UNIVERSITY OF CALIFORNIA

Los Angeles

Commonsense-Guided Text Generation with Knowledge Grounding and Scoring

A thesis submitted in partial satisfaction

of the requirements for the degree

Master of Science in Computer Science

by

Felix Zhang

2023

ABSTRACT OF THE THESIS

Commonsense-Guided Text Generation with Knowledge Grounding and Scoring

by

Felix Zhang
Master of Science in Computer Science
University of California, Los Angeles, 2023
Professor Nanyun Peng, Chair

This thesis investigates improving the world knowledge and commonsense reasoning abilities of Language Models (LMs) such as GPT2 and T5 (Radford et al., 2019; Raffel et al., 2020) through the task of commonsense language generation using the CommonGen benchmark (Lin et al., 2020). We propose a framework that guides pretrained LMs to generate more commonsensical sentences without updating the LMs' parameters. To do so, we introduce an automatic commonsense metric grounded on ConceptNet (Speer et al., 2017) inspired by ACCENT (Ghazarian et al., 2023). To this end, we introduce a parser to extract triplets of commonsense-related concepts from a input sentence trained on few-shot GPT3-annotated data. We take the extracted triplets and compute similarity scores using COMET (Bosselut et al., 2019) to measure how well the sentence is grounded to ConceptNet, which we assume as the oracle of commonsense knowledge. Finally, we extend the Neurally-Decomposed Oracle by Meng et al. (2022), adding our commonsense metric masked with the lexical constraint into the signal used to train the auxiliary network, and demonstrate our framework is able to guide LMs towards more commonsensical generations while satisfying lexical constraints.

The thesis of Felix Zhang is approved.

Yingnian Wu

Kai-Wei Chang

Nanyun Peng, Committee Chair

University of California, Los Angeles

2023

To my loving parents.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am extremely grateful for Yufei Tian and Professor Violet Peng for their continued mentorship and support throughout this project; their guidance and feedback was invaluable. I additionally would like to thank Sidi Lu and Sarik Ghazarian for their advice and foundational work on NADO and ACCENT, which inspire and inform this thesis. I would also like to thank PLUS Lab for access to compute and resources.

CHAPTER 1

Introduction

Natural Language Processing (NLP) has made significant strides in recent years. Large Language Models like GPT-3 (Brown et al., 2020) can now pass the sections of the Bar Exam (Bommarito and Katz, 2022), solve challenging Math Olympiad problems (Polu et al., 2022), and demonstrates mastery across a variety of difficult benchmarks (Brown et al., 2020).

While these models have made remarkable progress, there have been criticisms around its commonsense reasoning ability (Marcus, 2020), lack of world knowledge (Elazar et al., 2021), and inability to reliably and robustly incorporate commonsense knowledge (Mahowald et al., 2023). Specifically, Mahowald et al. (2023) points to its weaknesses in functional competence relating to language use to do things in the world, covering skills such as world knowledge and formal reasoning.

Our contributions explore addressing this gap in the functional competence area of world knowledge using the task of commonsense generation: generating a plausible sentence of an everyday scenario that obeys our underlying commonsense knowledge.

We explore this question using CommonGen (Lin et al., 2020), which is a constrained text generation task and benchmark designed explicitly to test for generative commonsense reasoning. The benchmark contains pairs of concept sets and their corresponding reference sentences. Each concept set is composed of a reasonable set of concepts that can be composed into a sentence and each reference sentence contains all the corresponding concepts. The objective is to assemble the concepts in the concept set into a plausible sentence defined by similarity to reference sentences. More detail can be found in Section 4.1.

Various systems have been proposed for generative commonsense reasoning on CommonGen. One type are systems like KG-BART (Liu et al., 2021) and DKMRˆ2 (He et al., 2022) that directly integrate an external knowledge corpus through retrieval or as inputs. These systems can be highly effectively, but require costly access to Knowledge Bases during inference time and require training updates to the pretrained LMs parameters.

On the other hand, controllable generation approaches like FUDGE (Yang and Klein, 2021), GeDi (Krause et al., 2021), and NADO (Meng et al., 2022) control generation without retraining the base language model. Specifically, Neurally-Decomposed Oracle (NADO) guides generation with a trained oracle to achieve high lexical control on CommonGen. However, these approaches don't address the commonsense aspect of sentences, sometimes resulting in implausible sentences.

For example, given the set of commonsense concepts { *food, customer, watch, employee, prepare* }, NADO achieves perfect lexical control and generates *"An employee is watching a customer prepare food."* This, however, is not commonsensical, since one would expect the employee to be preparing food for the customer.

To address this issue, we introduce a novel framework to guide the generation of pretrained LMs using a commonsense scorer in a controllable generation fashion.

In Table 1, we show a couple of motivating examples comparing NADO to generations produced by our framework. We are able to generate more commonsensical sentences than just guiding with lexical constraints (such as in Meng et al. (2022)). This increase in commonsense is reflected later in post-generation scoring using our commonsense scorer. We do note that post-generation sentences are not always scored higher due to variability in our parser. Still, even then the framework is able to guide generation toward sentences that are more commonsensical.

To accomplish this, we first propose an automatic commonsense metric grounded on ConceptNet (Speer et al., 2017) inspired by ACCENT (Ghazarian et al., 2023). In order to create our commonsense metric, we then introduce a parser to extract triplets of commonsense-related concepts. We additionally compute similarity scores using tails generated by COMET (Bosselut et al., 2019) to measure how well the sentence is grounded to ConceptNet.

Finally, we extend NADO by adding our commonsense metric masked with the lexical constraint into the signal used to train the auxiliary network.

We then demonstrate through a set of experiments that our framework is able to guide the generation of pretrained LMs towards more commonsensical generations while satisfying lexical constraints.

Model	Sentence	CS Score
NADO	A man is using a piece of burning wood to design a piece of furniture.	0.46
Ours	A man is using a *tool to burn wood to design* a piece of furniture.	0.84
NADO	An animal is being ridden by a man who is lassoing it and catching it.	0.71
Ours	man *rides a horse* and catches an *animal with a lasso.*	1.00
NADO	A man is holding a tree and sitting on it.	0.36
Ours	A man is sitting on a tree *holding an umbrella.*	0.44
NADO	A man is performing a spin throw with a baton.	0.48
Ours	A man *performs a spin* and *throws a baton* in the air.	0.68
NADO	An employee is watching a customer prepare food.	0.36
Ours	customers watch *employees prepare food*	0.32

Table 1.1: We compare sentences produced using lexically-guided generation with NADO (Meng et al., 2022) and sentences using commonsense-guided generation with our framework. Underlined are portions that are nonsensical while italicized are more commonsensical portions that we can generate. All sentences shown also match the lexical coverage of NADO.

1.1 Thesis Contributions

Our contributions are as follows:

1. We introduce a framework for guiding pretrained Language Models to more commonsensical generations without requiring retraining of the base Language Model (shown in Figure 1.1)

2. In order to guide our Language Model, we devise an automatic commonsense metric based on ConceptNet (Speer et al., 2017) which is inspired by ACCENT (Ghazarian et al., 2023). We find it has moderate correlation with human judgment of commonsense without reference sentences and outperforms or matches referenced-based metrics in a pilot study.

3. As part of our automatic commonsense metric, we propose a parser for ConceptNet triplet extraction that is trained on few-shot LLM-annotated relation data. We explore various settings for training our parser.

4. Finally, we propose a simple extension to Neurally-Decomposed Oracle (Meng et al., 2022) using our commonsense score, shifting from a lexically-constrained objective to a combined commonsense and lexical objective. We find that from initial experiments our oracle is able to guide generations toward higher commonsense scores while satisfying lexical constraints.

