

SPATIAL LANGUAGE UNDERSTANDING: DEEP LEARNING, REASONING, AND
EVALUATION

By

Roshanak Mirzaee Mazrae

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science—Doctor of Philosophy

2024

ABSTRACT

Spatial language understanding plays an essential role in human communication and perception of the physical world. It encompasses how people describe, understand, and communicate spatial relationships between objects and environmental entities, such as location, orientation, distance, and relative position. Spatial language processing presents numerous challenges, which often stem from the inherent ambiguity of natural language in describing spatial relations or the complexity of spatial reasoning to infer indirect relations, particularly when multi-hop reasoning is needed.

Despite the remarkable achievements of pretrained language models (PLMs) in various natural language processing (NLP) tasks, their effectiveness in spatial language processing has not yet been thoroughly examined. Therefore, in this thesis, we first aim to evaluate these models' performance in multi-hop spatial reasoning. Second, we intend to propose deep learning methods and models that can achieve better multi-hop spatial reasoning performance in both controlled and real-world settings. As a result, this thesis has four main contributions to the understanding and reasoning of spatial language.

The first contribution is proposing novel question-answering benchmarks to evaluate the spatial reasoning capability of deep neural models. These benchmarks include complex and realistic spatial phenomena not covered in previous work, making it more challenging for state-of-the-art PLMs.

The second contribution is an approach to generate large distance supervision for spatial question answering and spatial role labeling tasks to enhance the spatial language understanding of models. We design grammar and reasoning rules to automatically generate a spatial description of scenes and corresponding QA pairs. In this approach, we integrate a diverse set of spatial relation types and expressions, complemented by additional functions, to enhance the flexibility and extensibility of the data generation process. Further training PLMs on this data significantly improves their capability on spatial understanding, thereby enabling them to solve other benchmarks and external datasets better.

Furthermore, **the third** contribution explores the potential benefits of disentangling the processes of information extraction and reasoning in neural models to address the challenges of multi-hop

spatial reasoning. To explore this, we design various models that disentangle extraction and reasoning (either symbolic or neural) and compare them with state-of-the-art baselines with no explicit design for these parts. Our experimental results consistently demonstrate the efficacy of disentangling, showcasing its ability to enhance models' generalizability within realistic data domains.

Ultimately, **the fourth** contribution investigates the role of Large Language Models (LLMs) in multi-hop spatial reasoning tasks, focusing on their performance with and without in-context learning. Besides, we integrate LLMs as extraction modules within a pipeline for extraction and symbolic reasoning. While our case studies in controlled environments indicate the benefits of this idea, our experiments in real-world settings reveal that the model's efficiency decreases due to escalating errors in the extraction process. We also utilize probabilistic logical reasoning and LLMs' commonsense knowledge, improving the pipeline model's performance in real-world applications. Despite these enhancements, the pipeline model continues to exhibit inferior performance compared to standalone LLMs.

Copyright by
ROSHANAK MIRZAEI MAZRAE
2024

ACKNOWLEDGEMENTS

In the name of God,

Embarking on the journey to complete my PhD has been a tremendous undertaking that would not have been possible without the unwavering support and encouragement of numerous individuals who have touched my life in profound ways.

First and foremost, I express my deepest gratitude to my parents, Nasser and Fatemeh. Their boundless love, tireless belief in me, and guidance have been the cornerstone of my academic pursuits since the very beginning. They not only instilled in me the courage to dream big and the resilience to pursue those dreams tirelessly, but also made countless sacrifices to ensure I had the opportunities I needed to succeed.

To my spouse, teammate, and most importantly, best friend, Hossein, who has been my pillar of strength, confidant, and closest ally throughout this challenging journey, I owe an immeasurable debt of gratitude. His unwavering support was light through the darkest times, inspiring me to persevere and push beyond my limits.

Additionally, I extend my heartfelt thanks to my sisters, Faranak and Baharak, and my friends, whose willingness to lend a listening ear or a helping hand has made all the difference.

To all those who have supported me, believed in me, and encouraged me along the way, I offer my sincerest thanks. Their faith in me has been a driving force behind my success, and I am profoundly grateful for their role in shaping my academic journey.

TABLE OF CONTENTS

CHAPTER 1	INTRODUCTION	1
1.1	Motivation	1
1.2	Challenges and Contributions	3
CHAPTER 2	BACKGROUND KNOWLEDGE AND RELATED WORKS	7
2.1	Information Extraction	7
2.2	Question Answering Task	10
2.3	Pretrained Language Models	13
2.4	Transfer Learning	16
2.5	Deterministic Vs. Probabilistic Reasoning	16
CHAPTER 3	EVALUATION OF PRETRAINED LANGUAGE MODELS	18
3.1	Introduction	18
3.2	SPARTQA-HUMAN Collection	20
3.3	SPARTQA-AUTO	23
3.4	Spatial Question Answering Models	29
3.5	Experiments and Results	31
3.6	Related Research	35
3.7	Summary	36
CHAPTER 4	TRANSFER LEARNING FOR SPATIAL QUESTION ANSWERING	37
4.1	Introduction	37
4.2	SPARTUN: A Synthetic Data with Broader Relation Coverage	39
4.3	Spatial Question Answering in Real-World Domain	45
4.4	Experiments and Results	47
4.5	Related Research	56
4.6	Summary	57
CHAPTER 5	SPATIAL ROLE LABELING MODELS	58
5.1	Introduction	58
5.2	Models	58
5.3	Experiments and Results	60
5.4	Related Research	62
5.5	Summary	62
CHAPTER 6	DISENTANGLEING EXTRACTION AND REASONING IN MULTI- HOP SPATIAL REASONING	64
6.1	Introdcution	64
6.2	Related Research	66
6.3	Proposed Models	67
6.4	Experiments	72
6.5	Results and Discussion	74
6.6	Summary	81

CHAPTER 7	EXPLORING SPATIAL REASONING WITH LARGE LANGUAGE MODELS	83
7.1	Introduction	83
7.2	Large Language Models (LLM)	85
7.3	Probabilistic Spatial Reasoning	88
7.4	Experiments and Results	91
7.5	Summary	102
CHAPTER 8	CONCLUSION AND FUTURE DIRECTIONS	103
8.1	Summary of Contributions	103
8.2	Future Directions	105
PUBLISHED PAPERS	107
BIBLIOGRAPHY	108

CHAPTER 1

INTRODUCTION

1.1 Motivation

Spatial language understanding plays a crucial role in human communication and perception of the physical world. Spatial language refers to the way people describe, understand, and use the spatial relationships between objects and entities in the environment, such as location, orientation, distance, and relative position. Understanding spatial language is important in many natural language processing (NLP) applications such as navigation [156, 155, 9, 111, 52], medical domain [16, 47, 81], robotics [134, 49], human-machine interaction [60, 111], dialogue systems [130], and natural language interfaces for smart homes and buildings [96, 48, 63]. These applications can become more intuitive and effective by enabling machines to understand and interpret spatial language. In addition, spatial language understanding can provide insights into human cognition and perception.

On the other hand, recent transformer-based pretrained language models¹ (PLMs) have already demonstrated exceptional performance in various natural language processing tasks. However, their capacity for spatial language understanding and reasoning remains uncertain. As a result, this thesis seeks to evaluate the strengths and weaknesses of PLMs in this context. Following this assessment, our next goal is to develop methods and models with enhanced spatial language understanding. This could involve improving the spatial reasoning capabilities of existing models or designing entirely new architectures to achieve better performance. Ultimately, these advances will contribute to developing more effective systems that rely on spatial language understanding.

