model (CMR+W) vs. baseline and other comparison systems. Distributions are skewed towards CMR+W. The 5-point Likert scale has been collapsed to a 3-point scale. *Differences in mean preferences are statistically significant ($p \leq 0.0001$).

the total number of non-stop tokens in the document. We also compute the respective *F1* score to combine both. Looking only at exact unigram matches between the document and response is a major simplifying assumption, but the combination of the three metrics offers a plausible proxy for how greatly the response is grounded in the document. It seems further reasonable to assume that these can serve as a surrogate for less quantifiable forms of grounding such as paraphrase – e.g., *US* $\rightarrow$ *American* – when the statistics are aggregated on a large test dataset.

5.6.2 Automatic Evaluation

Table 5.2 shows automatic evaluation results for the different systems. In terms of appropriateness, the different variants of our models outperform the SEQ2SEQ and MEMNET baselines, but differences are relatively small and, in case of one of the metrics (NIST), the best system does not use grounding. Our goal, we would note, is not to specifically improve response appropriateness, as many responses that completely ignore the document (e.g., *I don't know*) might be perfectly appropriate. Our systems fare much better in terms of Grounding and Diversity: our best system (CMR+W) achieves an F1 score that is more than three times (0.38% vs. 0.12%) higher than the most competitive non-MRC system (MEMNET).

5.6.3 Human Evaluation

We sampled 1000 conversations from the test set. Filters were applied to remove conversations containing ethnic slurs or other offensive content that might confound judgments. Outputs from systems to be compared were presented pairwise to judges from a crowdsourcing service. Four judges were asked to compare each pair of outputs on Relevance (the extent to which the content was related to and appropriate to the

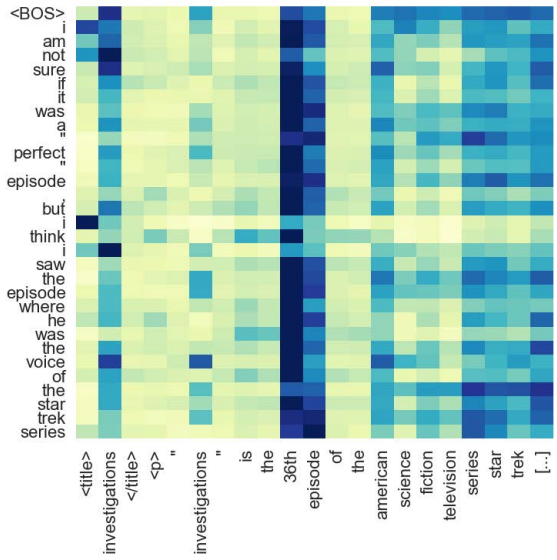


Figure 5.3: Attention weights between words of the documents and words of the response. Dark (blue) cells represent probabilities closer to 1.

conversation) and Informativeness (the extent to which the output was interesting and informative). Judges were asked to agree or disagree with a statement that one of the pair was better than the other on the above two parameters, using a 5-point Likert scale.[9] Pairs of system outputs were randomly presented to the judges in random order in the context of short snippets of the background text. These results are presented in summary form in Table 5.3, which shows the overall preferences for the two systems expressed as a percentage of all judgments made. Overall inter-rater agreement measured by Fliess' Kappa was 0.32 ("fair"). Nevertheless, the differences between the paired model outputs are statistically significant (computed using 10,000 bootstrap replications).

5.6.4 Qualitative Study

Table 5.4 illustrates how our best model (CMR+W) tends to produce more contentful and informative responses compared to the other systems. In the first example, our system refers to a particular *episode* mentioned in the article, and also uses terminology that is more consistent with the article (e.g., *series*). In the second example, *humorous song* seems to positively influence the response, which is helpful as the input doesn't mention singing at all. In the third example, the CMR+W model clearly grounds its response to the article as it states the fact (Steve Jobs: CEO of Apple) retrieved from the article. The outputs by the other

[9]The choices presented to the judges were *Strongly Agree*, *Agree*, *Neutral*, *Disagree*, and *Strongly Disagree*.

Doc	<title> Investigations </title> "Investigations" is the **36th episode** of the American science fiction **television series** Star Trek: Voyager which aired on the UPN network. It is the **36th episode** of the second season. Set in the 24th century, **the series** follows the adventures of the Federation starship Voyager during its journey home to Earth, having been stranded tens of thousands of light-years away. [...]
Input	[...] King Abdullah II of Jordon once cameo'd as an unnamed ensign on Star Trek Voyager
MEMNET	I think it should be noted that he is like a good guy who just killed her and her son.
SEQ2SEQ	well , i didn't know , but i was just saying that it wasn't a joke.
CMR+W	I am not sure if it was a **"perfect" episode**, but I think I saw the **episode** where he was the voice of the **Star Trek series**.
Doc	<title> "Weird Al" Yankovic </title> Alfred Matthew "Weird Al" Yankovic is an American singer, songwriter, parodist, record producer, satirist, actor, voice actor, music video director, film producer, and author. He is **known** for **his humorous songs** that make light of popular culture and often parody specific songs by contemporary musical acts, original songs that are style pastiches of the work of other acts, and polka medleys of several popular songs, featuring his favored instrument, the accordion. *(***Prince** *and* **West** *mentioned several times in the rest of the article.)* [...]
Input	That's fair, I believe Weird Al always got permission from artists before he parodied them which is nice.
MEMNET	Well then you don't need to make a sequel to the front page of the movie.
SEQ2SEQ	If I remember correctly. I had no idea who had any good chance in history class.
CMR+W	No, because **his song was a parody** of the artist **formerly known as "Kanye West"**.
Doc	<title> Steve Jobs </title> Steven Paul Jobs was an American entrepreneur, businessman, inventor, and industrial designer. He was the chairman, **chief executive officer (CEO)**, and co-founder of **Apple Inc**.; chairman and majority shareholder of Pixar; a member of The Walt Disney Company's board of directors following its acquisition of Pixar; and the founder, chairman, and CEO of next. [...]
Input	Steve Jobs was a mediocre programmer and one of the greatest designers and marketers in human history. But I prefer Bill Gates if we want to talk about hero worship myself. [...]
MEMNET	What if you don't want to see this movie?
SEQ2SEQ	No ... the whole point is that this is a good way to make a difference.
CMR+W	So he's the **CEO** of **Apple**.

Table 5.4: Sample output comparing our best system (CMR+W) against Memory Networks and a SEQ2SEQ baseline. The source documents were manually shortened to fit in the table, without significantly affecting meaning.

two baseline models are instead not relevant in the context.

Figure 5.3 displays the attention map of the generated response and (part of) the document from our full model. The model successfully attends to the key words (e.g., *36th, episode*) of the document. Note that the attention map is unlike what is typical in machine translation, where target words tend to attend to different portions of the input text. In our task, where alignments are much less one-to-one compared to machine translation, it is common for the generator to retain focus on the key information in the external document to produce semantically relevant responses.

5.7 Related Work

Dialogue: Traditional dialogue systems (see [Jurafsky and Martin, 2009] for an historical perspective) are typically grounded, enabling these systems to be reflective of the user's environment. The lack of grounding has been a stumbling block for the earliest end-to-end dialogue systems, as various researchers have noted that their outputs tend to be bland [Li et al., 2016a; Gao et al., 2019b], inconsistent [Zhang et al., 2018a; Li et al., 2016b; Zhang et al., 2019c], and lacking in factual content [Ghazvininejad et al., 2018; Agarwal et al., 2018]. Recently there has been growing interest in exploring different forms of grounding, including images, knowledge bases, and plain texts [Das et al., 2017; Mostafazadeh et al., 2017; Agarwal et al., 2018; Yang et al., 2019]. A recent survey is included in Gao et al. [2019a].

Prior work, e.g, [Ghazvininejad et al., 2018; Zhang et al., 2018a; Huang et al., 2019b], uses grounding in the form of independent snippets of text: Foursquare tips and background information about a given speaker. Our notion of grounding is different, as our inputs are much richer, encompassing the full text of a web page and its underlying structure. Our setting also differs significantly from relatively recent work [Dinan et al., 2019; Moghe et al., 2018] exploiting crowdsourced conversations with detailed grounding labels: we use Reddit because of its very large scale and better characterization of real-world conversations. We also require the system to learn grounding directly from conversation and document pairs, instead of relying on additional grounding labels. Moghe et al. [2018] explored directly using a span-prediction QA model for conversation. Our framework differs in that we combine MRC models with a sequence generator to produce free-form responses.

Machine Reading Comprehension: MRC models such as SQuAD-like models, aim to extract answer spans (starting and ending indices) from a given document for a given question [Seo et al., 2017; Liu et al., 2018b; Yu et al., 2018]. These models differ in how they fuse information between questions and documents. We chose SAN [Liu et al., 2018b] because of its representative architecture and competitive performance on existing MRC tasks. We note that other off-the-shelf MRC models, such as BERT [Devlin et al., 2018b], can also be plugged in. We leave the study of different MRC architectures for future work. Questions are treated as entirely independent in these "single-turn" MRC models, so recent work (e.g., CoQA [Reddy et al., 2019] and QuAC [Choi et al., 2018]) focuses on multi-turn MRC, modeling sequences of questions and answers

in a conversation. While multi-turn MRC aims to answer complex questions, that body of work is restricted to factual questions, whereas our work—like much of the prior work in end-to-end dialogue—models free-form dialogue, which also encompasses chitchat and non-factual responses.