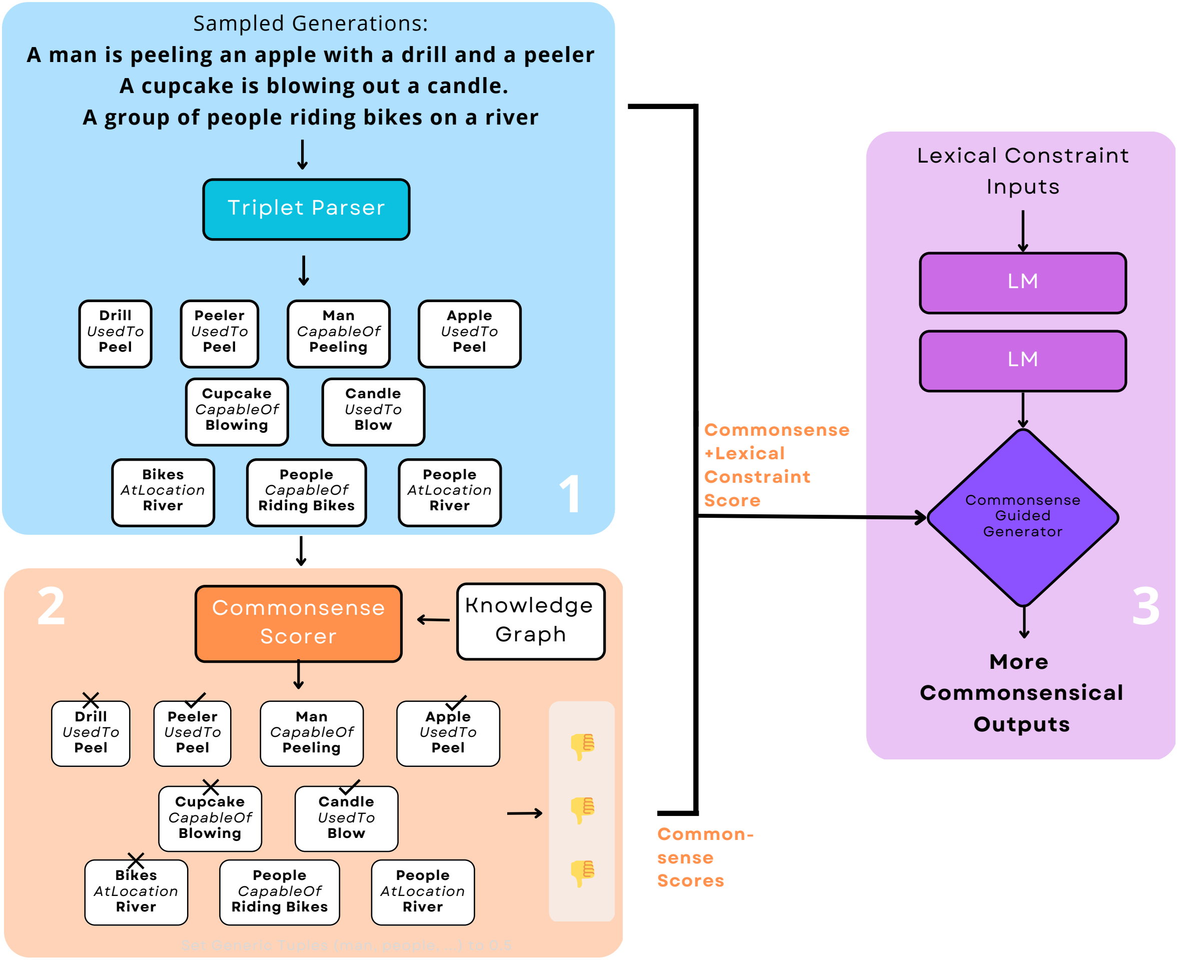

Sampled Generations:
A man is peeling an apple with a drill and a peeler
A cupcake is blowing out a candle.
A group of people riding bikes on a river
Triplet Parser
Drill UsedTo Peel
Peeler UsedTo Peel
Man CapableOf Peeling
Apple UsedTo Peel
Cupcake CapableOf Blowing
Candle UsedTo Blow
Bikes AtLocation River
People CapableOf Riding Bikes
People AtLocation River
1
2
Commonsense Scorer
Knowledge Graph
Drill UsedTo Peel
Peeler UsedTo Peel
Man CapableOf Peeling
Apple UsedTo Peel
Cupcake CapableOf Blowing
Candle UsedTo Blow
Bikes AtLocation River
People CapableOf Riding Bikes
People AtLocation River
Set Generic Tuples (man, people, ...) to 0.5
Common-sense Scores
Commonsense +Lexical Constraint Score
Lexical Constraint Inputs
LM
LM
Commonsense Guided Generator
3
More Commonsensical Outputs

Figure 1.1: Our pipeline consists of three main phases, Parsing, comprising a triplet parser; Compatability, comprising of a commonsense scorer ground on a knowledge graph; and Guidance, commonsense-guided generation with NADO. Triplets are parsed with our parser module and then aligned using our compatibility module to ConceptNet. We then use those scores along with lexical scores to guide the generation of pretrained LM using our commonsense-guided generation module.

1.2 Thesis Overview

The thesis is organized as follows:

Chapter 2 reviews related work in commonsense, knowledge-based and controllable generation approaches to CommonGen, and various automated and knowledge-based metrics.

Chapter 3 expands the core methodology of our framework, outlining our parser module, compatibility module, and guided-generation module.

Chapter 4 reports our experimental results that consist of parser experiments, a human correlation study, and guided-generation experiments.

Chapter 5 concludes with a discussion and areas of future work.

CHAPTER 2

Related Work

This chapter presents a review of relevant literature in Commonsense, Controllable Generation, and Automated Metrics in the context of Natural Language Processing. We begin by providing a discussion of some relevant research in commonsense, focusing on Knowledge Graphs and the CommonGen benchmark (Lin et al., 2020). We then discuss various knowledge-based and controllable generation systems proposed on CommonGen and provide a closer look into the NADO framework (Meng et al., 2022). We conclude by exploring various types of automated metrics.

2.1 Commonsense in NLP

Commonsense knowledge and reasoning has been a prominent topic in natural language processing (NLP) in recent years. Various avenues of research have been proposed to better endow machines with human-like understanding. Recent approaches have culminated in large graph structures of commonsense referred to as knowledge graphs which are generated by extracting or scraping commonsense data (Sap et al., 2020). These knowledge graphs can then later be directly integrated into models or used for retrieval.

Knowledge graphs have been a rich resource for the field of commonsense. ConceptNet (Speer et al., 2017) and ATOMIC (Hwang et al., 2020) are two popular knowledge graphs organized using triplets of the form head, relation, and tail. ConceptNet contains 3.4M triplets spanning 36 different relations such as AtLocation and PartOf. It consists of primarily high quality taxonomic, lexical, and physical commonsense knowledge. ATOMIC on the other hand focuses primarily on event-based commonsense relations, and comprises 880k triplets over 9 relations. It narrows its focus from ConceptNet, which contain a large portion of lexical and taxonomic relation data.

These knowledge graphs, while useful, are difficult to interact with. Searching a single triplet in knowledge graph often entails costly lookup. Additionally, these knowledge graphs often suffer from a lack of coverage; not all possible relations of commonsense could possibly be contained in the graph. Based on these issues, Bosselut et al. (2019) proposes the neural knowledge model COMET. COMET is trained to represent knowledge graphs and accepts a head or phrase subject, s, and a relation, r, and generates the corresponding tail or phrase object o. We explore COMET more in Section 3.2.1.

Many downstream commonsense benchmarks have been built on knowledge graphs for a variety of tasks such as question answering (Xiong et al., 2019) or dialogue (Zhou et al., 2018). We focus specifically on the task of generative commonsense reasoning using the CommonGen benchmark (Lin et al., 2020). It is organized as a dataset of concepts matching those in ConceptNet, and corresponding scraped and human-written evaluation sentences. We discuss CommonGen more in section 4.1.

2.2 Knowledge-Augmented Generation Systems

Various systems have been proposed to tackle the issue of generative commonsense reasoning on CommonGen. One common approach is to directly integrate Knowledge Bases into the neural network itself, we refer to these as Knowledge-Based Systems.

DKMRˆ2 (He et al., 2022) is one such Knowledge-Based System. It directly incorporates knowledge by retrieving from a external corpus of candidate sentences and re-ranks the retrieved sentences based on relevance. It then uses a metric distillation process to improve both the ranker and retriever.

KG-BART (Liu et al., 2021) is another example of a system of this type. KG-BART directly incorporates a Knowledge Graph, searching and adding relevant Knowledge Graph Data into the pretrained model as inputs.

These approaches can be highly effective, however they require accessing and searching an external corpus during inference, resulting in significant inference costs. Additionally, we wish for our commonsense addition to be compatible with future pretrained model architectures, without requiring the need for additional model design and costly retraining.

2.3 Controllable Generation with Guidance

Another approach to generative commonsense reasoning on CommonGen is using Controllable Generation to directly satisfy the lexical constraint of the task. These approaches don't incorporate Knowledge Graphs or address commonsense, instead focus more on controlling the existing pretrained model to satisfy a defined oracle-level constraint. Some advantages to these approaches is that they don't require access to an external corpus during inference, and are compatible with existing model architectures without need for retraining.

GeDi (Krause et al., 2021), FUDGE (Yang and Klein, 2021), and NADO (Meng et al., 2022) are some examples of Controllable Generation. GeDi controls generation by using a class-conditioned language model as a discriminator to guide the generation of the pretrained model. FUDGE on the other hand learns an attribute predictor on partial sequences, thereby allowing it to better guide generation during inference time. However, both these methods require an external dataset of positive and negative examples.