We describe spatial language understanding as identifying explicit spatial concepts, such as entities and relations directly mentioned in the text, followed by reasoning to deduce implicit relations that can be logically inferred. We term the former task as Spatial Role Labeling (SpRL) and the latter one as Spatial Reasoning.

Figure 1.1 shows an example of Spatial Role Labeling to find the explicit relations, such as

¹For brevity, we will refer to these as language models or PLMs.

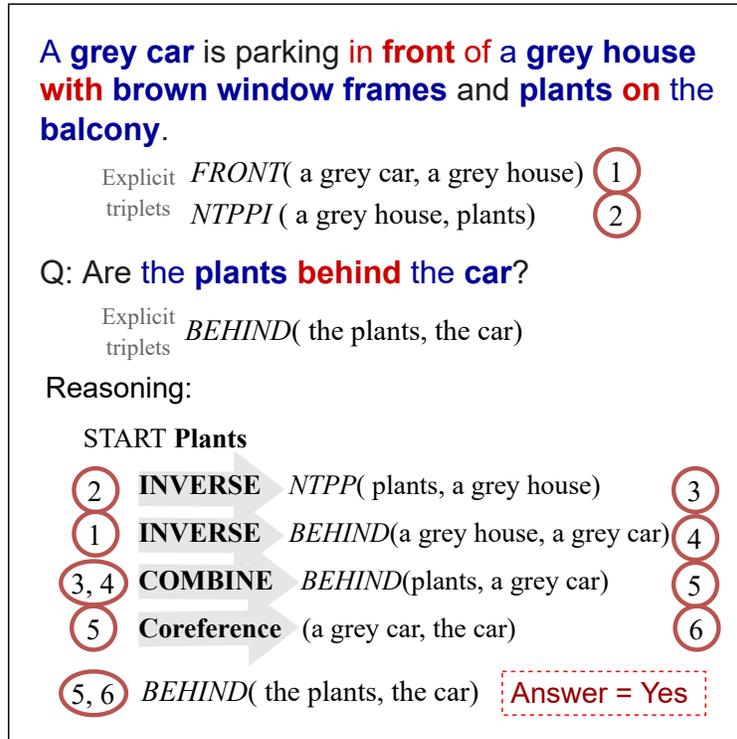


Figure 1.1 Spatial QA example from RESEQ dataset. To do reasoning, we start with searching for *the plants* in the question triplet, extract the explicit triplets (1,2), use spatial rules (e.g. *INVERSE*) to infer implicit triplets (3,4,5), and in the last step, use 5 and 6 to answer the question triplet. NTPP: Non-Tangential Proper Part (Table 2.1).

“FRONT(a grey car, a grey house)” and Spatial Reasoning to find implicit ones such as “BEHIND (the plants, the car).” Extracting implicit relations requires knowledge of spatial rules over various spatial relation types. For instance, logically, we know that if “a grey car” is in front of “a grey house”, then “the car” will be behind “the house”. This is an example of a reasoning rule also known as an *inverse* rule.

Traditionally, evaluating general language understanding has involved formal spatial semantic extractions and mapping language to symbolic representations that machines can reason over. However, in recent years, question-answering (QA) has been used as a criterion for evaluating deep neural models, such as language models. In the QA setting, the model is given both textual context and question, and the latent layers of these models are expected to perform extraction and reasoning to predict the final answer. For example, given the context and question in Figure 1.1, a language model [119, 70] can predict the final answer (Yes) that means the BEHIND relation holds

between the plants and the car without explicit extraction and reasoning. In this work, we follow both evaluation paradigms to address the challenges of spatial language understanding. We explore explicit semantic representations and reasoning and their influence on question-answering when complex reasoning is required. In the remainder of this chapter, we discuss the existing challenges in this research area and outline our approach to addressing them.

1.2 Challenges and Contributions

Spatial language processing tasks entail numerous challenges, which may stem from, firstly, the natural language ambiguity in describing relations and, secondly, the complexity of spatial reasoning to infer indirect relations. This study focuses on the latter case, specifically the multi-hop spatial reasoning challenge in question answering. Following, we identify two main challenges in this study and our four contributions to address them.

Challenge 1: Lack of Evaluation Benchmark. Modern language models (LM), e.g., BERT [18], ALBERT [59], and XLNet [147] have seen great successes in natural language processing (NLP). However, there has been limited investigation into *spatial reasoning capabilities of LMs*. To the best of our knowledge, bAbI [143] is the only dataset with direct textual spatial question answering ² which is synthetic and overly simplified. In particular, (1) The underlying scenes are spatially simple, with only three objects and relations only in four directions. (2) The stories for these scenes are too short, templated sentences, each describing a single relation between two objects. (3) The questions typically require up to two steps of reasoning due to the simplicity of those stories. Hence, the evaluation of LMs on this dataset does not demonstrate their actual spatial reasoning capability.

Contribution 1: To address these issues, this thesis proposes three new datasets. The first one is a human-curated benchmark, SPARTQA-HUMAN. Specifically, this dataset provides the following characteristics, (1) it is built on images containing more objects with richer spatial structures. (2) Its stories are more natural, have more sentences, and are richer in spatial relations in each sentence. (3) Its questions require deeper reasoning and have four types: *find relation* (FR), *find blocks* (FB), *choose object* (CO), and *yes/no* (YN), which allows for more fine-grained analysis of models'

²At the time of this project.

capabilities. Using this dataset, we evaluate the performance of modern LMs on multi-hop spatial reasoning.

The *human-generated datasets*, despite helping study the problem as evaluation benchmarks, are less helpful for training models that can reliably understand spatial language due to their small size. *As a solution*, this work proposes a way to obtain **large** synthetic supervision signals for spatial reasoning, called SPARTQA-AUTO. As spatial relationships are rarely mentioned in existing corpora, we take advantage of spatial language being grounded in the geometry of visual scenes. Thus, we generate stories for existing synthetic images [123] automatically via our newly designed context-free grammars (CFG) and context-sensitive rules. Then, we evaluate LMs’ spatial reasoning capability on this large and complex benchmark.

We also provide another human-generated dataset, named RESQ, to evaluate neural models on real-world spatial question answering. Compared to SPARTQA-HUMAN, this benchmark is more readable for humans and needs both spatial and commonsense reasoning to be answered.

Challenge 2: Complexity of Multi-hop Spatial Reasoning in Neural Models. After evaluating the spatial reasoning capability of language models, enhancing this feature in them is a challenging and complex task that requires a multidisciplinary approach involving advances in dataset creation, neural network architectures, and training methodologies. As we mentioned before, the deep neural models’ reliance on a large amount of training data may not always be feasible in practical scenarios (e.g., SPARTQA-HUMAN, RESQ). Besides, the existing large *synthetic datasets* (e.g., bAbI, SPARTQA-AUTO) often focus on specific types of relations with a small coverage of spatial semantics needed for spatial language understanding in various domains.

Contribution 2: As a solution, we use transfer learning as an effective method [109, 2, 77] for pretraining and further fine-tuning PLMs on our target tasks. Transfer learning is a simple way of using synthetic data as an extra source of supervision at no annotation cost. SPARTQA-AUTO proves to be a rich source of spatial knowledge that improved the performance of LMs on SPARTQA-HUMAN and on different data domains. We also build another synthetic dataset, called SPARTUN, to serve as a source of supervision with *broad coverage of spatial relation types and expressions* (which is easily

extendable to other reasoning tasks). We evaluate the transferability of the models pretrained with SPARTUN on multiple spatial question answering and spatial information extraction benchmarks and show significant improvements in SOTA results.