5.8 Conclusions

We have demonstrated that the machine reading comprehension approach offers a promising step to generating, *on the fly*, contentful conversation exchanges that are grounded in extended text corpora. The functional combination of MRC and neural attention mechanisms offers visible gains over several strong baselines. We have also formally introduced a large dataset that opens up interesting challenges for future research.

The CMR (Conversation with on-demand machine reading) model presented here will help connect the many dots across multiple data sources. One obvious future line of investigation will be to explore the effect of other off-the-shelf machine reading models such as BERT [Devlin et al., 2018b] within the CMR framework.

Chapter 6

TimeDial: Temporal Commonsense Reasoning in Dialog

The chapter discusses work originally published in [Qin et al., 2021].

Everyday conversations require understanding everyday events, which in turn, requires understanding temporal commonsense concepts interwoven with those events. Despite recent progress with massive pre-trained language models (LMs) such as T5 and GPT-3, their capability of temporal reasoning in dialogs remains largely under-explored. In this work, we present the first study to investigate pre-trained LMs for their temporal reasoning capabilities in dialogs by introducing a new task and a crowd-sourced English challenge set, TIMEDIAL. We formulate TIMEDIAL as a multiple choice cloze task with over 1.1K carefully curated dialogs. Empirical results demonstrate that even the best performing models struggle on this task compared to humans, with 23 absolute points of gap in accuracy. Furthermore, our analysis reveals that the models fail to reason about dialog context correctly; instead, they rely on shallow cues based on existing temporal patterns in context, motivating future research for modeling temporal concepts in text and robust contextual reasoning about them. The dataset is publicly available at: `https://github.com/google-research-datasets/timedial`.

A: May we see the wine list please. B: Sure. Our special wine today is a 1989 Chardonnay. A: **I'd like a bottle** please. B: I'll need to **see your ID** please. A: Here you go. B: Sorry about the inconvenience, you look so young. I had to make sure you are over _____. a) 21 years old ✓ b) 30 years old ✗ c) 4 years old ✗ d) 18 years old ✓
A: Good morning! May I help you? B: Yes. My wife and I are interested in renting a house for the summer. A: Very well. How long do you want the house? All summer? B: No, not all summer. Just for **six weeks** . A: I am afraid I can only rent it for **two months** . B: My holiday is only _____, but I think my brother and his family would take it for the other **two weeks** . a) six decades ✗ b) 45 days ✓ c) six weeks ✓ d) two months ✗

Table 6.1: Examples from our TIMEDIAL challenge set, demonstrating the need for commonsense knowledge and arithmetic reasoning over the context to infer the correct answers. Key contextual information for reasoning success is highlighted.

6.1 Introduction

Humans can effortlessly reason about temporal concepts of everyday events such as their duration, frequency, or relative ordering [Allen, 1984; Radvansky and Zacks, 2014] based on rich commonsense knowledge about how the world works, especially in relation to time. However, reasoning about such concepts has been challenging for machines [Kahn and Gorry, 1977; Kozareva and Hovy, 2011] since it requires both understanding the local temporal expressions and reasoning about their global contexts such as their relative ordering and relations [UzZaman et al., 2013; Ning et al., 2018b; Pustejovsky, 2017]. The problem becomes even more challenging in dialogs, where explicit and implicit inter-dependencies among temporal concepts can appear across conversation turns.

For instance, for the first dialog in Table 6.1, one must understand the context, i.e., selling wine, and use world knowledge of minimum legal drinking age in order to reason about correct answers to fill in the blank. Similarly, in the second conversation, commonsense about the durations *summer*, *month*, *week*, *day* and their relations, plus numerical reasoning, are necessary to make the inference.

Although previous works have studied temporal reasoning in natural language, they have either focused on specific time-related concepts in isolation, such as temporal ordering and relation extraction [Leeuwenberg and Moens, 2018; Ning et al., 2018a], and/or dealt with limited context, such as single-sentence-based question answering [Zhou et al., 2019] and natural language inference [Vashishtha et al., 2020; Mostafazadeh et al., 2016a].

In this work, we make the first systematic study of temporal commonsense reasoning in a multi-turn dialog setting. The task involves complex reasoning that requires operations like comparison and arithmetic reasoning over temporal expressions and the need for commonsense and world knowledge.

We design a new task for dialog-based temporal reasoning and present a new challenge set in English, called TIMEDIAL, to evaluate language understanding models on the task. We formulate the problem as a crowd-sourced cloze task with multiple choices based on dialogs in the DailyDialog dataset [Li et al., 2017]. Given a dialog with one temporal span masked out, the model is asked to find *all* correct answers from a list of four options to fill in the blank (Table 6.1).

The challenge set requires the models to demonstrate understanding of the context and use temporal commonsense to make right choices. Our final challenge set consists of 1.1K carefully curated dialog instances.

We then study the performance of several state-of-the-art pre-trained language models on TIMEDIAL along several dimensions including modeling paradigms (classification, mask filling, and generation), the scope of dialog contexts, in-domain vs. out-of-domain training, dependence on shallow text matching for reasoning, and the types of reasoning required. Our experiments demonstrate that off-the-shelf, pre-trained language models cannot effectively reason about temporal aspects in a dialog, even with domain-specific finetuning. Our findings indicate that large-scale pre-trained models even after fine-tuning may not be sufficient for robust temporal reasoning in dialogs, and motivate future research toward modeling temporal concepts over diverse everyday events, and contextual reasoning about them.

6.2 Task: Temporal Reasoning in Dialog

We formulate the dialog-based temporal commonsense reasoning problem as a *cloze* task [Taylor, 1953]. Formally, given a multi-turn dialog context of n conversational turns between two speakers A and B, where

a temporal words span within the context is masked out, the task is to predict the suitable temporal expression(s) for the masked-out span from a list of options. That is, we want the conversation model to select all the correct answers from the options based on the dialog context. Following similar cloze-style challenge datasets, we use accuracy as the evaluation metric [Mostafazadeh et al., 2016a; Onishi et al., 2016; Mihaylov and Frank, 2018].

Having a non-trivial set of options is crucial to build a challenge set and to avoid accidental spurious biases [Geirhos et al., 2020; Gururangan et al., 2018; Le Bras et al., 2020]. We ensure this via the following filtering process. **(1)** For each masked span, there is more than one correct answer in the options. This makes the task more challenging for models since more comprehensive understanding of the context is required to recognize all the correct choices. In our dataset (§6.3) we guarantee two incorrect answers for each masked span. **(2)** Some incorrect options are selected to be spuriously correlated with the dialog context. For example, we include temporal spans in the dialog context as negative options, which will challenge models that rely primarily only on shallow pattern matching without correct temporal reasoning. We present more information in §6.3 about how the negative options were created by human annotators.

6.3 Dataset: TIMEDIAL

The TIMEDIAL dataset is derived from DailyDialog data [Li et al., 2017], which is a multi-turn dialog corpus containing over 13K English dialogs. Dialogs in this dataset consist of turn-taking between two people on topics over 10 broad categories, ranging from daily lives to financial topics.

6.3.1 Data Collection

Our data collection process involves two steps: (1) identifying dialogs that are rich in temporal expressions, and (2) asking human annotators to provide correct and incorrect options for cloze instances derived from these dialogs. We now describe these steps in detail.

Temporal expression identification. Here, we select dialogs that are rich with temporal information, in order to focus on complex temporal reasoning that arises in natural dialogs. Temporal expressions are automatically identified with SUTime [Chang and Manning, 2012], an off-the-shelf temporal expression

Category	Dialog	Options
World Knowledge (5%)	A: May we see the wine list ? B: Sure . Our special wine today is a 1989 Chardonnay . A: That sounds pretty good! How much is it ? B: It's $4.25 cents by the glass . The whole bottle is $22.25 . A: **I'd like a bottle** please . B: I'll need to **see your ID** please . A: Here you go . B: Sorry about the inconvenience, I had make sure you are over _____.	✓ 21 years old ✗ 30 years old ✗ 4 years old ✓ 18 years old
Comparison (24%)	A: Yes , sir. May I help you? B: Please I'd like a ticket to New York. A: For today? B: No, **early Saturday morning** . A: We have a flight that we'll put you there at _____. Is that ok? B: **Nothing earlier? I prefer flight at 9 thirty.** A: I'm afraid not , unless you want a night flight. B: No, exactly not.	✓ ten AM ✗ 9:30 PM ✓ eleven AM ✗ four AM
Arithmetic (5%)	A: How long do you want the house ? All summer ? B: No , just for six weeks. A: I'm afraid I can only **rent it for two months** . B: My holiday is only _____, but I think my brother and his family would take it for the **other two weeks** .	✗ six decades ✓ 45 days ✓ six weeks ✗ two months
General Commonsense (60%)	A: Do you get up early every morning ? B: About 6 in the morning. **I like to walk to the office** . A: Good habit. How long does it take ? B: _____. Do you live alone ? A: No , my little sister lives with me . . .	✓ 20 minutes ✗ 10 seconds ✓ 15 minutes ✗ 20 hours
Others (6%)	A: How long does a facial service take? B: We have half-hour and one-hour treatments. A: What's the regular price? B: Well , **the half-hour** facial costs $50 and **the one-hour** costs $80. A: Good , I will take _____ facial. B: That's fine , madam.	✓ the one hour ✗ the 20 hour ✗ the 80 second ✓ the half hour

Table 6.2: Example dialogs and answer options from the TIMEDIAL dataset, categorized by the nature of reasoning required to correctly answer them, along with the percentage of each reasoning category in the set of 100 sampled examples. The relevant key information in the dialog context is highlighted.

detector.[1] We keep only the dialogs with more than 3 temporal expressions and at least one expression that contains **numerals** like "*two weeks*" (as opposed to non-numeric spans, like "*summer*", "*right now*", and "*later*"). In our initial experiment, we observe that language models can often correctly predict these non-numerical temporal phrases.