NADO solves this issue by introducing sampling from the base pretrained model and provides a mathematically rigorous framework for guided-generation with a trained decomposed oracle. NADO also directly applies this framework to the lexical constraint portion of the CommonGen benchmark. Due to the lack of an external dataset, we choose to use NADO for the Guided-Generation module of our framework.

2.3.1 Neurally-Decomposed Oracle for Guided-Generation

NADO guides generation by training an auxiliary model on sampled generations from the base pretrained model. It uses the auxiliary model to decompose a sentence-level oracle signal, which it later uses to guide the generation of the pretrained model to satisfy the conditions of the oracle.

It accomplishes this by training the auxiliary model to predict the success-rate prediction function R_p^C which is used to provide Guided-Generation through a token-level distribution q on inference time. During training, we train with a cross entropy objective between R_p^C which is derived from our oracle signal and our predicted function R_θ^C from our auxiliary model.

$$E_{y \sim p(y|x)} L_{CE}(x, y, R_\theta^C) = \sum_{i=0}^{T} \mathrm{CE}(R_p^C(\mathbf{x}, \mathbf{y}_{\leq i}), R_\theta^C(\mathbf{x}, \mathbf{y}_{\leq i}))$$

NADO also introduces a tunable regularization term λ which is included in our loss. We further describe the setting of NADO in Section 3.3.1.

2.4 Automated Metrics

Popular metrics in NLP, such as BLEU (Papineni et al., 2002) and ROUGE (Ganesan, 2018), are based around n-gram matching between the candidate and reference sentences. The sentence score is assigned corresponding to overlap between a single candidate and one or more reference sentences. For example, BLEU is calculated using the precision of n-gram overlap between the candidate and the corresponding reference sentences, with parameters for brevity and size of the N-gram.

These metrics provide a fast approximation of sentence quality. However, recently there has been a push toward more semantically meaningful scores for evaluation. Metrics such as SPICE (Niu et al., 2022) extend this framework over scene graphs to provide a more meaningful representation. Various model-based metrics have also been proposed to incorporate model understanding of candidate and reference sentences.

2.4.1 Model-Based Metrics

Model-Based Metrics are another type of metric that use the embedding similarity of the candidate and reference sentences. For example, BERTScore (Zhang et al., 2019) is a model-based metric that has been proposed to evaluate sentence similarity by comparing the contextual BERT (Devlin et al., 2019) embeddings of tokens.

By creating contextual embeddings of both the references and candidate sentences it can compare them using pairwise cosine similarity to produce the corresponding BERTScore after some adjustments in weighting and rescaling.

One significant limitation is that BERTScore isn't well defined in the case where there are no reference sentences. This prevents us from using the score unless we have reference sentences, and even then, we would only be able to evaluate how closely related it was to the reference sentences. It can, however, provide some approximation of commonsense since the human-written references sentences are typically commonsensical.

2.4.2 Knowledge-Based Metrics

One final type of metric that we review are Knowledge-Based Metrics. These are metrics based on commonsense, specifically those grounding on knowledge graphs. For example, Zhou et al. (2022) measures the commonsense of dialogue turns by hard and soft matching relations across each turn to ConceptNet. ACCENT (Ghazarian et al., 2023) focuses on measuring the event commonsense and leverages the ATOMIC knowledge graph (Bosselut et al., 2019) to ground the dialogue. Specifically, ACCENT trains a parser based on a human-written dataset of dialogue turns and runs a compatibility test using COMET-ATOMIC to measure the commonsense of the dialogue.

Our metric is inspired by ACCENT but is primarily focused on sentences over dialogue. Additionally we differ by grounding using ConceptNet grounding and training our parser on a LLM-annotated dataset.

CHAPTER 3

Methods

We present Commonsense Guidance as a framework for guiding the generations of pretrained Language Models (LMs) to more *commonsensical* generations. In our framework, we define commonsense using alignment with ConceptNet (Speer et al., 2017). ConceptNet is a Knowledge Graph connecting phrases with labeled edges in triplet form and is designed to represent general knowledge and relationships between phrases. We construct our notion of how *commonsensical* our generation is by considering the alignment of each parsed triplet to existing triplets in ConceptNet. For our work, we limit ourselves to examining one-hop relationships in the generation. Figure 3.1 provides a overview of our commonsense scorer.

Our framework consists of three main stages:

1. **Parsing: ConceptNet Triplet Extraction**. We first parse relevant triplets from the sentence for each relevant relation. This is done using a pretrained T5 (Raffel et al., 2020) finetuned on sentences parsed by a few-shot Large Language Model.

2. **Compatibility: ConceptNet Alignment of Parsed Triplets**. We then evaluate the alignment of each parsed triplet with ConceptNet. We notably leverage COMET (Bosselut et al., 2019) to access ConceptNet following ACCENT. We use our alignment scores to construct a final commonsense score (CS Score).

3. **Guidance: Guided-Generation Using Commonsense Score**. We finally guide the generation of a pretrained LM controlled by a modified NADO (Meng et al., 2022) using our final score along with the lexical coverage score described in NADO.

During inference after training with our framework, NADO is able to guide the generation of the pretrained Language Model without gradient updates, parsing, or access to the Knowledge Graph. In the following sections we will discuss each stage of our framework.

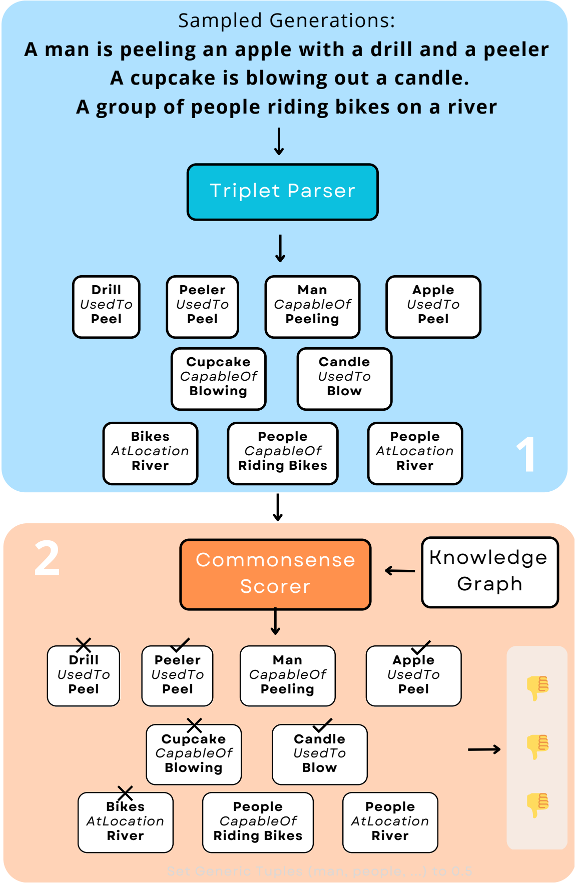


Figure 3.1: The commonsense scoring portion of our framework comprising the Parsing and Compatibility module is shown in this Diagram. Triplets are parsed with our parser module and then aligned using our compatibility module to ConceptNet. We then use it in the commonsense guided-generation explained in section 2.3.1.

3.1 Parser: ConceptNet Triplet Extraction

Our parser module serves to extract all relevant triplets of the form (head, relation, tail) from each generation for all interested relations. We focus on a subset of relations that exists in ConceptNet. Namely, we aim to parse any triplets in the generation containing relations of the type *UsedFor*, *CapableOf*, *AtLocation*, and *PartOf*.

We aim to extract triplets containing both reasonable and unreasonable relationships to inform our commonsense score. To achieve this, we align our parsing approach with the definition of relations in ConceptNet. It's worth noting that the relation *UsedFor* is defined in ConceptNet as both a connection between a concept and its intended use and an activity carried out on a concept. For instance, (Piano, *UsedFor*, Music) and (Piano, *UsedFor*, Playing) are both examples of *UsedFor* relationships. Figure 3.1 shows an overview of our triplet parser.

3.1.1 Few-Shot LLM Annotation to Train Parser

Large Language Models have demonstrated an emergent ability of few-shot in-context learning (Brown et al., 2020), allowing such LLMs to perform tasks given only demonstrations with relevant context without gradient-based parameter updates. LLMs have also demostrated promising performance for semantic parsing (Dong and Lapata, 2016; Dunn et al., 2022).

To refine our parser to accurately extract the relevant triplets, we leverage a few-shot Large Language Model to annotate our sentences with parsed triplets for each relation. Our sentences are human-written and sourced from CommonGen (Lin et al., 2020). A detailed discussion of CommonGen is presented in Section 4.1.