Contribution 3: While transfer learning helps improve spatial reasoning, the challenge still remains due to the significant gap between human and machine performance. As another direction to solve spatial reasoning, we investigate the interoperability between extraction and reasoning and evaluate whether disentangling information extraction and reasoning at the modeling level can help solve spatial reasoning tasks. To explore this, we design various models that disentangle extraction and reasoning (either symbolic or neural) and compare them with state-of-the-art (SOTA) baselines with no explicit design for these parts. We assessed the performance of our proposed models using various SQA datasets, showcasing the efficiency of our methods in both controlled and real-world settings. In controlled environments, our framework, which used separated extraction and symbolic reasoning, outperformed existing SOTA models by a significant margin, even when using the same or fewer training data. However, in real-world situations characterized by higher ambiguity in natural language and more complex rules, our model with distinct layers for extraction and reasoning, demonstrated superior performance.

Contribution 4: In our final contribution, we investigate the role and effectiveness of Large Language Models (LLMs) in spatial reasoning tasks. First, we evaluate them as an end-to-end model on the SQA task and highlight their shortcomings. Considering the strength of LLMs in information extraction, we also utilized LLMs as the extraction module within a pipeline of disentangled extraction and reasoning. In this model, the LLM is responsible for extracting direct and commonsense spatial relations from the text, which are then passed to deterministic or probabilistic spatial reasoners. Our case studies and prior experimental results on controlled environments demonstrate that this methodology effectively addresses spatial reasoning problems that end-to-end LLMs find challenging. Our experiments on a realistic environment demonstrate that probabilistic reasoning outperforms deterministic reasoning in this pipeline, and employing LLMs’ commonsense knowledge enhances its performance by covering some missed commonsense rules and relations

from the designed spatial reasoners. However, considering the inherent challenges of LLMs in extracting spatial information from ambiguous natural language, this approach faces limitations in outperforming end-to-end LLMs in real-world settings.

CHAPTER 2

BACKGROUND KNOWLEDGE AND RELATED WORKS

This chapter aims to provide a comprehensive overview of the relevant literature and research conducted in our research. We start with question-answering and information extraction, two fundamental natural language processing tasks. Question answering involves answering questions posed in natural language [107, 148], while information extraction involves identifying and extracting relevant information from unstructured text [151]. Specifically, we focus on two spatial tasks: spatial question answering (useful for spatial reasoning) and spatial information extraction (a.k.a spatial role labeling (SpRL)), where the goal is to answer questions or extract information about spatial entities, such as locations, directions, and distances. This chapter surveys related work on these tasks, including existing benchmarks and recent advances in deep learning methods, such as transformers and transfer learning, a widespread technique in natural language processing for leveraging pre-trained models to improve performance on downstream tasks. The chapter concludes by describing deterministic and probabilistic reasoning and their distinctions.

2.1 Information Extraction

Information extraction (IE) is the process of automatically extracting structured information from unstructured or semi-structured text [114]. This task aims to identify and extract specific information, such as named entities, events, and relationships between entities. One key challenge in IE is dealing with the variability and ambiguity of natural language text. To address this challenge, researchers have developed various approaches, including rule-based methods [145, 140, 28], statistical methods [141], and deep learning [66, 26] methods. These approaches have been applied to various areas, including news articles [153, 44], and social media data [75, 7].

Several benchmarks have been provided on this task, such as DocRED [151], SciERC [73], TAC KBP 2017 [31] and ACE 2005 [136]. These benchmarks have been used to evaluate the performance of various IE models that have achieved SOTA performance. Below you can see some examples of proposed models to tackle IE challenges:

BERT-based models: BERT is a pre-trained language model (Check Sec 2.3) fine-tuned for various IE tasks, including named entity recognition and relation extraction [18]. Several studies have shown that BERT-based models can achieve SOTA performance on IE tasks (e.g., [68, 119]).

Graph-based models: Graph-based models have been proposed for relation extraction tasks, where entities are represented as nodes in a graph and relations are represented as edges between nodes [154]. These models gain competitive performance on relation extraction tasks.

Multi-task learning models: Multi-task learning models can jointly learn multiple IE tasks, such as named entity recognition and relation extraction, using a single model [73]. These models achieve competitive performance on multiple IE tasks.

Few-shot learning models: Few-shot learning models can be used to adapt to new IE tasks with limited labeled data. For example, GPT-3 (Generative Pre-trained Transformer 3) performs well on named entity recognition and relation extraction tasks with only a few labeled examples [8].

2.1.1 Spatial Information Extraction

The white car in the street, is in front of the blue building.

Trajector1 = The white car
Landmark1 = the street
Spatial_indicator = in
General type = Topological (RCC8 (TPP))

TPP(the white car, the street)

Trajector1 = The white car
Landmark1 = the blue building
Spatial_indicator = in front
General type = Directional (FRONT)

FRONT(the white car, the blue building)

Figure 2.1 An example from mSPRL benchmark with spatial Role Labeling annotation.

Spatial information extraction, a.k.a spatial role labeling [54, 56] (SpRL) is the task of identifying and classifying the spatial roles (Trajector, Landmark, and spatial indicator) and their relations. We

call the former **spatial role extraction** and the latter **spatial relation¹ extraction**. Initially, we provide an overview of the spatial roles employed in this task. You can see an example of SpRL identification in Figure 2.1. The study by [56] defines three distinct roles: Trajector, Landmark, and Spatial indicator.

Trajector: is an entity referred to by its location or position. It can be static or dynamic, referring to people, objects, or events. In this study, we only focus on the static form of Trajectors.

Landmark: is the reference entity concerning which the location or motion of the Trajector is indicated.

Spatial indicator: is an expression that establishes constraints on the spatial properties of the Trajector, such as its location to the Landmark (e.g., in, on).

Spatial relation: is defined as a triplet of (Trajector, Spatial indicator, Landmark) with an assigned relation type which shows the relation between Trajector, and Landmark using Spatial indicator.

SpRL is particularly challenging due to the *ambiguity of natural language* in expressing spatial relations [6]. For instance, in Figure 1.1, it is ambiguous whether both “the window” and “plants” are on “the balcony” or only the “plants”, or in the phrases “plants on the balcony” and “plants on the wall”, *on* refers to a different position between plants, the balcony, and the wall. As a solution for this issue, we extract spatial relations first and map them to a formal representation (see Table 2.1) such as mapping (plants, on, the balcony) to NTPP and (plants, on, the wall) to FRONT and EC.

Few benchmarks exist with spatial roles and relation annotations, which are small-scale datasets for studying the SpRL problem.

mSpRL [55] is a human-curated dataset provided on SpRL task. This dataset contains spatial descriptions of real-world images collected from image CLEF IAPR TC-12 dataset [33] and corresponding SpRL annotations. A SpRL can have the following spatial semantic component [162] on the static environment, **trajector** (the main entity), **landmark**(the reference entity), and **spatial_indicator** (the spatial term describing the relationship between trajector and landmark.). The dynamic environment can also have *path*, *region*, *direction*, and *motion*. Figure 2.1 shows an

¹In different works like [54], the triplet and relation are used interchangeably.

example of context and spatial role labeling of mSPRL benchmark. The previous SOTA on this benchmark belongs to “R-Inf” [78], which uses external multi-modal resources and global inference. **SpaceEval (SemEval-2015 task 8) [1]** is a task in SemEval-2015 that focuses on evaluating models for the extraction of spatial information from text. The goal of SpaceEval was to identify and classify spatial relations between entities mentioned in the text, as well as to identify spatial signals that indicate the presence of a spatial relation. The task involved annotating a corpus of texts with spatial information, such as identifying the types of spatial relations that exist between entities in the text, such as “located in,” “part of,” and “near.” The corpus consisted of a set of texts from different domains, such as news articles, scientific papers, and travel guides. The previous SOTA models on this benchmark use a generation and a classification component for generating and extracting relations [139], or a model containing RoBERTa as word representation and a CRF [95].