We note that temporal expressions containing numerals serve as more challenging sets of options than non-numerical ones. This filtering step results in 1,127 unique dialogs for further processing.

Human annotated options. Next, we make spans in the dialogs. For a dialog, we mask out each temporal expression that contains numerals, each resulting in a cloze question that is then sent for human annotation.

This resulted in 1,526 instances for annotation. For each masked span in each dialog, we obtain human

[1] https://nlp.stanford.edu/software/sutime.shtml

annotation to derive a fixed set of correct and incorrect options given the context. Concretely, given a masked dialog and a seed correct answer (i.e., the original text) for the masked span, the annotators[2] were asked to (1) come up with *an alternative correct answer* that makes sense in the dialog adhering to commonsense, and (2) formulate *two incorrect answers* that have no possibility of making sense in the dialog context. We highlight all time expressions in the context to make it easier for annotators to select reasonable time expressions.

To ensure that the annotated incorrect options are not too trivially distinguishable by the models (as discussed in §6.2), we define three rules for the annotators to follow.

- **Rule 1: Phrase Matching.** The rater should first try to pick another temporal span from the dialog context that makes syntactic/semantic sense (e.g., when the span is of the appropriate type, such as duration, for the masked span) but is still incorrect according to commonsense.
- **Rule 2: Numeral Matching.** If Rule 1 does not apply, raters should follow a relaxed version of Rule 1, whereby the incorrect option should contain any numeral occurring in the dialog context.
- **Rule 3: Open-ended.** If neither of the above rules is applicable, then raters can come up with an incorrect option using their own judgment. The two incorrect options are required to differ from each other as much as possible.

Rules-1&2 are designed to confuse models that rely on shallow pattern matching. Finally, to ensure the quality of the human-annotated options, we perform a subsequent round of human validation on the gathered data. The validators identify and fix issues such as duplicate options, unreasonable or obscure annotations w.r.t natural usage, or ungrammatical annotations that do not fit in the context.

6.3.2 Properties of TIMEDIAL

Table 6.3 shows statistics of TIMEDIAL. The dataset contains over 1.1K test instances. Each dialog contains 11.7 turns and 3 temporal expressions on average, presenting richer and more complex context compared to the recent single-sentence-based temporal question answering benchmarks [e.g., Zhou et al., 2019; Vashishtha et al., 2020]. As above, each test instance contains two correct answers and two incorrect ones.[3]

[2] who are English linguists.

[3] We also collected 342 extra instances for which the annotators deem there is only one unique correct answer for the context. Thus, each of those instances contains one correct option and two incorrect ones. We release those instances along with the dataset, though we did not include them in empirical study in this work.

# Dialog instances	1,104
# Temporal Expressions	1,985
# Avg. Turns Per Dialog	11.7
# Avg. Words Per Turn	16.5
# Avg. Time Spans Per Dialog	3.0
Incorrect Options	
% Phrase Matching	16.3 %
% Numeral Matching	49.6 %
% Open-ended	45.4 %

Table 6.3: Statistics of our TIMEDIAL challenge set.

Over half of the incorrect options are annotated based on phrase and numeral matching from context, which pose a significant challenge for models relying on shallow text matching, as we show in our experimental analysis (§6.5).

Answering different instances in the dataset requires different types of core reasoning abilities, such as comparison, arithmetic inference, or reasoning based on world knowledge or general commonsense. To facilitate fine-grained analysis, we also annotate the **reasoning categories** for a randomly sampled set of 100 dialogs. Though each instance can involve multiple reasoning types, we associate it with one predefined category label that indicates the primary type of reasoning it requires. Table 6.2 shows the category distribution and examples in each of the category. We observe that the dataset requires general commonsense for 60% of the dialogs, making it the most common reasoning type.

6.4 Modeling

We consider a broad set of methods and evaluate their performance on our challenge TIMEDIAL dataset. These methods vary in terms of the modeling paradigms, the scope of the dialog contexts, and training settings. In particular, they encompass the major ways pre-trained LMs are currently used in downstream tasks (§6.4.1) which often outperform earlier specialized non-pretrained models. We also consider different lengths of context used in reasoning, varying by their vicinity to the masked span (§6.4.2). Finally, we study different training settings, including zero-shot, in-domain, and out-of-domain training (§6.4.3).

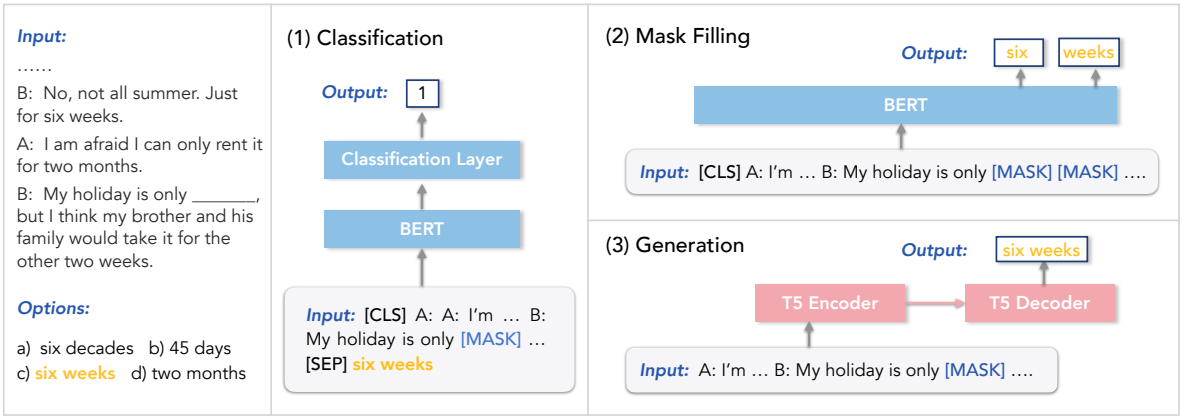


Figure 6.1: We study three modeling paradigms for the task, based on BERT and T5, including (1) Classification, (2) Mask Filling, and (3) Generation (§6.4.1). The models are finetuned with various training data, as discussed in §6.4.3.

6.4.1 Modeling Paradigms

We experiment across three major modeling paradigms: (i) Binary Classification, (ii) Mask Filling, and (iii) Generation. Figure 6.1 shows the different architectures. For each test instance, the model takes as input a pair of (masked dialog context, candidate), and outputs a score measuring how likely the candidate being a correct answer. Based on the prediction scores of all options, the model then chooses the top two positive candidates as the predicted answer for the instance. Each paradigm of models is finetuned using training data from different domains, as discussed in §6.4.3.

Binary Classification

In this setting, we formulate the task as a binary classification problem, i.e., we use a classifier to measure the probability of the candidate in the (masked dialog context, candidate) pair being a correct answer. Any powerful LM — e.g., BERT [Devlin et al., 2019], ALBERT [Lan et al., 2019], RoBERTa [Liu et al., 2019], etc. can be used to build the classifier.

This method's key challenge is the lack of annotated training data for direct supervision. We generate weak supervision training data as follows. In an unlabeled corpus, we use the SUTime tool to annotate temporal spans. We mask each temporal span in this corpus and use the masked text as one positive example for binary classification. To generate negative example, we randomly sample another temporal span from the dialog context and use it as a negative example for the masked temporal span. The resulting data is noisy

because the randomly sampled temporal span can also logically fit in the masked span in the given context; however, we assume the likelihood of that happening is low. We leave drawing harder negative instances using heuristics to future work.

Mask Filling

We also use the mask filling approach of BERT-like mask language models (MLMs). For each dialog context and a candidate temporal span of m tokens, we replace the blank in the dialog context with m masked tokens. We then evaluate the likelihood of predicting the temporal span tokens for those masked positions, and make average across the positions. A key advantage of this method is that we can directly apply a BERT model in the *zero-shot* manner since the model was pre-trained in the same way, as for accommodating for `[MASK]` fillings. Additionally, we also finetune BERT's MLM for learning task specific properties.

Generation

The third method is a fully generative approach using the *text-to-text* paradigm of T5 [Raffel et al., 2020]. Given a masked dialog context, the model is trained to generate the masked text in an encoder-decoder framework. As a result, evaluating the likelihood of generating the given temporal span (normalized with the length of the span) is used as the probability of it being correct. Similar to mask filling, we use T5 either in a zero-shot manner or with additional fine-tuning.

6.4.2 Dialog Context

We aim to study the influence of context on a model's temporal reasoning in dialog by incorporating varying scopes of dialog context based on their vicinity to the target span. Since the dialogs in TIMEDIAL are rich in temporal concepts, we want to evaluate LMs' dependence on shallow text matching vs. the ability to accurately understand the causal relations between those concepts (see Table 6.6). We use the following three settings:

- **Full** context, where the model is presented with the complete available dialog to reason on. Due to our design of challenging negatives, the full context can often confuse models that rely on shallow cues.

Mask Filling and *Generation*			
	# Train	# Dev	
In-domain (Daily)	14.5K	2.4K	
Out-domain (Meena)	1.26M	23K	
Classification			
	# Train	# Dev	# Spans
In-domain (Daily)	58.0K	9.6K	2,153
Out-domain (Meena)	5.04M	92K	38,750

Table 6.4: Number of training and development instances for different settings. An instance is derived by masking one temporal span of a dialog. For classification, we draw 3 negative samples per positive sample. "# Spans" is the size of temporal span pool from which negative samples are drawn for weak supervision.