The few-shot LLM we use is OpenAI's Da-Vinci-003 GPT3.5 Model. Davinci is a GPT-3-like model (Brown et al., 2020), finetuned on code and text with a mixture of supervised learning and reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022a).

For our few-shot prompt, we hand-curate five example sentences as shown in Appendix A.1. We then attach our few-shot prompt as the prefix to our sentence to generate our parsing result.

3.1.2 Finetuned T5 Parser in LLM-Annotated Data

Using GPT-3 Davinci to parse all our generations is computationally expensive. To address the issue of limited computational resources, we propose the implementation of a finetuned T5 model (Raffel et al., 2020) for the purpose of parsing generations. The T5 model has demonstrated strong learning capabilities in low-resource settings, and has been trained on our few-shot LLM-Annotated sentences dataset.

Fig. 3.2 depicts the relationship between our Few-Shot LLM and Finetuned T5 Parser. In our training approach, we parse all triplets of one relation individually per example. Each sentence thereby yields one example per relevant relation. We choose to use one unified T5 for all the relations. Our choice for T5 Parser design is informed by ACCENT (Ghazarian et al., 2023), and Appendix A.2 provides the prompt for each relation that is attached as a prefix during training and inference.

On inference time, we make multiple forward passes for each desired relation we want to parse. We then gather all parsed triplets to generate our parsing result.

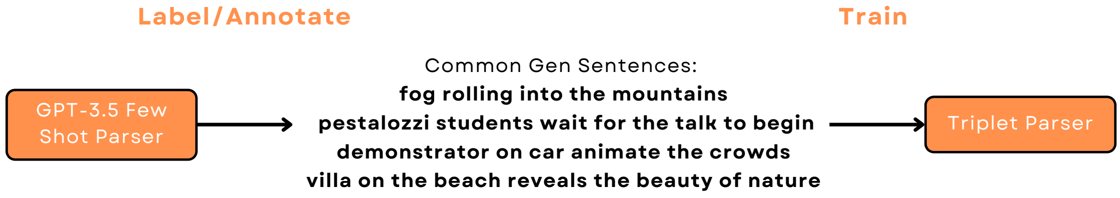

Label/Annotate
Train
GPT-3.5 Few Shot Parser
Common Gen Sentences:
fog rolling into the mountains
pestalozzi students wait for the talk to begin
demonstrator on car animate the crowds
villa on the beach reveals the beauty of nature
Triplet Parser

Figure 3.2: We use a GPT-3.5 Few-Shot Parser to annotate a dataset of CommonGen (Lin et al., 2020) Sentences. We then finetune a T5 model on our annotated sentences to create our triplet parser. The GPT-3.5 parsing is performed in one pass, while the Finetuned T5 parsing is performed in one pass per relation and later collated together.

3.2 Compatibility: ConceptNet Alignment of Parsed Triplets

Our Compatibility module constructs our Commonsense Score (CS Score) by evaluating how aligned each triplet is with ConceptNet. We adapt the Compatability module described in ACCENT (Ghazarian et al., 2023) to ConceptNet relations to score our concept-based triplets. Our goal is to check if our parsed triplets are sensible and aligned with ConceptNet.

Due to the size of ConceptNet, the Compatibility module uses COMET (Bosselut et al., 2019) to access triplets in ConceptNet. Figure 3.1 shows an overview of the commonsense scoring portion of our framework that uses our compatibility module.

3.2.1 Accessing ConceptNet with COMET

COMET is a generative approach to representing a given commonsense knowledge base (CSKB). Given a phrase subject, s, and a relation, r, it is trained to generate the phrase object, o, of the triplet $\{s, r, o\}$. In this manner, it is able to both efficiently return triplets stored in the CSKB as well as generalize to novel commonsense knowledge.

In our framework, we use COMET to query phrase objects given a phrase subject and relation, represented as $\{s_t, r_t\}$. Using k-beam search generation, we query the generalized top-k COMET phrase objects as o_g. We then compare them to our existing phrase object o_t to evaluate if a given triplet t is sensible.

3.2.2 Alignment of Parsed Triplets to ConceptNet

In order to evaluate the similarity between the phrase object o_t and top-k generalized COMET outputs o_g, we leverage SentenceBERT (Reimers and Gurevych, 2019) to embed our phrases. Following ACCENT (Ghazarian et al., 2023), we then calculate the similarity score by taking the closest phrase object o_g with our existing phrase object o_t.

$$\text{COMPAT}((h,r,o)|C) = \max_{1\leq i\leq k} \cos(\text{embed}(o_t), \text{embed}(o_g^i))$$

One problem we come across is that there are some phrase subjects such as *people*, *woman*, *man* where the set of phrase objects is prohibitively large. For these phrase subjects we simply ignore them and set their score to be 0.5.

We perform this compatibility test for all p parsed triplets for all relevant relations. We then take the minimum score for all triplets as our compatibility score. Our rationale is that given a sentence, a single incorrect triplet can result in a nonsensical sentence. We define the commonsense score as follows:

$$\text{CS Score} = \min \sum_{0\leq i\leq p} \text{COMPAT}((h_i, r_i, o_i)|C)$$

The commonsense score reflects the level of how well a generation aligns with commonsense knowledge in ConceptNet. The CS ranges between 0 and 1, where 1 indicates that all parsed triplets in the generation align perfectly with commonsense knowledge in ConceptNet. A CS of 0 indicates that one of the parsed triplets is completely misaligned with the commonsense knowledge in ConceptNet.

In the following section, we describe how we use our derived commonsense score to guide the generations of our Language Model toward more *commonsensical* generations.

3.3 Guidance: Guided-Generation Using Commonsense Score

In order to guide generations of our pretrained language model with our commonsense score, we train a Neurally Decomposed Oracle (NADO) introduced by Meng et al. (2022). We follow the sampling scheme of the original NADO and slightly finetune our pretrained GPT-2 base to produce the sampled generations. We then diverge from the original and train with continuous rather than binary labels as well as introduce our commonsense score into our oracle guided-generation.

3.3.1 NADO Method Overview

NADO decomposes a sequence-level boolean oracle to token-level guided-generation. It does this by training an auxiliary neural model on sampled generations from the base model scored by oracle C. The auxiliary neural model approximates the token-level guided-generation function R_θ^C.

Then, during inference, the auxiliary neural model, R_θ^C, guides the generations of the base model to satisfying oracle constraints. In practice, NADO also introduces a KL divergence hyper-parameter, λ, to regularize the effect of the guided-generation model. We also find this parameter necessary to prevent overfitting the oracle signal. Further details are discussed in Section 2.3.1.

Our auxiliary neural model, R_θ^C, is trained on sampled generations from the base model. It uses a temperature parameter to sample from a distribution of generations. In order to sample plausible generations, we also use importance sampling and fine-tune the base model on sentences from the dataset.

NADO demonstrates successful control given the lexical constraint task on CommonGen. The oracle formulation simply checks if all input keywords are in the generation and returns a boolean value used as the oracle label.

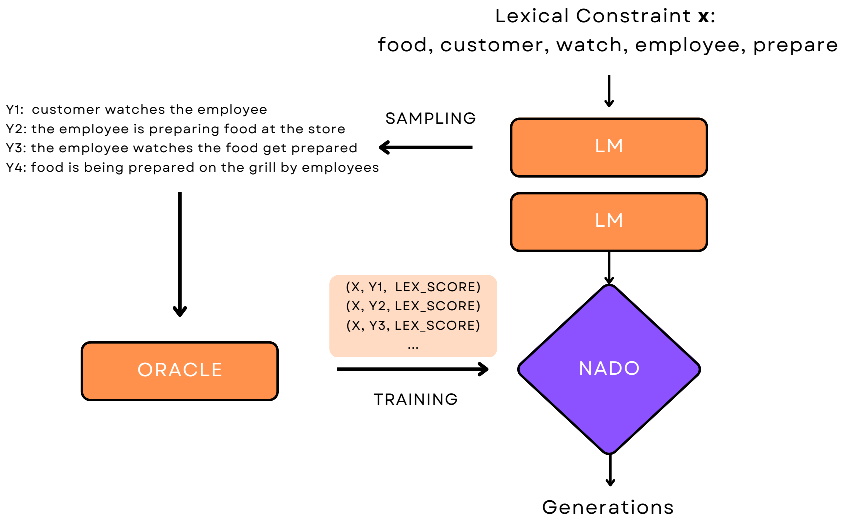

Lexical Constraint **x**:
food, customer, watch, employee, prepare
Y1: customer watches the employee
Y2: the employee is preparing food at the store
Y3: the employee watches the food get prepared
Y4: food is being prepared on the grill by employees
SAMPLING
LM
LM
(X, Y1, LEX_SCORE)
(X, Y2, LEX_SCORE)
(X, Y3, LEX_SCORE)
...
ORACLE
TRAINING
NADO
Generations

Figure 3.3: This diagram describes the application of NADO on the task of lexical control. The oracle function checks if all keywords x are in the generation y. Using samples from the Base Model, we then train R_θ to provide token-level guided-generation pushing the generated text to satisfy the oracle function. Diagram inspired from Meng et al. (2022).