2.2 Question Answering Task

Question answering is a helpful format to evaluate machines’ capability of reading comprehension [30] and many recent works have been implementing this strategy to test machines’ understanding of linguistic formalisms: [37, 85, 64, 45, 97, 21]. An important advantage of QA is using natural language to annotate natural language, thus having the flexibility to get annotations on complex phenomena such as *spatial reasoning*. Besides, question answering tasks can get even more challenging when requiring multi-hop reasoning. Recently, pretrained language models achieved the SOTA result on most of the QA benchmarks[70, 142]. We discuss language models more in the next section.

2.2.1 Spatial Question Answering

Despite the high performance of recent deep neural models on question answering (QA) tasks, solving questions that require multi-hop reasoning is still challenging [82]. Usually, various types of reasoning like spatial [160], temporal [106], numerical [92], commonsense [127] and many more are needed to solve QA tasks. In this work, we focus on spatial reasoning and, as a result, on spatial question answering tasks. Below, we first describe spatial reasoning and then describe it as the task of spatial question answering.

Formalism (General Type)	Specific value	Spatial type/Spatial value)	Expressions (e.g.)
Topological	RCC8	DC (disconnected) EC (Externally Connected) PO (Partially Overlapped) EQ (Equal) TPP (Tangential Proper Part) NTPP (Non-Tangential Proper Part) TPPI (Tangential Proper Part inverse) NTPPI (Non-Tangential Proper Part inverse)	disjoint touching overlapped equal covered by in, inside covers has
Directional	Relative	LEFT, RIGHT BELOW, ABOVE BEHIND, FRONT	left of, right of under, over behind, in front
Distance	Qualitative	Far, Near	far, close

Table 2.1 Spatial relation types and examples of spatial language expressions.

2.2.1.1 Spatial Reasoning

Spatial reasoning over text can be described as inferring the implicit² spatial relations from direct relations described in the text. To better understand spatial reasoning, we must first describe two key concepts: spatial relation types and rules. Table 2.1 presents sixteen 3D **spatial relation types** collected from existing resources [108, 144, 110]. These relations have been categorized into three groups: Directional, Topological, and Distance relations.

Directional relations describe how objects or entities are positioned to each other in terms of orientation or direction.

Topological relations refer to spatial relationships between objects or entities based on their relative position and shape rather than their orientation or direction. These relationships describe how objects or entities are spatially connected to each other, such as touching, overlapping, or containing. Figure 2.2 shows RCC8, eight different topological relations described in [110].

Distance relations are a scalar entity and can be expressed either qualitatively as "near" or "far," or quantitatively as, for instance, "two meters far" [54]. This work, however, concentrates only on the former type of expression.

Spatial rules refer to the logical principles that govern how spatial relations operate and interact with each other. These rules can assist in reasoning about and inferring spatial relations from

²By implicit, we meant indirect relations, not metaphoric usages or implicit meaning for the relations.

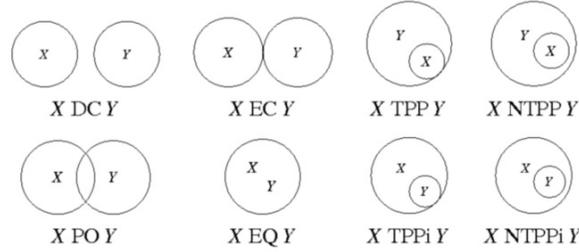


Figure 2.2 Eight topological relations in RCC8. Images from Wikipedia.

Not	$\forall(X, Y) \in Entities$	$R \in \{Dir \vee PP\}$	$IF R(X, Y)$	$\Rightarrow NOT(R_reverse(X, Y))$
Inverse	$\forall(X, Y) \in Entities$	$R \in \{Dir \vee PP\}$	$IF R(Y, X)$	$\Rightarrow R_reverse(X, Y)$
Symmetry	$\forall(X, Y) \in Entities$	$R \in \{Dis \vee (RCC - PP)\}$	$IF R(Y, X)$	$\Rightarrow R(X, Y)$
Transitivity	$\forall(X, Y, Z) \in Entities$	$R \in \{Dir \vee PP\}$	$IF R(X, Z), R(Z, Y)$	$\Rightarrow R(X, Y)$
Combination	$\forall(X, Y, Z, H) \in Entities$	$R \in Dir, *PP \in PP$	$IF *PP(X, Z), R(Z, H), *PPi(Z, Y)$	$\Rightarrow R(X, Y)$

Table 2.2 Designed spatial rules. *Dir*: Directional relations (e.g., LEFT), *Dis*: Distance relations (e.g., FAR), *PP*: all Proper parts relations (NTPP, NTPPI, TPPI, TPP), *RCC - PP*: All RCC8 relation except proper parts relations. **PP*: one of TPP or NTPP. **PPi*: one of NTPPi or TPPI.

text [144, 20]. For example, the transitivity rule for topological relations states that if object A is "inside" object B, and object B is "inside" object C, then object A is "inside" object C. Similarly, the inverse rule for directional relations states that if object A is "behind" object B, then object B is "in front of" A. Understanding these spatial rules is crucial for accurate spatial language understanding and reasoning. We define five spatial rules in our work, Not, Inverse, Symmetry, Transitivity, and Combination, which all are defined in Table 6.2.

Given a set of spatial primitives, relation types, and rules, we define spatial reasoning as the procedure of applying spatial rules to explicit relation types in written language, aiming to infer implicit relations.

In spatial question answering, given a scene description, the task is to answer questions about the spatial relations between entities. Here, we focus on challenging questions that need multi-hop spatial reasoning over explicit relations. Figure 1.1 shows an example of the spatial reasoning process over a given text. To find the answer to the question, we first need to identify all spatial entities and relations (SpRL) and then apply rules to find the asked relation.

Spatial reasoning phenomena have been covered minimally in the existing benchmarks we mentioned. To the best of our knowledge, Task 17 and 19 of the bAbI project [143] is the only QA

dataset focused on textual spatial reasoning until our project. Task 17 is on spatial reasoning and contains binary Yes/No questions. Task 19 is on pathfinding and contains FR questions with answers in {West, East, North, South} set. The original dataset contains west, east, north, and south, which we mapped to their corresponding relative relation type. bAbI is synthetic and does not reflect the complexity of spatial reasoning in natural language. Solving Task 17 and 19 of bAbI typically does not require sophisticated reasoning, which is an important capability emphasized by more recent works (e.g., [23, 50, 148, 14, 97]).

2.3 Pretrained Language Models

Pretrained language models (PLM) are large neural network models that have been trained on vast amounts of text data such as the entire text of Wikipedia or the Common Crawl web corpus in a un/supervised way. These models use transformers to learn the underlying patterns and structure of natural language and can then be fine-tuned for specific NLP tasks such as question answering [107], information extraction [75], or sentiment analysis [79].