- **Local** context, where we provide only with the utterances that immediately precede and follow the target utterance.
- **Target** context, where the context is restricted to only the particular utterance that contains the masked span.

6.4.3 Training Details

For all models, we consider two common training settings, e.g., in-domain data, which is typically small, and out-of-domain training where a large amount of data is available. Table 6.4 shows training data statistics. For mask-filling and generation, we also evaluate in a zero-shot setup with no finetuning.

In-domain training. Our challenge TIMEDIAL test set is derived from contextually rich dialogs from the DailyDialog dataset, based on the number of temporal spans. However, this still leaves remaining data with less than 3 temporal spans or with no numeric span. By masking each temporal span in each dialog, we obtain 14.5K training instances to use in our domain specific fine-tuning.

Out-of-domain training. In this setting, we consider a much larger corpus from a general domain. Specifically, we use the large scale training set based on the Meena dataset Adiwardana et al. [2020], which is mined and filtered from public domain social media conversations over 341GB of text (40B words).[4] Compared to the above in-domain data from DailyDialog which were manually written by human annotators in

[4]We acquired a trimmed down version of the Meena dataset by contacting the authors.

SIZE-TRAIN	2-*best Acc (%)*
Classification (BERT)	
BASE-OUT	43.1
BASE-IN	51.1
LARGE-OUT	48.7
LARGE-IN	53.2
Mask Filling (BERT)	
BASE-ZERO	44.8
BASE-OUT	47.4
BASE-IN	67.4
LARGE-ZERO	47.7
LARGE-OUT	54.8
LARGE-IN	70.0
Generation (T5)	
BASE-ZERO	39.8
BASE-OUT	50.6
BASE-IN	59.2
LARGE-ZERO	39.1
LARGE-OUT	61.9
LARGE-IN	**74.8**
Human	**97.8**

Table 6.5: Model and human performance on TIMEDIAL. BASE and LARGE denote the size of the pre-trained BERT and T5; ZERO, IN, and OUT denote that the model is zero-shot (with no finetuning), fintuned using the in-domain DailyDialog data, or finetuned using the out-of-domain Meena data, respectively. THe full dialog context is used for all models.

a clean and consistent way, the dialogs in the Meena corpus tend to be noisy, casual, and usually short. Like our DailyDialog processing, we identify all temporal expressions for dialogs in Meena using SUTime.

6.5 Experiments and Analyses

Using the proposed TIMEDIAL challenge set, we next conduct extensive experiments and analyses on the different model variants and context settings. We use either 4x4 or 8x8 Cloud TPUs V3 pod slices[5] for fine-tuning and one V100 GPU for inference. We provide more details of the experiment configurations in the appendix.

[5] https://cloud.google.com/tpu

Dialog Context	Options	GOLD	CLS	MF	GEN
A: What's the date **today**? B: **Today** is **September 28th, 2007**. A: I have a meeting this **afternoon**. B: When will it begin? A: It will begin at **three o'clock**. What's the time **now**? B: It is _____. A: I have to go **now**. I don't want to be late. B: Don't worry, time is enough.	half past one	✓	✓	✗	✓
	quarter to two	✓	✗	✗	✗
	half past three	✗	✓	✓	✓
	half past nine	✗	✗	✓	✗
A: Doctor, I feel much better **now**. Will I be able to go home **some time this week**? B: That's good to hear. You've had an ideal recovery from your operation. We're going to send you home tomorrow. A: Do you think I can get back to work **very soon**? B:Don't be in such a hurry. I'm confident that you'll be completely recovered in _____. A: Is there anything I should do? B: You'd better have a good rest for **a week**.	4 to 6 weeks	✓	✗	✗	✓
	5 to 7 weeks	✓	✓	✗	✗
	a week	✗	✓	✓	✓
	a day	✗	✗	✓	✗

Table 6.6: Example prediction errors made by different models for cases with challenging options, based on the phrase and numeral matching rules (§6.3). GOLD denotes the true labels. The model predictions show that the models get confused by learning shallow text matching in terms of pre-existing temporal concepts (marked by bold faced text) in the context.

Evaluation. Since each example of TIMEDIAL contains two correct answers, we report the metric 2-*best accuracy*, which measures whether *both* of the model's top-ranked answers are correct. In other words, if the model erroneously ranks an incorrect answer over a correct one, we consider it to be an error case. Note that we use the ranking-based metric as opposed to classification-based ones (for example, by asking the model to classify whether each individual candidate answer is correct or not [e.g., Zhou et al., 2019]) and because it presents a stricter measure that penalizes any incorrect answers being ranked over correct answers, and the ranking metric is not influenced by specific choices of the threshold hyperparameter that cuts off positive and negative predictions.

6.5.1 Model Performance

Table 6.5 shows model results and human performance. Human performance achieves a near-perfect level (97.80, with Cohen's kappa score of 0.86 showing almost perfect inter-rater agreement [Landis and Koch, 1977]).

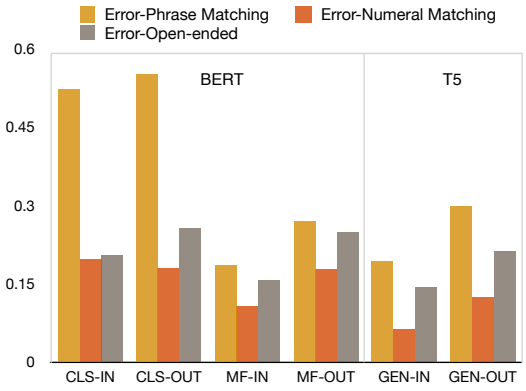


Figure 6.2: Percentage of errors on options created by different rules. CLS, MF, and GEN represent classification, mask-filling, and generation models, respectively; and IN and OUT denote in-domain and out-of-domain training. All models are of large size.

Overall. The generation model based on T5-LARGE and finetuned on the in-domain DailyDialog data achieves the best performance. However, its 2-best accuracy (74.8) lagged far behind the human performance, demonstrating the difficulty of the TIMEDIAL challenge set.

Zero-shot vs. out-of-domain vs. in-domain. When comparing the different training data setup, we observe that models with in-domain training using the DailyDialog data (e.g., LARGE-IN) consistently outperforms those trained on the large out-of-domain Meena dataset (e.g., LARGE-OUT). Both setups outperform the zero-shot models (without any fine-tuning) (e.g., LARGE-ZERO). The results show that the large LMs still highly depend on in-domain or at least dialog data to grasp and enhance their temporal reasoning ability in dialog context. Further, we see increasing performance with increasing model size, which is not unexpected given the complexity of the task.

6.5.2 Error Analysis

Next, we analyze the different types of errors based on different rules for negative option creation in the annotation process. In particular, the *phrase matching* rule picks an exact time span from the dialog context, and *numeral matching* picks numerals from the dialog context. Thus, models picking those incorrect options imply reliance on spurious shallow text matching features.

Figure 6.2 shows the percentage of errors in terms of the different rules. For example, the BERT-based classification model CLS-IN erroneously picks 52% of negative options created by the phrase matching rule

Size	BASE		LARGE	
Training	IN	OUT	IN	OUT
Classification (BERT)				
TARGET	50.5	40.0	50.5	47.5
LOCAL	+ **3.4**	+ **3.3**	+ **7.5**	+ **2.0**
FULL	− 0.6	− 0.1	+ 2.7	+ 1.2
Mask Filling (BERT)				
TARGET	57.8	44.3	60.3	46.8
LOCAL	+ 5.4	+ 3.0	+ 8.1	+ 4.9
FULL	+ 9.6	+ 3.1	+ 9.6	+ 8.0
Generation (T5)				
TARGET	55.5	45.9	66.7	56.1
LOCAL	+ 3.7	+ 2.7	+ 6.1	+ 3.7
FULL	+ 3.7	+ 4.7	+ 8.2	+ 5.8

Table 6.7: Impact of dialog context on reasoning accuracy. IN and OUT denote in-domain and out-of-domain training, respectively. We use *2-best accuracy* of *target* context as reference and report the absolute changes in performance of *local* and *full* context, respectively. Local dialog context results in better performance to full dialog context on 5 of the 12 cases, which are highlighted in the table.

as correct answers (i.e., by ranking those negative options over the true correct options). We observe that the various models are all most vulnerable to the phrase matching options compared to other types of negative options, showing that they rely on spurious text matching to a significant extent. Between BERT and T5, we find T5 being more robust to shallow text matching.

Table 6.6 provides further examples of prediction errors, illustrating confusions due to shallow text matching. In the first dialog, both incorrect answers already partially occur in the context or are related to preexisting concepts (i.e., *"three"* to *"three o'clock"*, and *"nine"* to *"September"*). All the three models were confused and chose either of the two as the top prediction for the blank, even though the options clearly violate the context. Interestingly, the mask filling model was completely confused and ranked both incorrect answers over the correct ones. Similarly in the second example, the models fail to capture the contextual semantics.