3.3.2 Commonsense Soft Constraint with NADO

We modify NADO by introducing our Commonsense Score into the oracle function. As outlined in Section 3.3.1, NADO uses a boolean oracle function to check if all keywords are satisfied. Our oracle function uses the same lexical boolean and multiples it with our Commonsense Score in order towards push it to generations that are both lexically-controlled as well as more commonsensical. Inference time we then use our trained NADO to produce the desired generated text. Figure 3.3 provides a summary of this process.

One consideration is that our commonsense score is not able to generalize to completely unseen concepts since our oracle needs some notion of commonsense relations for the concepts. One future approach that can be used to alleviate this issue is by sampling using concepts of interest.

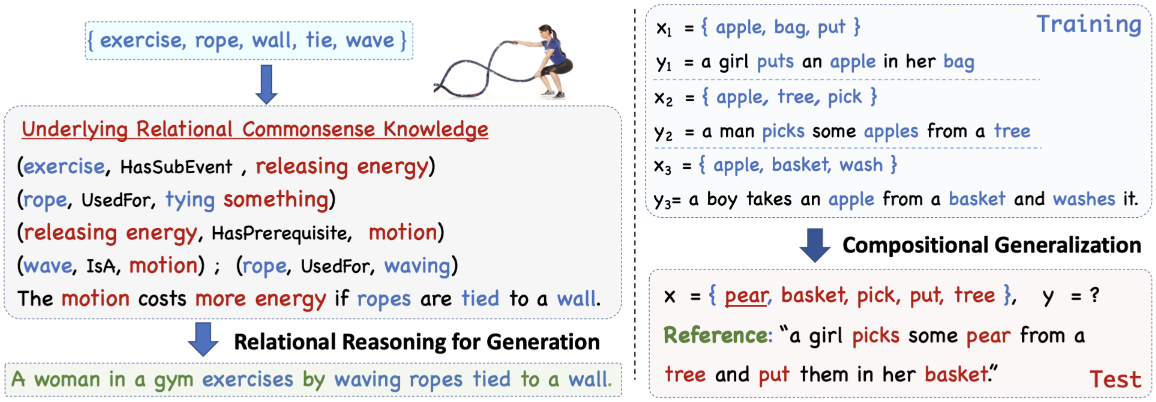
CHAPTER 4

Experiments and Results

This section presents an evaluation of the efficacy of various stages of our framework. The evaluation is conducted through a series of experiments that aim to assess the impact of varying settings for the parser module, the validity of our commonsense score, and the overall performance of commonsense guided-generation with the full framework.

To evaluate the parser module, we conduct experiments using different training dataset sizes and model settings. We also perform a pilot human correlation study to compare our commonsense score against reference-based methods. To assess the effectiveness of the commonsense-guided generation module, we conduct experiments exploring the impact of guided-generation with the commonsense score and the commonsense and lexical combined score.

We leverage the CommonGen dataset (Lin et al., 2020) for both the parser training set and guided-generation. All experiments are run on a single NVIDIA RTX A6000.



{ exercise, rope, wall, tie, wave }
Underlying Relational Commonsense Knowledge
(exercise, HasSubEvent , releasing energy)
(rope, UsedFor, tying something)
(releasing energy, HasPrerequisite, motion)
(wave, IsA, motion) ; (rope, UsedFor, waving)
The motion costs more energy if ropes are tied to a wall.
Relational Reasoning for Generation
A woman in a gym exercises by waving ropes tied to a wall.
Training
x1 = { apple, bag, put }
y1 = a girl puts an apple in her bag
x2 = { apple, tree, pick }
y2 = a man picks some apples from a tree
x3 = { apple, basket, wash }
y3= a boy takes an apple from a basket and washes it.
Compositional Generalization
x = { pear, basket, pick, put, tree }, y = ?
Reference: "a girl picks some pear from a tree and put them in her basket."
Test

Figure 4.1: This diagram is from Lin et al. (2020). CommonGen is composed of concept sets x_i and corresponding reference sentences y_i. It aims to measure relation reasoning and compositional generalization. One example shown above is the concept set of { exercise, rope, wall tie, wave } which align with the shown relations. A provided reference sentence for the concept-set is *a woman in a gym exercises by waving ropes tied to a wall.*

4.1 Dataset Overview

The CommonGen task is a generative commonsense reasoning task designed to evaluate commonsense relation reasoning and compositional generalization (Lin et al., 2020). The associated dataset is collected by sampling sentences containing 3-5 concepts (of which are present in ConceptNet) from several caption-related datasets. The development and testing set is then created using Amazon Turk by annotating a collection of held-out concept sets resulting in a set of human-written sentences. Figure 4.1 provides an illustration of the dataset.

The training set contains 32,651 unique concept sets while the development set contains 993 concepts. For reference sentences, The training set has 67,389 sentences and the development set has 4,018 sentences. We use the training set sentences to train our parser, evaluate the correlation of referenced-based metrics, and fine-tune our base model for NADO.

4.2 Parser Experiments

The triplet parser plays an integral role in the framework to accurately generate our commonsense score. We aim to create a parser that can accurately understand the composition of the sentence and parse out the relevant relations. We want these relations to accurately reflect the composition of the sentence, regardless of whether they are sensible or not. Additionally, we want to cover all existing triplets in the sentences.

4.2.1 Parser Training Setup

As described in 3.1, in order to train our parser we leverage Davinci, a few-shot LLM, to annotate our training data with triplet parsings. We access Davinci through the OpenAI API and generate using our few-shot prompt with greedy decoding and a temperature of 0.7. We use Davinci to few-shot parse a training set of 1097 training sentences. We additionally few-shot parse a validation set of 250 sentences, on which we then perform a second human validation pass to fix any mistakes or incorrect parsings.

We note that there is a slight domain shift between our sentences and later sampled generations. Typically, sentences from CommonGen are more commonsensical and coherent than sampled generations. In Section 4.3 we find the score using our parser can still able to achieve moderate human correlation despite the mismatch.

We also choose to use a pretrained T5 (Raffel et al., 2020) for our model following ACCENT (Ghazarian et al., 2023). We note that any sequence-to-sequence model could work in our framework.

Figure 4.2 show the distribution of the number of triplets per relation in our eventual train and validation sets. Most relations have 1-2 triplets per sentence, while *PartOf* is less present in most sentences.

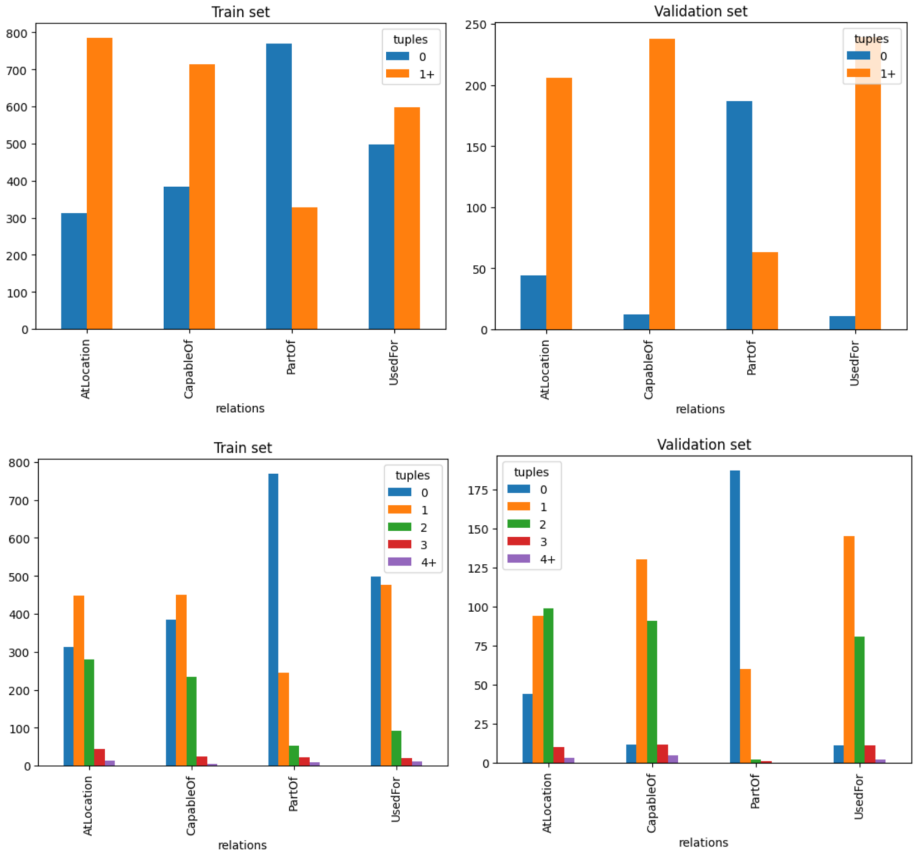


Figure 4.2: Our parser dataset consists of a training split of 1097 generations and a validation split of 250 generations. In our dataset, we use the following relations: { AtLocation, CapableOf, PartOf, UsedFor }. PartOf is biased toward not being contained in the generation, and we find a slight shift in the positive and negative ratios between the splits.