2.3.1 Transformer

Transformers are deep neural network architecture introduced in [133] that significantly impacts NLP tasks. Unlike previous neural network architectures, which relied on recurrent neural networks (RNNs) or convolutional neural networks (CNNs), transformers use an attention mechanism to capture dependencies between different parts of a sequence of inputs. The transformer architecture enables the model to learn long-range dependencies between different parts of a sentence, allowing it to capture the context and meaning of words in a way that previous neural network architectures could not.

The basic building block of a transformer is the self-attention mechanism, which computes the importance of each word in a sequence relative to every other word in the sequence. This allows the model to focus on the most relevant words in the sequence when making predictions. The transformer architecture consists of a series of encoder layers to process the input sequence and decoder layers to generate the output sequence. Each layer contains a multi-head self-attention mechanism, which allows the model to attend to different parts of the sequence simultaneously, as well as a feed-forward

neural network that processes the output of the self-attention mechanism. During training, the model is typically pre-trained on a large corpus of text using a un/supervised learning task, such as masked language modeling (MLM) or next sentence prediction. This pretraining enables the model to learn a general representation of natural language that can be fine-tuned on specific downstream tasks with smaller datasets. During inference, the model takes an input sequence and generates an output sequence by iteratively processing the sequence through the encoder and decoder layers. Overall, transformers have revolutionized natural language processing by enabling models to capture better the complex relationships between different parts of a sequence and achieve state-of-the-art performance on a range of NLP benchmarks.

Once pretrained, language models that use the transformer's flexibility and effectiveness can be fine-tuned on specific NLP tasks with much smaller amounts of data. Fine-tuning involves training the model on a smaller, task-specific dataset to improve its performance on that particular task. Using pretrained language models as a starting point, NLP models can achieve state-of-the-art performance on a wide range of tasks with less data and training time than would be required to train a model from scratch. Some examples of popular pretrained language models are:

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model that uses a bidirectional transformer to learn contextual relations between words in a text. BERT uses bidirectional context, where the model is trained to predict missing words in a sentence based on the surrounding words on both sides rather than just the words that come before or after the missing word. This allows BERT to capture more complex relationships between words and produce more accurate predictions. BERT has achieved SOTA performance on various NLP tasks, including question answering.

RoBERTa (Robustly Optimized BERT Pretraining Approach) [70] is an improved version of BERT. RoBERTa uses a larger training corpus and longer training time to perform even better than BERT on several NLP benchmarks, including question answering.

ALBERT (A Lite BERT) is a modification to improve efficiency and reduce the number of parameters in BERT, which makes it a useful model for low-resource environments and applications

where memory and computation resources are limited. ALBERT performs better than BERT on several NLP tasks using fewer parameters.

XLNet [147] is another pre-trained language model that uses a permutation-based approach to capture both forward and backward dependencies between words in a text. While BERT predicts the probability of the next word in a sequence given the previous words, XLNet takes a different approach and uses a sequence-to-sequence auto-regressive model to predict the probability of each word in the sequence given all of the previous words. This enables the model to capture bidirectional relationships between the words in the sequence without needing an MLM objective. XLNet training objective called "permutation language modeling," which samples a random permutation of the input sequence and trains the model to predict the probability of the next word in the sequence given the previous words in the sampled permutation.

T5 (Text-to-Text Transfer Transformer) [104] is a transformer-based model designed to generalize a wide range of natural language tasks by casting them all as a text-to-text problem, where the model maps a given input text to an output text. This approach allows T5 to generalize to new tasks without additional training by simply changing the task representation in the input.

GPT-3 (Generative Pre-trained Transformer 3) [8] is an autoregressive language model that uses a transformer-based architecture. It is trained using a general language modeling objective, where the model is trained to predict the next word in a given sequence of text. This unsupervised pretraining allows the model to learn a rich representation of language, capturing both local and global dependencies in text. GPT-3 is notable for its massive size, with 175 billion parameters, making it the largest publicly available language model to date. While GPT-3 has shown significant improvements in language generation and understanding, it also has limitations, such as a tendency to generate biased or offensive language and difficulty understanding context and sarcasm in text. Nevertheless, GPT-3 has shown the potential for large-scale language models (LLM) to transform the field of natural language processing and enable a wide range of new applications.

2.4 Transfer Learning

Transfer learning has been used effectively in different NLP tasks to further fine-tune the PLMs [25, 120, 109, 2, 77]. This is a machine learning technique where a model is first trained on a large dataset or task. Then, the knowledge gained from that training is applied to a different, usually smaller, dataset or task. This approach can be more efficient than training a model from scratch on a smaller dataset or task, as the pre-trained model already has a rich representation of the underlying features of the data. Using the benefits of pretrain language models such as BERT or GPT3, which already have been pretrained on large corpora, and next fine-tuning them on a smaller, labeled dataset for a specific task is a hot trend in today's NLP research. By leveraging the pretrained language model's knowledge of the language, the fine-tuned model can often achieve SOTA performance with much less training data than would be required to train a model from scratch.

Besides transfer learning, several other approaches are used to tackle the lack of training data in various NLP areas, such as providing techniques to label the unlabeled data [24], using semi-supervised models [132, 65] or data augmentation with synthetic data [67, 86]. However, transfer learning is a simple way of using synthetic data as an extra source of supervision at no annotation cost. Compared to the augmentation methods, the data in the transfer learning only needs to be close to the target task/domain [76] and not necessarily the same.

2.5 Deterministic Vs. Probabilistic Reasoning

Deterministic Reasoning involves making decisions based on certain, fixed rules or logic. In this approach, the outcome of a given situation is determined by the specific conditions and rules applied to it. If the same conditions and rules are applied again, the outcome will be the same every time. Deterministic reasoning is often used when the rules and relationships between variables are well-known and do not change.

Probabilistic Reasoning involves making decisions based on probabilities and uncertainty. In this approach, the outcome of a given situation is not fixed and can vary depending on the probabilities of different events. Probabilistic reasoning is often used in uncertain scenarios or when the rules

and relationships between different variables are not well-known.

The main difference between deterministic and probabilistic reasoning is how they deal with uncertainty. Deterministic reasoning assumes that the outcome of a situation is certain and can be predicted exactly if the rules and conditions are known. Probabilistic reasoning acknowledges that the outcome of a situation may be uncertain and can vary based on probabilities.

CHAPTER 3

EVALUATION OF PRETRAINED LANGUAGE MODELS

3.1 Introduction

Modern language models (LM), e.g., BERT [18], ALBERT [59], and XLNet [147] have seen great successes in natural language processing (NLP). However, there has been limited investigation into *spatial reasoning capabilities of LMs*. To the best of our knowledge, bAbI [143] (Figure 3.1) is the only dataset with direct textual spatial question answering (QA) (Task 17), but it is synthetic and overly simplified: (1) The underlying scenes are spatially simple, with only three objects and relations only in four directions. (2) The stories for these scenes are two short, templated sentences, each describing a single relation between two objects. (3) The questions typically require up to two steps of reasoning due to the simplicity of those stories.

**“The pink rectangle is below the red square.
The red square is below the blue square.”**
1. Is the red square below the pink rectangle? No
2. Is the pink rectangle below the blue square? Yes

Figure 3.1 An example of bAbI dataset, task 17.

To address these issues, in this chapter, we propose a new dataset, SPARTQA¹ (see Fig. 3.2). This benchmark contains more realistic spatial phenomena not covered by prior work and is challenging for state-of-the-art language models (LM). Specifically, (1) SPARTQA is built on NLVR’s [123] images containing more objects with richer spatial structures (Fig. 3.3b). (2) SPARTQA’s stories are more natural, have more sentences, and are richer in spatial relations in each sentence. (3) SPARTQA’s questions require deeper reasoning and have four types: *find relation* (FR), *find blocks* (FB), *choose object* (CO), and *yes/no* (YN), which allows for more fine-grained analysis of models’ capabilities.