6.5.3 Influence of Dialog Context

Table 6.7 shows how different scopes of dialog context (§6.4.2) affect model performance. First, the most restrictive target-only context is insufficient for accurate reasoning, by producing the weakest performance of

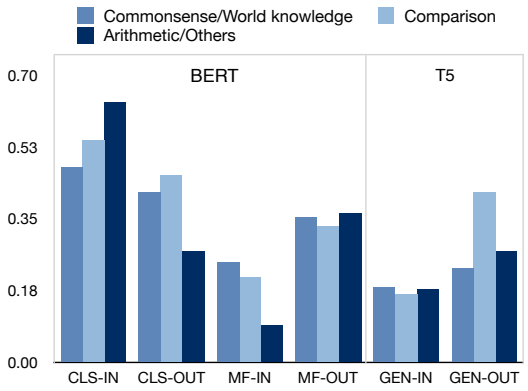


Figure 6.3: Percentage of errors on different reasoning types. CLS, MF, and GEN represent classification, mask-filling, and generation models, respectively. All models are of large size.

most models. This highlights the importance of context information for temporal commonsense reasoning in dialog, which differs from previous temporal reasoning studies based on limited context (e.g., single-sentence question answering). Second, we note that the full dialog context does not always lead to the best performance. In 5 out of the 12 cases, using the local context yields equal or higher reasoning accuracy. The results show that the LMs still fall short of properly modeling the rich dialog contexts and making effective use of all information to do reasoning.

6.5.4 Errors of Reasoning Categories

Figure 6.3 shows the percentage of errors in each reasoning category. We observe that the models tend to make non-trivial portions of errors on commonsense/world knowledge questions. For example, the strongest model, T5 GEN-IN, failed on 18% of the instances that require commonsense or world knowledge, while BERT CLS-IN made errors on 48% of such instances. The performance on comparison-based instances seems similar.

6.6 Related Work

Temporal commonsense reasoning. Early studies related to temporal analysis define time in the context of sets and relations [Bruce, 1972; Allen, 1983]. More recent works often associate time with events and focus on identifying time expressions [Chang and Manning, 2012; Angeli et al., 2012; Lee et al., 2014], extracting temporal relations among events [Setzer and Gaizauskas, 2000; Pustejovsky et al., 2005; Lapata

and Lascarides, 2006; Chambers et al., 2007; Ning et al., 2018b], and timeline construction [Do et al., 2012; Leeuwenberg and Moens, 2018].

Some recent work has focused on building challenging benchmarks for temporal commonsense reasoning. Story Cloze Test focuses on stereotypical causal temporal and causal relations between events [Mostafazadeh et al., 2016a]. Vashishtha et al. [2020] recast temporal reasoning datasets for event duration and event ordering into the natural language inference (NLI) format. Turque [Ning et al., 2020] is an reading comprehension dataset where the model needs to answer questions such as "what happens before/after [event]". Most related to our work is McTaco [Zhou et al., 2019], a dataset for evaluating temporal commonsense in the form of multiple-choice reading comprehension, where the context usually consists of a single sentence. Our work instead studies temporal commonsense reasoning in dialogs which often require significant commonsense and world knowledge to reason over rich context [Qin et al., 2019b; Dinan et al., 2018].

Commonsense reasoning with LMs. With the recent success of large pre-trained language models (LMs) [Devlin et al., 2019; Brown et al., 2020b], it is an open question whether these models, pretrained on large amounts of data, capture commonsense knowledge. Several works have been proposed to assess the ability of LMs for commonsense or numerical reasoning [Zhang et al., 2020b; Bouraoui et al., 2020], or to mine commonsense knowledge from LMs [Davison et al., 2019]. Lin et al. [2020a] showed that state-of-the-art LMs such as BERT and RoBERTa performs poorly on numerical reasoning tasks without any finetuning. Works have also been proposed to improve language model's commonsense reasoning [Qin et al., 2020b, 2019a; Zhou et al., 2020] and numerical reasoning abilities [Geva et al., 2020]. In our work, we study several modeling approaches and finetuning settings of large LMs, and establish strong baselines for temporal commonsense reasoning in dialogs.

6.7 Conclusions

We introduced TIMEDIAL, a challenge set consistting of 1.1K multiple-choice cloze questions for temporal commonsense reasoning in dialog. The dataset is carefully curated to evaluate a models' ability to do temporal commonsense/numerical reasoning over dialog context. In order to establish strong baselines and provide information on future model development, we conducted extensive experiments with state-of-the-

art language models with different settings: the scope of context, weak supervision strategies, and learning objectives. While humans can easily answer these questions (97.8% accuracy), even our best model variant (T5-large with in-domain training) struggles on this challenge set (73%). Moreover, our qualitative error analyses show that these large language models often rely on shallow, spurious features (particularly text matching) when answering these questions, instead of truly doing reasoning over the context.

6.8 Appendix

We provide all model and training configurations used across our experiments:

BERT Experiments for Classification and Mask-Filling

- Model configuration for BERT-BASE classification and mask-filling:

```
attention_dropout_rate: 0.1
dropout_rate: 0.1
hidden_activation: gelu
hidden_size: 768
initializer_range: 0.02
intermediate_size: 3072
max_position_embeddings: 512
num_attention_heads: 12
num_layers: 12
type_vocab_size: 2
vocab_size: 30522
```

- Model configuration for BERT-LARGE classification and mask-filling:

```
attention_dropout_rate: 0.1
dropout_rate: 0.1
hidden_activation: gelu
hidden_size: 1024
initializer_range: 0.02
```

```
intermediate_size: 4096
max_position_embeddings: 512
num_attention_heads: 16
num_layers: 24
type_vocab_size: 2
vocab_size: 30522
```

- Training configuration for classification with BERT-BASE and in-domain data:

```
num_classes: 2
train_data:
  global_batch_size: 128
  seq_length: 512
validation_data:
  global_batch_size: 32
  seq_length: 512
trainer:
  max_to_keep: 3
  checkpoint_interval: 1000
  decay_steps: 30000
  end_learning_rate: 0.0
  initial_learning_rate: 1.0e-5
  power: 1.0
  optimizer: adam
  warmup_steps: 5000
  steps_per_loop: 1000
  train_steps: 30000
  validation_steps: 256
```

- Training configuration for classification with BERT-LARGE and in-domain data:

```
num_classes: 2
train_data:
  global_batch_size: 128
```

```
  seq_length: 512
validation_data:
  global_batch_size: 32
  seq_length: 512
trainer:
  max_to_keep: 3
  checkpoint_interval: 1000
  decay_steps: 100000
  end_learning_rate: 0.0
  initial_learning_rate: 1.0e-6
  power: 1.0
  optimizer: adam
  warmup_steps: 10000
  steps_per_loop: 1000
  train_steps: 100000
  validation_steps: 3000
```

- Training configuration for classification with BERT-BASE and out-domain data:

```
num_classes: 2
train_data:
  global_batch_size: 128
  seq_length: 512
validation_data:
  global_batch_size: 128
  seq_length: 512
trainer:
  max_to_keep: 3
  checkpoint_interval: 5000
  decay_steps: 500000
  end_learning_rate: 0.0
  initial_learning_rate: 1.0e-6
  power: 1.0
```

```
  optimizer: adam
  warmup_steps: 10000
  steps_per_loop: 1000
  train_steps: 500000
  validation_steps: 512
```

- Training configuration for classification with BERT-LARGE and out-domain data:

```
num_classes: 2
train_data:
  global_batch_size: 128
  seq_length: 512
validation_data:
  global_batch_size: 128
  seq_length: 512
trainer:
  max_to_keep: 3
  checkpoint_interval: 5000
  decay_steps: 500000
  end_learning_rate: 0.0
  initial_learning_rate: 1.0e-6
  power: 1.0
  optimizer: adam
  warmup_steps: 10000
  steps_per_loop: 1000
  train_steps: 500000
  validation_steps: 512
```

- Training configuration for mask-filling with BERT-BASE and in-domain data:

```
train_data:
  global_batch_size: 128
  seq_length: 512
  max_predictions_per_seq: 20
```

```
validation_data:
  global_batch_size: 128
  seq_length: 512
  max_predictions_per_seq: 20
trainer:
  checkpoint_interval: 2000
  max_to_keep: 30
  decay_steps: 30000
  end_learning_rate: 0.0
  initial_learning_rate: 1.0e-8
  power: 1.0
  optimizer: adam
  warmup_steps: 5000
  steps_per_loop: 1000
  train_steps: 30000
  validation_interval: 1000
```

- Training configuration for mask-filling with BERT-LARGE and in-domain data:

```
train_data:
  global_batch_size: 128
  seq_length: 512
  max_predictions_per_seq: 20
validation_data:
  global_batch_size: 128
  seq_length: 512
  max_predictions_per_seq: 20
trainer:
  checkpoint_interval: 2000
  max_to_keep: 30
  decay_steps: 30000
  end_learning_rate: 0.0
  initial_learning_rate: 1.0e-8
```

```
  power: 1.0
  optimizer: adam
  warmup_steps: 5000
  steps_per_loop: 1000
  train_steps: 30000
  validation_interval: 1000
```

- Training configuration for mask-filling with BERT-BASE and out-domain data:

```
train_data:
  global_batch_size: 512
  seq_length: 512
validation_data:
  global_batch_size: 512
  seq_length: 512
trainer:
  checkpoint_interval: 5000
  max_to_keep: 10
  decay_steps: 300000
  end_learning_rate: 0.0
  initial_learning_rate: 1.0e-6
  power: 1.0
  optimizer: adam
  warmup_steps: 10000
  steps_per_loop: 1000
  train_steps: 300000
  validation_steps: 1000
```

- Training configuration for mask-filling with BERT-LARGE and out-domain data:

```
train_data:
  global_batch_size: 512
  seq_length: 512
validation_data:
```

```
    global_batch_size: 512
    seq_length: 512
  trainer:
    checkpoint_interval: 5000
    max_to_keep: 10
    decay_steps: 300000
    end_learning_rate: 0.0
    initial_learning_rate: 1.0e-6
    power: 1.0
    optimizer: adam
    warmup_steps: 10000
    steps_per_loop: 1000
    train_steps: 300000
    validation_steps: 1000
```

T5 Experiments for Generation

- The training configuration for generation with T5-BASE and in-domain data:

```
encoder_seq_length: 512
decoder_max_length: 128
train_batch_size: 128
max_train_steps: 100000
valid_batch_size: 128
dropout_rate: 0.2
optimizer: adam
learning_rate: 1.0e-6
```

- The training configurations for generation with T5-BASE/LARGE and in-domain/out-domain data are similar as above, except that the `learning rate` is set to `5.0e-6` for T5-LARGE in-domain data, `5.0e-4` for T5-BASE out-domain data, and `1.0e-4` for T5-LARGE out-domain data.