4.2.2 Comparison of Different Parser Architectures

We explore the space of parser architectures and try three different settings. We first explore using the instruction tuned T5-Flan (Chung et al., 2022) for our base model. We then compare against using Davinci Few-Shot. Finally, we do a quick ablation using only the CommonGen concept set as our parsed triplets. We call this ”T5-Large w/ CG Naive.”

4.2.3 Parser Results

For all experiments, we report the precision, recall, and f1-score of the set of predicted triplets and ground-truth validation triplets averaged for each relation.

We explore balancing the relations by removing positive or negative (empty set) sentences until the desired ratio is reached for each relation. This did not improve our performance.

We experiment with training our parser on various dataset sizes and balances of positive and negative (empty set) triplets for each relation. Specifically, we train on 200, 400, 600, 800, 1000 sentences, randomly sampling from our original training set to explore the effect of our low-resource setting. As shown in Table 4.1, we find that our model improves with scale but already starts to plateau. Hence, we choose to train with our full training set of 1097 sentences.

Using an instructed-tuned T5 doesn't seem to improve our results for our setting. We also find our trained T5 outperforms using Few-Shot Davinci on our validation set. We also find just using the concept-set of CommonGen as the parsed triplets performs poorly due to low recall of existing triplets in the sentence and lack of coverage for nonsensical relations.

Model	**Precision**	**Recall**	**F1-score**
T5-Large w/ LLM labeled			
200 examples	70.14	64.04	66.34
400 examples	73.32	65.84	68.90
600 examples	78.24	65.62	70.79
800 examples	77.76	65.54	70.50
1000 examples	79.92	66.75	72.01
T5-Large w/ LLM labeled			
50/50 positive/negative	78.36	62.50	68.82
Drop 25% negatives	76.65	66.25	70.46
Drop 50% negatives	77.35	65.35	70.23
T5-Large w/ CG Naive	42.16	37.58	38.85
T5-Large w/ LLM labeled	79.60	67.90	**72.56**
T5-Flan w/ LLM labeled	78.22	68.20	72.24
Da-Vinci Few-Shot Parser	74.68	65.18	69.17

Table 4.1: Table showing precision, recall, and F1-score for various parser models. In the first section, we display the results of our experiments scaling the dataset size. The second portion shows initial testing in dropping the posiitve and negative ratio. The last section explores ablations in model settings, specifically testing instruction-tuned and naive common-gen based models.

4.3 Pilot Human Correlation Studies

To confirm the accuracy of our score, we conducted a pilot study on a sample of NADO sentences. We compare our score against BLEU4 and SPICE, popular N-gram reference-based metrics. For our references, we use the corresponding sentences from CommonGen. To further validate our score, we also compare against BERTScore (Zhang et al., 2019), which serves a strong baseline against embedding-based metrics.

4.3.1 Pilot Study Settings

The settings of our pilot study is as follows: we recruit two undergraduate students to rate 60 sampled generations on a scale from 1 to 3, where 1 is not commonsensical, and 3 is commonsensical. We instruct them to disregard coherency and instead focus on the commonsense of the relations between concepts in the sentence. The full instruction prompt is found in Appendix B.1.

For our sampled generation, we randomly sample from the top and bottom quartile of our generations, resulting in a non-normal distribution. To address this issue, we run Spearman's Rank correlation, but also include Pearson's correlation for interest.

Metric	Spearman's Rank	Pearson's
FT T5 - CS Score	0.4508	0.4203
GPT3.5 - CS Score	**0.5727**	0.4531
Referenced-Based:		
BERTScore	0.3796	0.4597
BERTScore - ALL	0.4899	0.5125
BLEU4	0.2509	0.2615
SPICE	-0.1594	-0.0761

Table 4.2: We run a correlation study comparing our commonsense metric against various reference-based metrics. We primarily evaluate the correlation using Spearman's Rank correlation, due to the non-normal distribution of our data. We evaluate BERTScore with one reference to mimic our low resource setting, and set BERTScore-ALL for using all the provided references. Our Inter-annotator Spearman's Rank is 0.4841.

4.3.2 Pilot Study Results

We now can compare the correlation of our commonsense score and automated reference-based measurements against human commonsense scores. We see that the both our trained T5 parser and few-shot LLM have substantially higher correlation compared to N-gram based metrics. Our trained T5 parser lags behind our few-shot parser, which we attribute to the domain shift of our training versus sampling and correlation studies.

We compare against both BERTScore with one reference and BERTScore with full references (1-3 sentences) and find our LLM-based parser outperforms both. Our finetuned T5 also beats BERTScore in the low-resource setting while matching or slightly lagging behind BERTScore in the full resource setting.

One explanation of the strong performance of BERTScore is that it essentially approximates some measure of commonsense, since it measures the distance from the commonsensical reference sentences. It is also able to capture underlying coherence patterns that can bias annotators, such as sentence fragments, thereby boosting its score. There are also instances of commonsensical sentences that BERTScore misses due to missing coverage of the reference sentences.

We conclude that our T5 trained parser has higher correlation with humans on a commonsense rating task than popular CommonGen N-gram measurements and matches the strong baseline of a reference-based BERTScore while requiring no human-written reference sentences.

4.4 Guided-Generation Experiments

In this section, we conduct a series of experiments using our framework to guide the generation of a pretrained GPT2-Large (Radford et al., 2019) on the CommonGen dataset. We investigate guided-generation with just our commonsense score and our commonsense score with the lexical boolean. We also compare the effect of guided-generation with different scores, reporting common automated metrics as well as lexical coverage and our commonsense score.

4.4.1 Guided-Generation Experiment Setup

Our experimental setup follows the original settings in NADO (Meng et al., 2022). We first finetune the pretrained GPT2-Large on the CommonGen training split for 3 epochs in a Sequence-to-Sequence fashion, where our input is our concept-set and output generation is the reference sentence.

We then sample 1.5M sentences from the finetuned base with a temperature of 0.8 conditioned on the 32,651 unique concept-sets of the CommonGen training split. Using our framework, we then score our sampled generations. With our scored sampled generations, we then train NADO for 19 epochs with a learning rate of $1e - 5$. Our validation split is the CommonGen validation split and all referenced-based metrics are run against provided CommonGen validation sentences.

Epoch	CS-Score	BLEU3	BLEU4	ROUGE	CIDEr	SPICE	Coverage %
Base	0.4792	39.1	29.2	53.1	15.81	30.7	90.74
1	0.5290	38.7	28.8	52.9	15.72	30.4	90.74
3	0.5241	38.7	28.8	52.6	15.59	30.4	90.66
5	0.5253	39.0	29.2	52.9	15.70	30.7	90.96
7	0.5321	38.8	29.0	52.8	15.65	30.6	90.72
9	0.5321	38.8	29.0	52.7	15.59	30.5	90.53
11	0.5297	38.5	28.7	52.4	15.48	30.4	90.53
13	0.5329	38.7	28.9	52.6	15.60	30.5	90.74
15	0.5358	38.6	28.8	52.6	15.55	30.6	90.50
17	0.5349	38.7	29.0	52.5	15.61	30.5	90.77
19	**0.5359**	38.4	28.7	52.5	15.52	30.6	90.64

Table 4.3: This table presents our results for Guided-Generation with Commonsense Score. We specifically train NADO solely on the commonsense score without any lexical constraints. We evaluate on standard metrics as well as our commonsense metric. NADO clearly seems to be able to guide the generation of the model toward a higher commonsense score, but since it does not train to improve lexical constraints, the coverage does not change. Additionally, the automated metrics, which are shown to have a low correlation with human judgments of commonsense, also do not show a noticeable change.

4.4.2 Investigation of Guided-Generation with Commonsense Score

We train NADO on only the commonsense score without the lexical boolean. Specifically, we take the minimum compatibility score of all extracted relation triplets from the sample generation and use that as our soft label for the sentence.