We showed annotators random images from NLVR and instructed them to describe objects and relationships not exhaustively at the cost of naturalness (Sec. 3.2). In total, we obtained 1.1k unique QA pair annotations on spatial reasoning, evenly distributed among the aforementioned

¹SPatial Reasoning on Textual Question Answering.

Story:
There are three blocks called A, B, and C. A is to the left of B and B is to the left of C. In A, there is a medium blue square near and to the left of a medium black triangle. The medium black triangle is near and below a medium yellow square. In B, there is a small blue square **far to the left of** a large black circle. There is a medium yellow circle **far above** the black circle. In C, there is a large yellow square.

Q1: Is the medium yellow thing in A near and above a black thing? **Yes**

Q2: What is the relation between the medium yellow thing in A and the medium black thing in A? **Above, Near**

Figure 3.2 An example story and corresponding questions and answers in SPARTQA-HUMAN.

types. Similar to bAbI, we keep this dataset on a relatively small scale and suggest to use as little training data as possible.

Here thus, we propose a way to generate a large synthetic benchmark for spatial reasoning (Sec. 3.3). As spatial relationships are rarely mentioned in existing corpora, we take advantage of the fact that spatial language is grounded to the geometry of visual scenes. We are able to generate stories for NLVR images automatically [123] via our newly designed context-free grammars (CFG) and context-sensitive rules. In the story generation process, we store the information about all objects and relationships such that QA pairs can also be generated automatically. In contrast to bAbI, we use various spatial rules to infer new relationships in these QA pairs, which requires more complex reasoning capabilities. Hereafter, we call this automatically-generated dataset SPARTQA-AUTO and the human-annotated one SPARTQA-HUMAN.

Our contributions can be summarized as follows:

First, we propose the first human-curated benchmark, SPARTQA-HUMAN, for spatial reasoning with richer spatial phenomena than the prior synthetic dataset bAbI (Task 17). Experiments show that modern LMs (e.g., BERT) do not perform well in this low-resource setting.

Second, we exploit the scene structure of images and design novel CFGs and spatial reasoning rules to automatically generate data (i.e., SPARTQA-AUTO) for spatial reasoning over text.

Third, we also proposed the consistency and contrast sets to further evaluate the consistency of

STORY:

We have three blocks, A, B and C. Block B is to the right of block C and it is below block A. Block A has two black medium squares. Medium black square number one is below medium black square number two and a medium blue square. It is touching the bottom edge of this block. The medium blue square is below medium black square number two. Block B contains one medium black square. Block C contains one medium blue square and one medium black square. The medium blue square is below the medium black square.

QUESTIONS:

FB: Which block(s) has a medium thing that is below a black square? *A, B, C*

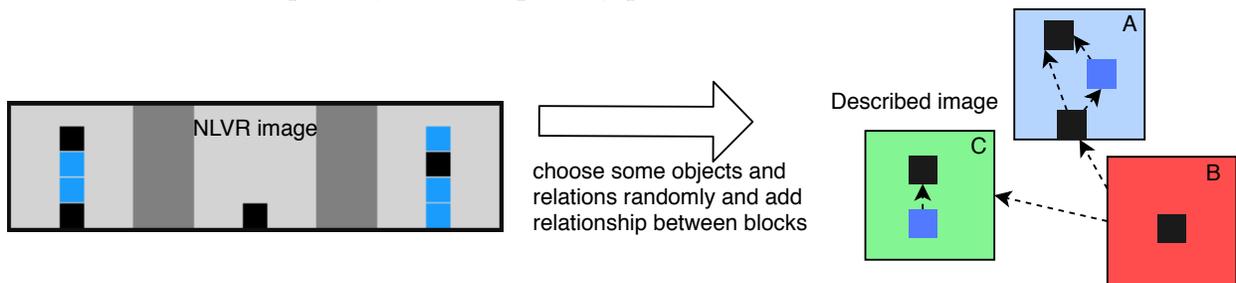
FB: Which block(s) doesn't have any blue square that is to the left of a medium square? *A, B*

FR: What is the relation between the medium black square which is in block C and the medium square that is below a medium black square that is touching the bottom edge of a block? *Left*

CO: Which object is above a medium black square? the medium black square which is in block C or medium black square number two? *medium black square number two*

YN: Is there a square that is below medium square number two above all medium black squares that are touching the bottom edge of a block? *Yes*

(a) An example story and corresponding questions and answers in SPARTQA-AUTO.



(b) An example NLVR image and the scene created in Fig. 3.3a, where the blocks in the NLVR image are rearranged.

Figure 3.3 Example from SPARTQA (specifically from SPARTQA-AUTO).

the spatial reasoning capability of models.

3.2 SPARTQA-HUMAN Collection

To mitigate the aforementioned problems of Task 17 of bAbI, i.e., simple scenes, stories, and questions, we describe the data annotation process of SPARTQA-HUMAN, and explain how those problems were addressed in this section.

First, we randomly selected a subset of NLVR images, each of which has three blocks containing multiple objects (see Figure 3.3b). The scenes shown by these images are more complicated than those described by bAbI because (1) there are more objects in NLVR images; (2) the spatial relationships in NLVR are not limited to just four relative directions as objects are placed arbitrarily within blocks.

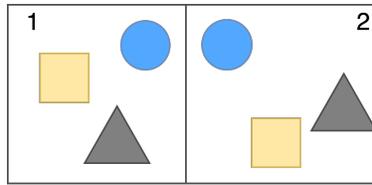


Figure 3.4 For “A blue circle is above a big triangle. To the left of the big triangle, there is a square,” if the question is: “Is the square to the left of the blue circle?”, the answer is neither Yes nor No. Thus, the correct answer is “Do not Know” (DK) in our setting.

Second, two student volunteers produced a textual description of those objects and their corresponding spatial relationships based on these images. Since the blocks are always horizontally aligned in each NLVR image, to allow for more flexibility, annotators could also rearrange these blocks (see Fig. 3.2). Relationships between objects within the same block can take the forms of relative direction (e.g., left or above), qualitative distance (e.g., near or far), and topological relationship (e.g., touching or containing).

However, we instructed the annotators not to describe all objects and relationships, (1) to avoid unnecessarily verbose stories, and (2) to intentionally miss some information to enable more complex reasoning later. Therefore, annotators describe only a random subset of blocks, objects, and relationships.

To query more interesting phenomena, annotators were encouraged to write questions requiring detecting relations and reasoning over them using multiple spatial rules. A spatial rule can be one of the transitivity ($A \rightarrow B, B \rightarrow C \Rightarrow A \rightarrow C$), symmetry ($A \rightarrow B \Rightarrow B \rightarrow A$), converse ($(A, R, B) \Rightarrow (B, reverse(R), A)$), inclusion (*obj1 in A*), and exclusion (*obj1 not in B*) rules.

There are four types of questions (Q-TYPE). (1) *FR*: find relation between two objects. (2) *FB*: find the block that contains certain object(s). (3) *CO*: choose between two objects mentioned in the question that meet certain criteria. (4) *YN*: a yes/no question that tests if a claim on spatial relationship holds.

FB, FR, and CO questions are formulated as multiple-choice questions² and receive a list of candidate answers, and YN questions’ answer is choosing from Yes, No, or “DK” (Do not Know).