Chapter 7

Conclusion and Future Work

In this dissertation, we developed efficient inference and learning algorithms to enable (pretrained) language models to do constrained, causal, and logical reasoning. The three aspects of the reasoning capabilities are the foundations to enormous applications of language models in the real world. To achieve this goal, we established new formulations and benchmarks to enable large-scale computational studies of the language model reasoning capabilities, including counterfactual reasoning (Chapters 3 and 4) and temporal reasoning (Chapter 6), as well as social reasoning as studied in [Sap et al., 2020; Zellers et al., 2021]. We developed new inference (decoding) algorithms for language model reasoning, including differentiable reasoning over symbolic text (Chapters 2 and 4), as wel as discrete reasoning with efficient structured search as discussed in [Jung et al., 2022; Lu et al., 2022b]. We also studied rich applications of neural language reasoning and generation, including (temporal) knowledge-rich dialog systems (Chapters 5 and 6), as well as language detoxification (with reinforcement learning) [Lu et al., 2022a], summarization [Tan et al., 2020], advice giving [Zellers et al., 2021], and so forth.

Reasoning and generation in language is really an important area that has a lot of exciting directions for future study. Here I discuss some of the topics.

Reasoning between and beyond the lines. Natural language involves rich hidden information not explicitly stated in the text. Understanding and reasoning over the text thus require the ability to read *between* the lines. Consider the text *"Ellie looks in the mirror. The lipstick looks right and she adjusts the new woolly scarf a bit."* We can infer that it is cold weather, probably winter. We also know that Ellie can see

her own reflection with the mirror. Even further, humans can imagine *beyond* the text: why is she wearing lipstick and a new scarf? Is she going to a party or date? Is she feeling excited or nervous? Reasoning for all the hidden facts and questions requires connecting and generalizing the reasoning abilities I have studied. I am excited to develop the flexible reasoning mechanism to infer the rich implicature of text, understand the likely cause-effects of events and more abstract aspects (e.g., intents and emotions), and forecast diverse futures creatively with dynamic uncertainty as more context is observed.

Interactive reasoning between language and physical world. Human intelligence has the hallmark of interacting sophisticated language with the complex physical world. On one hand, grounding on the physical world allows efficient communication and collaboration with language. On the other hand, language offers a powerful abstraction for *compositional* reasoning over concepts (e.g., "a *man* holding a *mobile phone* is riding a *horse*"), which substantially improves the efficiency of understanding complex situations and tasks. Compared to the current ungrounded large language models, I am interested in building a new language and reasoning engine that connects language with massive world concepts in a variety of modalities (e.g., knowledge bases, images/videos, interactive environments). Given a question or a task, the engine will localize the relevant subset of concepts out of the extreme-scale external knowledge, unfold the reasoning with generated text as the scaffold, invoke external functions and actions (e.g., calculator, navigator), and consolidate all the information to produce the final reasoning results.

Building trustworthy AI assistants. Commonsense knowledge and reasoning forms the necessary basis for building AI agents that communicate with and assist human in a trustworthy way. I have been involved in developing learning and inference approaches to understanding bias implications in text, language detoxification, and advice giving [Lu et al., 2022a; Sap et al., 2020; Zellers et al., 2021], demonstrating the potential of language-based generative reasoning in building robust, fair, controllable, and helpful machines. Besides, today's AI agents still lack the abilities of explaining reliably their own prediction behaviors, and recognizing faithfully what they do not know (instead of hallucinating spurious outputs). Those problems collectively pose challenges for building trustworthy machines for individual and societal use. They demand interdisciplinary collaborations across different fields like human-computer interaction, computer security, cognitive science, public policy, etc. I am devoted to expanding and deepening my world-grounded language-driven reasoning research in collaboration with the diverse experts to strive toward this goal.

Bibliography

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Shubham Agarwal, Ondrej Dusek, Ioannis Konstas, and Verena Rieser. 2018. A knowledge-grounded multimodal search-based conversational agent. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 59–66, Brussels, Belgium.

Ekaterina Ageeva, Mikel L Forcada, Francis M Tyers, and Juan Antonio Pérez-Ortiz. 2015. Evaluating machine translation for assimilation via a gap-filling task. In *Proceedings of the 18th Annual Conference of the European Association for Machine Translation*, pages 137–144.

James F Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.

James F Allen. 1984. Towards a general theory of action and time. *Artificial intelligence*, 23(2):123–154.

Henning Andersen. 1973. Abductive and deductive change. *Language*, pages 765–793.

Gabor Angeli, Christopher D Manning, and Dan Jurafsky. 2012. Parsing time: Learning to interpret time expressions. In *Proc. of NAACL*.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*.

Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xinyu Dai, and Jiajun Chen. 2019. Generating sentences from disentangled syntactic and semantic spaces. *arXiv preprint arXiv:1907.05789*.

Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. 2019. Seq3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. In *NAACL-HLT*.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2019a. Abductive commonsense reasoning. In *International Conference on Learning Representations* (ICLR).

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2019b. Abductive commonsense reasoning. In *International Conference on Learning Representations*.

Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 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.

Leon Bottou. 2019. Learning representations using causal invariance. In *ICLR*.

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *JMLR*.

Zied Bouraoui, José Camacho-Collados, and S. Schockaert. 2020. Inducing relational knowledge from bert. In *Proc. of AAAI*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, and et al. 2020a. Language models are few-shot learners. In *Advances in Neural Information Processing Systems* (NourIPS).

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language models are few-shot learners.

Bertram C Bruce. 1972. A model for temporal references and its application in a question answering program. *Artificial intelligence*, 3:1–25.

Ruth MJ Byrne. 2002. Mental models and counterfactual thoughts about what might have been. *Trends in cognitive sciences*, 6(10):426–431.

Nathanael Chambers. 2013. Event schema induction with a probabilistic entity-driven model. In *EMNLP*, pages 1797–1807.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797.

Nathanael Chambers, Shan Wang, and Dan Jurafsky. 2007. Classifying temporal relations between events. In *Proc. of ACL*.

Angel X Chang and Christopher D Manning. 2012. Sutime: A library for recognizing and normalizing time expressions. In *Proc. of LREC*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, and Michael Petrov et al. 2021. Evaluating large language models trained on code.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184.

Elizabeth Clark, Asli Celikyilmaz, and Noah A. Smith. 2019. Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In *ACL*.

Rémi Coulom. 2006. Efficient selectivity and backup operators in monte-carlo tree search. In *International Conference on Computers and Games* (ICCG), pages 72–83. Springer.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M.F. Moura, Devi Parikh, and Dhruv Batra. 2017. Visual Dialog. In *CVPR*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations* (ICLR).

Joe Davison, Joshua Feldman, and Alexander Rush. 2019. Commonsense knowledge mining from pretrained models. In *Proc. of EMNLP*.

Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. 2020. Residual energy-based models for text generation. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018a. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018b. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. In *Proc. of ICLR*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *ICLR*.

Quang Do, Wei Lu, and Dan Roth. 2012. Joint inference for event timeline construction. In *Proc. of EMNLP*.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proc. of HLT*.

C. Donahue, M. Lee, and P. Liang. 2020. Enabling language models to fill in the blanks. In *Annual Meeting of the Association for Computational Linguistics* (ACL).

Yilun Du and Igor Mordatch. 2019. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems* (NourIPS), 32.

David Duvenaud, Jacob Kelly, Kevin Swersky, Milad Hashemi, Mohammad Norouzi, and Will Grathwohl. 2021. No MCMC for me: Amortized samplers for fast and stable training of energy-based models. In *International Conference on Learning Representations* (ICLR).

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proc. of EMNLP*.

Bryan Eikema and Wilker Aziz. 2021. Sampling-based minimum bayes risk decoding for neural machine translation.

Kai Epstude and Neal J Roese. 2008. The functional theory of counterfactual thinking. *Personality and Social Psychology Review*, 12(2):168–192.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Annual Meeting of the Association for Computational Linguistics* (ACL), pages 889–898.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104.

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

Michel Galley, Chris Brockett, Xiang Gao, Jianfeng Gao, and Bill Dolan. 2019. Grounded response generation task at DSTC7. In *AAAI Dialog System Technology Challenges Workshop*.

Jianfeng Gao, Michel Galley, and Lihong Li. 2019a. Neural approaches to conversational ai. *Foundations and Trends in Information Retrieval*, 13(2-3):127–298.

Xiang Gao, Sungjin Lee, Yizhe Zhang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2019b. Jointly optimizing diversity and relevance in neural response generation. In *NAACL-HLT 2019*.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proc. of ACL*.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Proc. of AAAI*.

Matthew L Ginsberg. 1986. Counterfactuals. *Artificial intelligence*, 30(1):35–79.

Nelson Goodman. 1947. The problem of counterfactual conditionals. *The Journal of Philosophy*, 44(5):113–128.

Kartik Goyal, Chris Dyer, and Taylor Berg-Kirkpatrick. 2021. Exposing the implicit energy networks behind masked language models via metropolis–hastings. *arXiv preprint arXiv:2106.02736*.