As shown in Table 4.4.1, we find that we are able to successfully increases our Commonsense Score in this setting, but it doesn't significantly move any of our referenced-based metrics or coverage. An explanation for this is that these metrics are measuring the similarity between our output sentence and references, which as shown in our pilot study, have little correlation with human judgments of commonsense. Our aim is to guide the generation of our sentences to be both commonsensical as well as satisfying the lexical constraint objective for the CommonGen Task, and in the next experiment, we explore guided-generation with both our lexical and commonsense score.

Epoch	CS-Score	BLEU3	BLEU4	ROUGE	CIDEr	SPICE	Coverage %
Base	0.4792	39.1	29.2	53.1	15.81	30.7	90.74
1	0.5282	38.7	28.8	52.9	15.77	30.7	92.08
3	0.5311	38.1	28.3	53.2	15.80	30.9	92.43
5	0.5318	39.2	29.3	53.3	16.04	30.9	93.24
7	0.5311	38.1	28.4	54.4	16.03	31.2	93.72
9	0.5303	39.4	29.5	53.4	16.10	31.1	94.74
11	0.5325	39.6	29.6	53.5	16.16	31.2	94.98
13	0.5364	39.6	29.5	53.7	16.08	31.2	95.44
15	**0.5655**	39.7	29.8	53.5	16.05	31.2	95.52
17	0.5622	39.7	29.6	53.6	16.08	31.2	95.46
19	0.5645	40.0	30.1	53.9	16.32	31.3	95.44

Table 4.4: This table presents our results for Guided-Generation with a combined Commonsense and Lexical Score. We evaluate on standard metrics as well as our commonsense metric. NADO is able to successfully guide generations to both higher coverage as well as higher commonsense score. Due to our lexical constraint, the automated metrics also seem to noticable improve. More generations can be found in Appendix **??**.

4.4.3 Investigation of Guided-Generation with Lexical and Commonsense Score

We conduct another experiment where we train NADO using both our lexical and commonsense score. As described in Section 3.3.1, our score is commonsense score masked by the lexical coverage boolean in order to encourage generations that are both commonsensical as well as lexically constrained.

Table 4.4.2 shows our results. We are able to successfully improve our Commonsense Score while also improving lexical coverage and reference-based automated metrics. Figure 4.3 shows a sample of selected generations compared to the base model and our best model trained only on lexical constraints. We find that empirically it is able to produce more commonsensical sentences compared to the base model.

A limitation to our evaluation is that the only systematic measurement of commonsense for our generated sentences is the Commonsense Score we are training on and our selected samples. We address possible future systematic evaluation methods in Section 5.2.

Concepts		: design piece burn wood tool
Base	(0.84):	A man is using a tool to burn a piece of wood.
LEX	(0.46):	A man is using a piece of burning wood to design a piece of furniture.
CS & LEX	(0.84):	A man is using *a tool to burn wood to design* a piece of furniture.
Concepts		: walk dog ride
Base	(0.34):	A man is walking a dog and riding it.
LEX	(0.47):	A dog is walking on a leash while someone rides on it.
CS & LEX	(0.31):	A man is *walking his dog* while *riding a scooter*.
Concepts		: hold sit tree
Base	(0.00):	A group of people sitting under a tree.
LEX	(0.36):	A man is holding a tree and sitting on it.
CS & LEX	(0.44):	A man is sitting on a tree *holding an umbrella*.
Concepts		: air baton perform spin throw
Base	(0.48):	A man is performing a spin throw with his baton.
LEX	(0.48):	A man is performing a spin throw with a baton.
CS & LEX	(0.64):	A man *performs a spin* and *throws a baton* in the air.
Concepts		: food customer watch employee prepare
Base	(0.36):	An employee is watching a customer prepare food.
LEX	(0.36):	An employee is watching a customer prepare food.
CS & LEX	(0.32):	customers watch *employees prepare food*.
Concepts		: horse ride animal lasso catch
Base	(1.00):	A man is riding a horse and catching a wild animal with a lasso.
LEX	(0.71):	An animal is being ridden by a man who is lassoing it and catching it.
CS & LEX	(1.00):	man rides a horse and *catches an animal with a lasso*.

Figure 4.3: Here we present a selected sample of generations with implausible portions underlined and more *commonsensical portions italicized*. We also provide the predicted commonsense score in parentheses. LEX is only lexical following the setting in (Meng et al., 2022), CS + LEX is using both commonsensical and lexical signal, and BASE is the pretrained model finetuned for 3 epochs. We find that the commonsense addition helps generations make more sense while preserving lexical constraints compared to just using NADO. We notice that sometimes the model is able produce a more commonsensical sentence even if the predicted score after generation is lower. This is likely due to variability and noise of the parser and is a area of future improvement.

4.4.4 Investigation of Effect of Lexical and Commonsense Scoring

We perform an ablation to investigate the effect of each component of our score. Using our finetuned base, we run three experiments, choosing the highest Commonsense Score for each run. As shown in Table 4.4.4, We find that the lexical-only score does result in high CS-Score, but together they achieve a higher CS-Score than any component alone. We also find that using our combined lexical and commonsense score preserves our automated metrics with a slight degradation of Coverage compared to our lexical-only score.

	CS-Score	BLEU3	BLEU4	ROUGE	CIDEr	SPICE	Coverage
Base (3 epochs)	47.92	39.1	29.2	53.1	15.81	30.7	90.7
+ Lexical	55.61	39.0	28.9	52.8	15.86	31.2	**96.5**
+ Commonsense	53.58	38.6	28.8	52.6	15.55	30.6	90.5
+ Commonsense & Lexical	**56.55**	39.7	29.8	53.5	16.05	31.2	95.5

Table 4.5: Here we provide an ablation of Lexical and Commonsense Signal in Guided-Generation. We choose top performing epoch and use a base that is pretrained for 3 epochs. We evaluate on our standard automated and commonsense scores. All scores are able to provide noticeable changes in commonsense scoring to the base, we hypothesize that this is due to the lexical constraints being commonsensical in nature, causing higher coverage to result in more commonsensical sentences. However, we still find guided-generation with the combined commonsense and lexical scores to result in a higher commonsense score.

CHAPTER 5

Conclusion

This thesis has presented a framework for guiding text generation models toward more commonsensical generations using a commonsense score grounded on a Knowledge Graph. We first introduced our triplet parser that captures the relevant triplets from a sentence based on the desired relations. Through a series of experiments, we showed that our parser performs well in our low-resource settings and is able to generate accurate triplets for guiding text generation.

We then introduced our commonsense scoring method, which leverages the Knowledge Graph to score the commonsense of the triplets generated by our parser from the sampled generations. Our evaluation showed that the score produced by our method is correlated with human judgments of commonsense and outperforms many automated referenced-based metrics, validating the effectiveness of our approach.

Finally, we integrated our commonsense score into a text generation pipeline and evaluated the impact of the commonsense guided-generation on generated texts. Our experiments demonstrated that incorporating commonsense guided-generation results in generated texts that are more commonsensical compared to those generated with just lexical guided-generation.

5.1 Discussion

Overall, our framework provides a promising approach for improving the quality of text generation by incorporating commonsense knowledge through guided-generation grounded on a Knowledge Graph. However, there are still limitations to our approach that should be addressed in future work.

One limitation is the quality of the underlying Knowledge Graph, which can impact the accuracy of the commonsense scores generated. Improvements to the Knowledge Graph, such as expanding its coverage and increasing the granularity of its relationships, could enhance the accuracy of our framework.

Another limitation is the accuracy of our parser trained on annotated data from a few-shot LLM. Using existing symbolic parsing methods, providing additional syntactical data, or training on a broader distribution of sentences could improve the parsing results. Furthermore, using a larger and more diverse few-shot prompt could improve the few-shot annotation quality, resulting in a more accurate trained parser.

Given these possible areas of improvement, another significant limitation is the existence of error propagation in our framework. To address this issue, simplifying the number of steps or using a more unified framework could result in improvement in eventual accuracy.

5.2 Future Work

Knowledge Graphs are a rich and expansive resource. One possible line of future work that we would like to consider is expanding both the number of relations considered as well as exploring our framework on an event-based task with an event-based Knowledge Graph like ATOMIC (Hwang et al., 2020). Additionally, other forms of Knowledge Graphs such as distilled LLM-based Knowledge Graphs (West et al., 2021) could be interesting avenue to explore.

Another line of investigation is providing a more systematic analysis of our results. One significant study that we plan to do is to investigate the coherence and commonsensicalness of our generated text through a human correlation study on quality of generations. Additionally, we could expand our pilot correlation study by by increasing participants and number of rated generations.