²CO can be considered as both single-choice and multiple-choices question.

The “DK” option is due to the open-world assumption of the stories, where if something is not described in the text, it is not considered as false (See Fig. 3.4).

Finally, annotators were able to create 1.1k QA pairs on spatial reasoning on the generated descriptions distributed among the aforementioned types. We intentionally keep this data on a relatively small scale due to two reasons. First, there has been some consensus in our community that modern systems, given their sufficiently large model capacities, can easily find shortcuts and overfit a dataset if provided with large training data [29, 115]. Second, collecting spatial reasoning QAs is very costly: The two annotators spent 45-60 mins on average to create a single story with 8-16 QA pairs. We estimate that SPARTQA-HUMAN cost about 100 hours in total. The expert performance on 100 examples of SPARTQA-HUMAN’s test set measured by their accuracy of answering the questions is 92% across four Q-TYPES on average, indicating its high quality.

3.2.1 Collect Spatial Role Labeling Annotations

Collecting spatial role labeling annotation is a complex and time-consuming task that involves identifying and labeling the spatial relationships between different objects or entities within a given context. This process requires a great deal of domain-specific knowledge and a keen understanding of linguistic and syntactic structures. In this task, annotators must possess a deep understanding of the spatial linguistic and semantic nuances of the language they are annotating. This requires extensive training and expertise, which can be challenging to find and cultivate.

Here, we asked two expert volunteers familiar with spatial role labeling tasks to annotate the story/questions of this dataset. They spent 270 hours annotating 2k sentences in SPARTQA-HUMAN using WebAnno framework³.

Obtaining consistent annotations can be complex, as different annotators may interpret the same context or scenario differently. This can lead to inconsistencies and inaccuracies in the annotations, significantly impacting the performance of any models trained on the data. To resolve this, another expert annotator checked the annotation and discarded the erroneous ones. As a result, half of this training data, and full dev and test data are annotated with SpRL tags (Check Tab 5.2).

³<https://webanno.github.io/webanno/>

Dataset	Train	Dev	Test
SPARTQA-HUMAN (story)	176	99	272
SPARTQA-HUMAN (question)	155	127	367

Table 3.1 Number of sentences in SPARTQA-HUMAN.

3.3 SPARTQA-AUTO

Since human annotations are costly, it is important to investigate ways to generate a synthetic evaluation and supervision signals for spatial reasoning. However, unlike conventional synthetic approaches (e.g., [88, 152, 158]) where synthetic data can be selected from large corpora by implementing specialized filtering rules, spatial reasoning does not often appear in existing corpora. Therefore, similar to SPARTQA-HUMAN, we take advantage of the ground truth of NLVR images, design CFGs to generate stories and use spatial reasoning rules to ask and answer spatial reasoning questions. This automatically generated data is called SPARTQA-AUTO, and below, we describe its generation process in detail.

3.3.1 Story Generation

Since NLVR comes with structured descriptions of the ground truth locations of those objects, we were able to choose random blocks and objects from each image programmatically. The benefit is two-fold. First, a random selection of blocks and objects allows us to create multiple stories for each image; second, this randomness also creates spatial reasoning opportunities with missing information.

Once we decide on a set of blocks and objects to be included, we determine their relationships: Those relationships between blocks are generated randomly; as for those between objects, we refer to the ground truth of these images to determine them.

Now we have a scene containing a set of blocks and objects and their associated relationships. To produce a story for this scene, we design CFGs to produce natural language sentences that describe those blocks/objects/relationships in various expressions (see Fig. 3.5 for two portions of our CFG describing relative and nested relations between objects).

Being grounded to visual scenes guarantees spatial coherency in a story, and using CFGs helps to have correct sentences (grammatically) and various expressions. We also design context-sensitive

The big black shape is above the medium triangle.

$S \rightarrow \langle \text{Article} \rangle \langle \text{Object} \rangle \text{ is } \langle \text{Relation} \rangle \langle \text{Article} \rangle \langle \text{Object} \rangle.$

Article \rightarrow *the* | *a*
Relation \rightarrow *above* | *left* | ...
Object \rightarrow $\langle \text{Size} \rangle^* \langle \text{Color} \rangle^* \langle \text{Shape} | \text{Ind_shape} \rangle$
Size \rightarrow *small* | *medium* | *big*
Color \rightarrow *yellow* | *blue* | *black*
Shape \rightarrow *square* | *triangle* | *circle*
Ind_shape \rightarrow *shape* | *object* | *thing*

(a) Part of the grammar describing relations between objects

The big black shape is above the object that is to the right of the medium triangle

$S \rightarrow \langle \text{Article} \rangle \langle \text{Object} \rangle \text{ is } \langle \text{Relation} \rangle \langle \text{Article} \rangle \langle \text{Object} \rangle.$

Object \rightarrow $\langle \text{Size} \rangle^* \langle \text{Color} \rangle^* \langle \text{Shape} | \text{Ind_shape} \rangle | \langle \text{Ind_shape} \rangle \text{ that is } \langle \text{Relation} \rangle \langle \text{Object} \rangle$

(b) Part of the grammar describing nested relationships.

Figure 3.5 Two parts of our designed CFG.

rules to limited options for each CFG’s variable based on the chosen entities (e.g., black circle) or what is described in the previous sentences (e.g., Block A has a circle. *The* circle is below a triangle.)

Table 3.2 shows some generated sentences in SPARTQA-AUTO with some specific features that challenge models to understand different forms of relation description in spatial language.

3.3.2 Question Generation

To generate questions based on a passage, there are rule-based systems [38, 58], neural networks [22], and their combinations. [19]. However, in our approach, during generating each story, the program stores the information about the entities and their relationships. Thus, without processing the raw text, which is error-prone, we generate questions by only looking at the stored data.

The question generation operates based on four primary functionalities, *Choose-objects*, *Describe-objects*, *Find-all-relations*, and *Find-similar-objects*. These modules are responsible to control the logical consistency, correctness, and the number of steps required for reasoning in each question.

Choose-objects randomly chooses up to three objects from the set of possible objects in a story under a set of constraints such as preventing selection of similar objects, or excluding objects with relations that are directly mentioned in the text.

Describe-Objects generates a mention phrase for an object using parts of its full name (presented in the story). The generated phrase is either pointing to a unique object or a group of objects such as

Examples	Features
Block A is above Block C and B.	Using conjunction to describe the relation between more than two blocks.
The small circle is above the yellow square and the big black shape.	Using conjunction to describe relationships between more than two objects.
The yellow square number one is to the right of and above the blue circle.	Using conjunction for more than one relation.
Block B has two medium yellow squares and two blue circles .	Describing a group of objects with the same properties. In the next sentences, they are mentioned by an assigned number. For example, the blue circle number two.
The blue circle is below the object which is to the right of the big square.	Using nested relations between objects in their description.
A small blue circle is near to the big circle. It is to the left of the medium yellow square.	Using coreferences for an entity described in the previous sentences.
There is a block named A. One small yellow square is touching the bottom edge of this block.	The verb matches the number of the subject.
What is the relation between black object and a big circle?	Using shape, object, and thing, which are a general description of an object. It could be the "black triangle" or the "black circle" mentioned in the story.

Table 3.2 Example of Particular features of SPARTQA-AUTO.

"the big circle," or "big circles." To describe a unique object, it chooses an attribute or a group of attributes that apply to a unique object among others in the story. To increase the steps of reasoning, the description may include the relationship of the object to other objects instead of using a direct unique description. For example, "the circle which is above the black triangle."