Mark Granroth-Wilding and Stephen Clark. 2016. What happens next? event prediction using a compositional neural network model. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *Proc. of NAACL*, pages 107–112.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics* (TACL), 6:437–450.

Geoffrey E. Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation*.

Cong Duy Vu Hoang, Gholamreza Haffari, and Trevor Cohn. 2017. Towards decoding as continuous optimisation in neural machine translation. In *Conference on Empirical Methods in Natural Language Processing* (EMNLP), pages 146–156.

Jerry R Hobbs. 2005. Toward a useful concept of causality for lexical semantics. *Journal of Semantics*, 22(2):181–209.

Jerry R Hobbs, Mark E Stickel, Douglas E Appelt, and Paul Martin. 1993. Interpretation as abduction. *Artificial intelligence*, 63(1-2):69–142.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Annual Meeting of the Association for Computational Linguistics* (ACL), pages 1535–1546.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Zhiting Hu, BowenTan HaoranShi, ZichaoYang WentaoWang, Tiancheng Zhao, Junxian He, Lianhui Qin, Di Wang, Xuezhe Ma, Zhengzhong Liu, Xiaodan Liang, et al. 2019. Texar: A modularized, versatile, and extensible toolkit for text generation. *ACL 2019*, page 159.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International Conference on Machine Learning* (ICML), pages 1587–1596.

Zhiting Hu, Zichao Yang, Russ R Salakhutdinov, Lianhui Qin, Xiaodan Liang, Haoye Dong, and Eric P Xing. 2018. Deep generative models with learnable knowledge constraints. *Advances in Neural Information Processing Systems (NeurIPS)*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019a. 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.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2019b. Challenges in building intelligent open-domain dialog systems. *arXiv preprint arXiv:1905.05709*.

Steve D Isard. 1974. What would you have done if...? *Theoretical Linguistics*, 1(1-3):233–256.

Vivek Jayaram and John Thickstun. 2021. Parallel and flexible sampling from autoregressive models via Langevin dynamics. In *International Conference on Machine Learning* (ICML), pages 4807–4818. PMLR.

Philip Nicholas Johnson-Laird. 2006. *How we reason*. Oxford University Press, USA.

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic Prompting: Logically Consistent Reasoning with Recursive Explanations. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Dan Jurafsky and James H Martin. 2009. *Speech & language processing*. Prentice Hall.

Kenneth Kahn and G.Anthony Gorry. 1977. Mechanizing temporal knowledge. *Artificial Intelligence*, 9(1):87 – 108.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. 2020. A distributional approach to controlled text generation. In *International Conference on Learning Representations*.

Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. 2017. Re-evaluating automatic metrics for image captioning. In *EACL*.

Rik Koncel-Kedziorski, Ioannis Konstas, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2016a. A theme-rewriting approach for generating algebra word problems. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1617–1628.

Rik Koncel-Kedziorski, Ioannis Konstas, Luke S. Zettlemoyer, and Hannaneh Hajishirzi. 2016b. A theme-rewriting approach for generating algebra word problems. In *EMNLP*.

Z. Kozareva and E. Hovy. 2011. Learning temporal information for states and events. In *2011 IEEE Fifth International Conference on Semantic Computing*, pages 424–429.

Sarit Kraus, Daniel Lehmann, and Menachem Magidor. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial intelligence*, 44(1-2):167–207.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Conference on Empirical Methods in Natural Language Processing* (EMNLP) - *Findings*, pages 4929–4952.

Klaus Krippendorff. 2007. Computing krippendorff's alpha reliability. *Departmental papers (ASC)*, page 43.

Sachin Kumar, Eric Malmi, Aliaksei Severyn, and Yulia Tsvetkov. 2021. Controlled text generation as continuous optimization with multiple constraints. *Advances in Neural Information Processing Systems* (NourIPS), 34.

Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From word embeddings to document distances. In *ICML*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. *arXiv preprint arXiv:1804.07755*.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019a. Multiple-attribute text rewriting. In *ICLR*.

Guillaume Lample, Sandeep Subramanian, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019b. Multiple-attribute text rewriting. In *ICLR*.

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

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Mirella Lapata and Alex Lascarides. 2006. Learning sentence-internal temporal relations. *Journal of Artificial Intelligence Research*, 27:85–117.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proc. of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 228–231.

Carolin Lawrence and Stefan Riezler. 2018a. Improving a neural semantic parser by counterfactual learning from human bandit feedback. In *ACL*.

Carolin Lawrence and Stefan Riezler. 2018b. Improving a neural semantic parser by counterfactual learning from human bandit feedback. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1820–1830.

Carolin Lawrence, Artem Sokolov, and Stefan Riezler. 2017. Counterfactual learning from bandit feedback under deterministic logging: A case study in statistical machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2566–2576.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *Proc. of ICML*, pages 1078–1088. PMLR.

Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. 2006. A Tutorial on Energy-Based Learning. *Predicting Structured Data*.

Ann Lee, Michael Auli, and Marc'Aurelio Ranzato. 2021. Discriminative reranking for neural machine translation. In *Annual Meeting of the Association for Computational Linguistics* (ACL), pages 7250–7264, Online.

Kenton Lee, Yoav Artzi, Jesse Dodge, and Luke Zettlemoyer. 2014. Context-dependent semantic parsing for time expressions. In *Proc. of ACL*.

Artuur Leeuwenberg and Marie Francine Moens. 2018. Temporal information extraction by predicting relative time-lines. In *Proc. of EMNLP*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proc. of NAACL-HLT*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016b. A persona-based neural conversation model. In *Proc. of ACL*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Constructing narrative event evolutionary graph for script event prediction. In *IJCAI*.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020a. Birds have four legs?! NumerSense: Probing Numerical Commonsense Knowledge of Pre-Trained Language Models. In *Proc. of EMNLP*.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020b. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Conference on Empirical Methods in Natural Language Processing* (EMNLP) - *Findings*.

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

Shuai Lin, Wentao Wang, Zichao Yang, Xiaodan Liang, Frank F Xu, Eric Xing, and Zhiting Hu. 2020c. Record-to-text generation with style imitation. In *Conference on Empirical Methods in Natural Language Processing* (EMNLP) - *Findings*, pages 1589–1598.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016a. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *EMNLP*.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016b.

How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proc. of EMNLP*.

Guangyi Liu, Zichao Yang, Tianhua Tao, Xiaodan Liang, Zhen Li, Bowen Zhou, Shuguang Cui, and Zhiting Hu. 2021. Don't take it literally: An edit-invariant sequence loss for text generation. *arXiv preprint arXiv:2106.15078*.

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

Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2018b. Stochastic Answer Networks for machine reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1694–1704, Melbourne, Australia.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ryan Lowe, Michael Noseworthy, Iulian Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. In *ICLR*.

Ximing Lu, Sean Welleck, Liwei Jiang, Jack Hessel, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022a. Quark: Controllable Text Generation with Reinforced Unlearning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, et al. 2022b. Neurologic A*esque decoding: Constrained text generation with lookahead heuristics. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Neurologic decoding:(un) supervised neural text generation with predicate logic constraints. In *Conference*

of the North American Chapter of the Association for Computational Linguistics (NAACL), pages 4288–4299.

Thang Luong, Hieu Pham, and Christopher D. Manning. 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, Lisbon, Portugal.

Yi-An Ma, Yuansi Chen, Chi Jin, Nicolas Flammarion, and Michael I Jordan. 2019. Sampling can be faster than optimization. *Proceedings of the National Academy of Sciences* (PNAS), 116(42):20881–20885.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6297–6308.

Hugo Mercier and Dan Sperber. 2017. *The enigma of reason*. Harvard University Press.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019a. CGMH: Constrained sentence generation by metropolis-hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6834–6842.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019b. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6834–6842.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proc. of ACL*.

Fatemehsadat Mireshghallah, Kartik Goyal, and Taylor Berg-Kirkpatrick. 2022. Mix and match: Learning-free controllable text generation using energy language models. *arXiv preprint arXiv:2203.13299*.

Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M Khapra. 2018. Towards exploiting background knowledge for building conversation systems. In *Proc. of EMNLP*.

Nasrin Mostafazadeh, Chris Brockett, Bill Dolan, Michel Galley, Jianfeng Gao, Georgios Spithourakis, and Lucy Vanderwende. 2017. Image-grounded conversations: Multimodal context for natural question and response generation. In *Proc. of IJCNLP*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016a. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proc. of NAACL*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016b. A corpus and evaluation framework for deeper understanding of commonsense stories. *arXiv preprint arXiv:1604.01696*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. 2016c. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *HLT-NAACL*.

Radford M Neal et al. 2011. MCMC using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2.

Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. 2020. On the anatomy of MCMC-based maximum likelihood learning of energy-based models. In *Conference on Artificial Intelligence* (AAAI), volume 34, pages 5272–5280.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018a. Joint reasoning for temporal and causal relations. In *Proc. of ACL*.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020. TORQUE: A Reading Comprehension Dataset of Temporal Ordering Questions. In *Proc. of EMNLP*.

Qiang Ning, Hao Wu, Haoruo Peng, and Dan Roth. 2018b. Improving temporal relation extraction with a globally acquired statistical resource. In *Proc. of NAACL*.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355*.

Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. In *Proc. of EMNLP*.

Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. 2020. Learning latent space energy-based prior model. *Advances in Neural Information Processing Systems*, 33:21994–22008.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics* (ACL), pages 311–318.

Tetiana Parshakova, Jean-Marc Andreoli, and Marc Dymetman. 2019. Global autoregressive models for data-efficient sequence learning. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 900–909.