There has been significant work using Reinforcement Learning to train Language Models (Ouyang et al., 2022b; Bai et al., 2022). Our framework specifically uses guided-generation from scores generated by our Guided-Generation and Compatibility Module, but it could be interesting to explore using the setting of reinforcement learning using knowledge graph scores.

Finally, one of the most interesting future directions is to extend this framework to event-based commonsense. ATOMIC and ACCENT (Hwang et al., 2020) already supports event commonsense, and guiding with event commonsense scoring could improve tasks like question answering and chain-of-thought reasoning.

APPENDIX A

Prompts

A.1 Few-Shot LLM Parser Prompt

```
Extract a list of tuples (A, B) from the sentence for the relations based on the description.
Put None if there are no tuples.

UsedFor: A is used to do B. B being used for A.
AtLocation: A is at the location or larger area B.
CapableOf: A is capable of doing B
PartOf: A is part of the bigger whole B.

The boy put his foot into the sock.
UsedFor: None
AtLocation: (foot is at location sock)
CapableOf: None
PartOf: (foot is part of boy)

a man sings into a microphone on stage wearing a shirt
UsedFor: (microphone is used to sing), (shirt is used for wear)
AtLocation: (man is at location stage), (microphone is at location stage)
CapableOf: (man is capable of singing into a microphone), (man is capable of wearing a shirt)
PartOf: None

Bride wearing her wedding dress receives help by her bridesmaids wearing red dresses.
UsedFor: None
AtLocation: None
CapableOf: (bride is capable of receiving help), (bridesmaids is capable of wearing red dresses), (bride
    is capable of wearing wedding dress), (bridesmaids are capable of helping)
PartOf: None

I used a chisel and hammer to carve a piece of wood.
UsedFor: (chisel is used to carve), (hammer is used to carve), (wood is used to carve)
AtLocation: None
CapableOf: (I am capable of using a chisel and hammer), (chisel and hammer are capable of carving)
PartOf: None
```

Figure A.1: This is our few-shot prompt for Davinci. We feed it in as a prefix p, and add our sentence x for our input $p + x$.

A.2 Finetuned T5

```
AtLocation_prompt = "Extract a list of tuples (A, B) from the sentence \
                     where A is at the location or larger area B in \
                     the sentence. Sentence: "
UsedFor_prompt    = "Extract a list of tuples (A, B) from the sentence \
                     where A is used to do B or B being used for A \
                     in the sentence. Sentence: "
CapableOf_prompt  = "Extract a list of tuples (A, B) from the sentence \
                     where A is capable of doing B \
                     in the sentence. Sentence: "
PartOf_prompt     = "Extract a list of tuples of (A, B) from \
                     from the sentence where A is part of the \
                     bigger whole B in the sentence. Sentence: "
```

Figure A.2: These are the prompts we use for the T5 parser trained on LLM-annotated data.

APPENDIX B

Correlation Study Specification

```
Score each sentence from 1 to 3 based on how 'commonsensical' the sentence is. Specifically,
     consider the
relations between the concepts in the sentence and the commonsensical—ness of those
    relations.

The rubric is as follows:

1: Is not commonsensical. The sentence has relations that do not obey commonsense knowledge.
2: Is somewhat commonsensical. The sentence does not completely agree with commonsense
    knowledge.
3: Is commonsensical. All relations in this sentence obey commonsense knowledge.

We define 'commonsense knowledge' as the following:
"Commonsense knowledge includes the basic facts about events (including actions) and their
    effects, facts about knowledge and how it is obtained,  facts about beliefs and desires.
     It also includes the basic facts about material objects and their properties."

Score the sentences according to what sentences agree most with the 'commonsense knowledge'
    you possess.

— Only penalize grammatical mistakes if they alter the meaning of the sentence.
```

Figure B.1: This is our specification for our correlation study. Annotators are prompted to rate generations on a scale of 1 to 3, where 1 is not commonsensical and 3 is commonsensical. They are directed to rate it considering commonsense relations within the sentence and to not penalize grammatical mistakes if they don't alter the underlying meaning of the sentence.

REFERENCES

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., Chen, C., Olsson, C., Olah, C., Hernandez, D., Drain, D., Ganguli, D., Li, D., Tran-Johnson, E., Perez, E., Kerr, J., Mueller, J., Ladish, J., Landau, J., Ndousse, K., Lukosuite, K., Lovitt, L., Sellitto, M., Elhage, N., Schiefer, N., Mercado, N., Darma, N., Lasenby, R., Larson, R., Ringer, S., Johnston, S., Kravec, S., Showk, S. E., Fort, S., Lanham, T., Telleen-Lawton, T., Conerly, T., Henighan, T., Hume, T., Bowman, S. R., Hatfield-Dodds, Z., Mann, B., Amodei, D., Joseph, N., McCandlish, S., Brown, T., and Kaplan, J. (2022). Constitutional ai: Harmlessness from ai feedback.

Bommarito, M. and Katz, D. M. (2022). Gpt takes the bar exam.

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

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. *CoRR*, abs/2005.14165.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. (2022). Scaling instruction-finetuned language models.

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

Dong, L. and Lapata, M. (2016). Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany. Association for Computational Linguistics.

Dunn, A., Dagdelen, J., Walker, N., Lee, S., Rosen, A. S., Ceder, G., Persson, K., and Jain, A. (2022). Structured information extraction from complex scientific text with fine-tuned large language models.

Elazar, Y., Kassner, N., Ravfogel, S., Ravichander, A., Hovy, E., Schütze, H., and Goldberg, Y. (2021). Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Ganesan, K. (2018). Rouge 2.0: Updated and improved measures for evaluation of summarization tasks.

Ghazarian, S., Shao, Y., Han, R., Galstyan, A., and Peng, N. (2023). Ghazarian, sarik and shao, yijia and han, rujun and galstyan, aram and peng, nanyun. unpublished.

He, X., Gong, Y., Jin, A.-L., Qi, W., Zhang, H., Jiao, J., Zhou, B., Cheng, B., Yiu, S., and Duan, N. (2022). Metric-guided distillation: Distilling knowledge from the metric to ranker and retriever for generative commonsense reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 839–852, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Krause, B., Gotmare, A. D., McCann, B., Keskar, N. S., Joty, S., Socher, R., and Rajani, N. F. (2021). GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lin, B. Y., Zhou, W., Shen, M., Zhou, P., Bhagavatula, C., Choi, Y., and Ren, X. (2020). CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Liu, Y., Wan, Y., He, L., Peng, H., and Yu, P. S. (2021). Proceedings of the aaai conference on artificial intelligence.

Mahowald, K., Ivanova, A. A., Blank, I. A., Kanwisher, N., Tenenbaum, J. B., and Fedorenko, E. (2023). Dissociating language and thought in large language models: a cognitive perspective.

Marcus, G. (2020). The next decade in AI: four steps towards robust artificial intelligence. *CoRR*, abs/2002.06177.

Meng, T., Lu, S., Peng, N., and Chang, K.-W. (2022). Controllable text generation with neurally-decomposed oracle. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

Niu, C., Shan, H., and Wang, G. (2022). Spice: Semantic pseudo-labeling for image clustering. *IEEE Transactions on Image Processing*, 31:7264–7278.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. (2022a). Training language models to follow instructions with human feedback.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. (2022b). Training language models to follow instructions with human feedback.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Polu, S., Han, J. M., Zheng, K., Baksys, M., Babuschkin, I., and Sutskever, I. (2022). Formal mathematics statement curriculum learning.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Sap, M., Shwartz, V., Bosselut, A., Choi, Y., and Roth, D. (2020). Commonsense reasoning for natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 27–33, Online. Association for Computational Linguistics.

Speer, R., Chin, J., and Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press.

West, P., Bhagavatula, C., Hessel, J., Hwang, J. D., Jiang, L., Bras, R. L., Lu, X., Welleck, S., and Choi, Y. (2021). Symbolic knowledge distillation: from general language models to commonsense models. In *North American Chapter of the Association for Computational Linguistics*.

Xiong, W., Yu, M., Chang, S., Guo, X., and Wang, W. Y. (2019). Improving question answering over incomplete kbs with knowledge-aware reader.

Yang, K. and Klein, D. (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.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *ArXiv*, abs/1904.09675.

Zhou, H., Young, T., Huang, M., Zhao, H., Xu, J., and Zhu, X. (2018). Commonsense knowledge aware conversation generation with graph attention. In *International Joint Conference on Artificial Intelligence*.

Zhou, P., Gopalakrishnan, K., Hedayatnia, B., Kim, S., Pujara, J., Ren, X., Liu, Y., and Hakkani-Tür, D. (2022). Think before you speak: Explicitly generating implicit commonsense knowledge for response generation. In *ACL 2022*.