Damian Pascual, Beni Egressy, Florian Bolli, and Roger Wattenhofer. 2020. Directed beam search: Plug-and-play lexically constrained language generation. *arXiv preprint arXiv:2012.15416*.

Judea Pearl. 2000. *Causality: models, reasoning and inference*, volume 29. Springer.

Judea Pearl and Dana Mackenzie. 2018. *The book of why: the new science of cause and effect*. Basic Books.

Charles Sanders Peirce. 1960. *Collected papers of charles sanders peirce*, volume 2. Harvard University Press.

Charles Sanders Peirce. 1974. *Collected papers of charles sanders peirce*, volume 2. Harvard University Press.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Karl Pichotta and Raymond Mooney. 2014. Statistical script learning with multi-argument events. In *EACL*, pages 220–229.

James Pustejovsky. 2017. Iso-timeml and the annotation of temporal information. In *Handbook of Linguistic Annotation*, pages 941–968. Springer.

James Pustejovsky, Robert Knippen, Jessica Littman, and Roser Saurí. 2005. Temporal and event information in natural language text. *Language resources and evaluation*, 39(2):123–164.

Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. 2019a. Counterfactual Story Reasoning and Generation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Lianhui Qin, Michel Galley, Chris Brockett, Xiaodong Liu, Xiang Gao, William B Dolan, Yejin Choi, and Jianfeng Gao. 2019b. Conversing by Reading: Contentful Neural Conversation with On-demand Machine Reading. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Lianhui Qin, Aditya Gupta, Shyam Upadhyay, Luheng He, Yejin Choi, and Manaal Faruqui. 2021. TimeDial: Temporal Commonsense Reasoning in Dialog. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2020a. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena D Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2020b. Back to the future: Backpropagation-based decoding for unsupervised counterfactual and abductive reasoning. In *Proc. of EMNLP*.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. 2022. COLD decoding: Energy-based constrained text generation with langevin dynamics. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019a. Language models are unsupervised multitask learners. -.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019b. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Gabriel A Radvansky and Jeffrey M Zacks. 2014. *Event cognition*. Oxford University Press.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas.

Hannah Rashkin, Antoine Bosselut, Maarten Sap, Kevin Knight, and Yejin Choi. 2018. Modeling naive psychology of characters in simple commonsense stories. *arXiv preprint arXiv:1805.06533*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association of Computational Linguistics (TACL)*.

Raymond Reiter. 1988. Nonmonotonic reasoning. In *Exploring artificial intelligence*, pages 439–481. Elsevier.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011a. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011b. Semeval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *SemEval@NAACL-HLT*.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A Smith, and Yejin Choi. 2020. Social Bias Frames: Reasoning about Social and Power Implications of Language. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Maarten Sap, Ronan LeBras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. ATOMIC: An atlas of machine commonsense for if-then reasoning. In *AAAI*.

Roger C Schank and Robert P Abelson. 1977. Scripts, plans, goals and understanding: An inquiry into human knowledge structures.

Raphael Schumann, Lili Mou, Yao Lu, Olga Vechtomova, and Katja Markert. 2020. Discrete optimization for unsupervised sentence summarization with word-level extraction. *arXiv preprint arXiv:2005.01791*.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *ICLR*.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. *AAAI*.

Andrea Setzer and Robert J Gaizauskas. 2000. Annotating events and temporal information in newswire texts. In *Proc. of LREC*.

Lei Sha. 2020. Gradient-guided unsupervised lexically constrained text generation. In *Conference on Empirical Methods in Natural Language Processing* (EMNLP), pages 8692–8703.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proc. of ACL-IJCNLP*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *NeurIPS*.

Youngseo Son, Anneke Buffone, Joe Raso, Allegra Larche, Anthony Janocko, Kevin Zembroski, H Andrew Schwartz, and Lyle Ungar. 2017a. Recognizing counterfactual thinking in social media texts. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 654–658.

Youngseo Son, Anneke Buffone, Joe Raso, Allegra Larche, Anthony Janocko, Kevin Zembroski, H. Andrew Schwartz, and Lyle H. Ungar. 2017b. Recognizing counterfactual thinking in social media texts. In *ACL*.

Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems* (NourIPS), 32.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proc. of NAACL-HLT*.

Martin Šošić and Mile Šikić. 2017. Edlib: a c/c++ library for fast, exact sequence alignment using edit distance. *Bioinformatics*, 33(9):1394–1395.

William Starr. 2019. Counterfactuals. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, fall 2019 edition. Metaphysics Research Lab, Stanford University.

Sainbayar Sukhbaatar, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Proc. of NIPS*.

Qing Sun, Stefan Lee, and Dhruv Batra. 2017. Bidirectional beam search: Forward-backward inference in neural sequence models for fill-in-the-blank image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6961–6969.

Ilya Sutskever, Oriol Vinyals, and Quoc Le. 2014. Sequence to sequence learning with neural networks. In *Proc. of NIPS*, pages 3104–3112.

Bowen Tan, Lianhui Qin, Eric Xing, and Zhiting Hu. 2020. Summarizing Text on Any Aspects: A Knowledge-Informed Weakly-Supervised Approach. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Niket Tandon, Bhavana Dalvi Mishra, Keisuke Sakaguchi, Antoine Bosselut, and Peter Clark. 2019. Wiqa: A dataset for" what if..." reasoning over procedural text. In *EMNLP*.

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

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In *Proc. of SemEval*.

Siddharth Vashishtha, Adam Poliak, Yash Kumar Lal, Benjamin Van Durme, and Aaron Steven White. 2020. Temporal reasoning in natural language inference. In *Proc. of Findings of EMNLP*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz

Kaiser, and Illia Polosukhin. 2017a. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017b. Attention is all you need. In *Advances in Neural Information Processing Systems* (NourIPS).

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), pages 4566–4575.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. In *ICML Deep Learning Workshop*.

Sean Welleck, Jiacheng Liu, Jesse Michael Han, and Yejin Choi. 2021. Towards grounded natural language proof generation. *MathAI4Ed Workshop at NeurIPS*.

Max Welling and Yee W Teh. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning (ICML)*.

Peter West, Ari Holtzman, Jan Buys, and Yejin Choi. 2019. Bottlesum: Unsupervised and self-supervised sentence summarization using the information bottleneck principle. 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 3743–3752.

Peter West, Ximing Lu, Ari Holtzman, Chandra Bhagavatula, Jena D. Hwang, and Yejin Choi. 2021. Reflective decoding: Beyond unidirectional generation with off-the-shelf language models. 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 1435–1450, Online. Association for Computational Linguistics.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017. Challenges in data-to-document generation. In *EMNLP*.

Jim Woodward. 2002. What is a mechanism? a counterfactual account. *Philosophy of Science*, 69(S3):S366–S377.

Jianwen Xie, Yifei Xu, Zilong Zheng, Song-Chun Zhu, and Ying Nian Wu. 2021. Generative pointnet: Deep energy-based learning on unordered point sets for 3D generation, reconstruction and classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14976–14985.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online. Association for Computational Linguistics.

Liu Yang, Junjie Hu, Minghui Qiu, Chen Qu, Jianfeng Gao, W Bruce Croft, Xiaodong Liu, Yelong Shen, and Jingjing Liu. 2019. A hybrid retrieval-generation neural conversation model. *arXiv preprint arXiv:1904.09068*.

Koichiro Yoshino, Chiori Hori, Julien Perez, Luis Fernando D'Haro, Lazaros Polymenakos, R. Chulaka Gunasekara, Walter S. Lasecki, Jonathan K. Kummerfeld, Michel Galley, Chris Brockett, Jianfeng Gao, Bill Dolan, Xiang Gao, Huda AlAmri, Tim K. Marks, Devi Parikh, and Dhruv Batra. 2019. Dialog system technology challenge 7. In *In NeurIPS Conversational AI Workshop*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. QANet: Combining local convolution with global self-attention for reading comprehension. In *ICLR*.

Yoel Zeldes, Dan Padnos, and Barak Peleg. 2020. Haim-1.5 - the next generation.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *EMNLP*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019a. HellaSwag: Can a machine really finish your sentence? In *ACL*.

Rowan Zellers, Ari Holtzman, Elizabeth Clark, Lianhui Qin, Ali Farhadi, and Yejin Choi. 2021. TuringAdvice: A Generative and Dynamic Evaluation of Language Use. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019b. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, pages 9051–9062.

Maosen Zhang, Nan Jiang, Lei Li, and Yexiang Xue. 2020a. Language generation via combinatorial constraint satisfaction: A tree search enhanced monte-carlo approach. In *Learning Meets Combinatorial Algorithms at NeurIPS2020*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018a. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019a. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations* (ICLR).

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019b. BERTScore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675.

Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yanai Elazar, and Dan Roth. 2020b. Do language embeddings capture scales? In *Proc. of Findings of EMNLP*.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018b. Generating informative and diverse conversational responses via adversarial information maximization. In *Proc. of NeurIPS*.

Yizhe Zhang, Xiang Gao, Sungjin Lee, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2019c. Consistent dialogue generation with self-supervised feature learning. *arXiv preprint arXiv:1903.05759*.

Yang Zhao, Jianwen Xie, and Ping Li. 2021. Learning energy-based generative models via coarse-to-fine expanding and sampling. In *International Conference on Learning Representations*.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *Proc. of EMNLP*.

Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. 2020. Temporal common sense acquisition with minimal supervision. In *Proc. of ACL*.

Wanrong Zhu, Zhiting Hu, and Eric Xing. 2019. Text infilling. *arXiv preprint arXiv:1901.00158*.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan R. Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *ICCV*, pages 19–27.