Find-all-relations completes the relationship graph between objects by applying a set of spatial rules such as transitivity, symmetry, converse, inclusion, and exclusion on top of the direct relations described in the story. As shown in Fig. 3.6, it does an exhaustive search over all combinations of the relations that link two objects to each other.

Find-similar-objects finds all the mentions matching a description from the question to objects in the story. For instance, for the question "is there any blue circle above the big blue triangle?", this module finds all the mentions in the story matching the description "a blue circle".

Similar to the SPARTQA-HUMAN, we provide four Q-TYPES FR, FB, CO, and YN. To generate

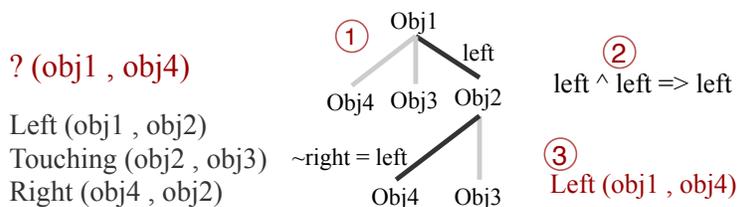


Figure 3.6 Find the implicit relation between *obj1* and *obj4* by *Transitivity* rule. (1) Find a set of objects that have a relation with *obj1*. Continue the same process on the new set until *obj4* is found. (2) Get the union of the intermediate relations between these two objects and it is the final answer.

FR questions, we choose two objects using *Choose-objects* module and question their relationships. The YN Q-TYPE is similar to FR, but the question specifies one relationship of interest chosen from all relation extracted by *Find-all-relations* module to be questioned about the objects. Since most of the time, Yes/No questions are simpler problems, we make this question type more complex by adding quantifiers (adding “all” and “any”). These quantifiers help to evaluate the models’ capability to aggregate relations between more than two objects in the story and do the reasoning over all find relations to find the final answer. In FB Q-TYPE, we mention an object by its indirect relation to another object using the nested relation in *Describe-objects* module and ask to find the blocks containing or not containing this object. Finally, the CO question selects an anchor object (*Choose-objects*) and specifies a relationship (using *Find-all-relations*) in the question. Two other objects are chosen as candidates to check whether the specified relationship holds between them and the anchor object. We tend to force the algorithm to choose objects as candidates that at least have one relationship to the anchor object.

3.3.2.1 Question Templates and Statistics Information

Table 3.3 shows the templates used to create questions in SPARTQA-AUTO. The “<object>” is a variable replaced by objects from the story (using *Choose-objects* and *Describe-objects* modules), and the “<relation>” variable can be replaced by the chosen relations between objects (using *Find-all-relations* module).

The articles and the indefinite pronouns in each template play an essential role in understanding the question’s objective. For example, “Are all blue circles near to a triangle?” is different from “Are there any blue circles near to a triangle?”, and “Are there any blue circles near to all triangles?”.

Q-Type	Q-Templates	Candidate answer
FR	what is the relation between <object>and <object>?	Left, Right, Below, Above, Touching, Far from, Near to
CO	What is <relation >the <object>? an <object1>or an <object2>? Which object is <relation >an <object>? the <object1>or the <object2>?	Object1, object2, Both, None
YN	Is (the a) <object1><relation>(the a) <object2>? Is there any <object1>s <relation>all <object2>s?	Yes, No, Don't Know
FB	Which block has an <object>? Which block doesn't have an <object>?	Name of blocks, None

Table 3.3 Questions and answers templates.

Therefore, we check the uniqueness of the object definition, using “a” or “the” in proper places and randomly place the terms “any” or “all” in the YN questions to generate different questions.

Answer generation We compute all direct and indirect relationships between objects using *Find-all-relations* function and based on the Q-TYPES generate the final answer.

For instance, in YN Q-TYPE if the asked relation exists in the found relations, the answer is "Yes", if the inverse relation exists it must be "No", and otherwise, it is "DK"⁴.

3.3.3 Corpus Statistics

We generate the train, dev, and test set splits based on the same splits of the images in the NLVR dataset. On average, each story contains 9 sentences (Min:3, Max: 22) and 118 tokens (Min: 66, Max: 274). Also, the average tokens of each question (on all Q-TYPE) is 23 (Min:6, Max: 57).

Table 3.4 shows the total number of each question type in SPARTQA-AUTO and Table 3.5 shows the percentage of each correct label in train and test sets. Table 3.5 shows the percentage of correct labels in train and test sets. In multi-choice Q-TYPES, more than one label can be true.

3.3.4 Extra Annotations

Alongside the main SPARTQA-AUTO’s stories and questions, we provided some extra annotation to help the models understand the spatial language better.

⁴The SPARTQA-AUTO generation code and the file of dataset are available at https://github.com/HLR/SpertQA_generation

Sets	FB	FR	YN	CO	Total
SPARTQA-HUMAN:					
Test	104	105	194	107	510
Train	154	149	162	151	616
SPARTQA-AUTO:					
Seen Test	3872	3712	3896	3594	15074
Unseen Test	3872	3721	3896	3598	15087
Dev	3842	3742	3860	3579	15023
Train	23654	23302	23968	22794	93673

Table 3.4 Number of questions per Q-TYPE.

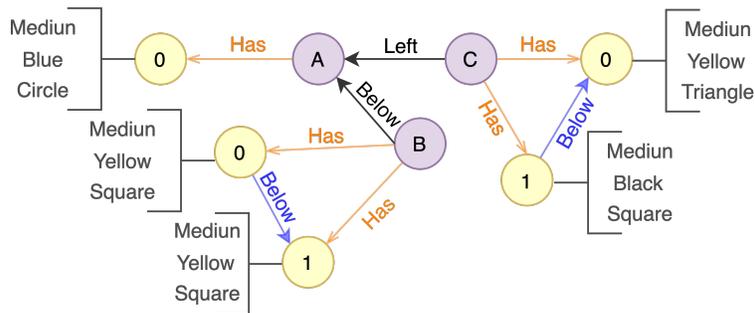


Figure 3.7 Scene-graph.

3.3.4.1 Detailed Annotation and Scene-Graphs

Providing in-depth human annotations is quite expensive and time-consuming. In SPARTQA-AUTO, we generated a fine-grained scene-graph based on the story. This scene graph contains the blocks' description, their relations, and the objects' attributes alongside their direct relations. The scene graphs can be used for the models to understand all spatial relations directly mentioned in the textual context. Figure 3.7 shows an example of this scene-graph. The scene graph can provide strong supervision for question answering challenges and can be used to evaluate models based on their steps of reasoning and decisions.

3.3.4.2 SpRL Annotation

We also provided spatial annotations for each sentence and question, based on the Spatial Role Labeling annotation scheme [54](Fig. 3.8). This annotation is generated by hand-crafted rules during the main data generation. SpRL is used for recognizing spatial expressions and arguments in a sentence. This annotation is useful for applications that need to detect and reason about spatial

Q-TYPE	Candidate Answers	train	test
FR (Multiple Choices)	Left	20.7	17.9
	Right	21.4	16.7
	Above	26.9	25.4
	Below	37.2	42.9
	Near to	5.8	2.9
	Far from	1.3	0.56
	Touching	0.57	0.27
	DK	0.52	0.32
FB (multiple Choices)	A	49.8	49.4
	B	50.1	50
	C	35.1	62
	[]	7.1	90.5
CO (Single choice)	Object1	25.4	26
	Object2	25.3	24.9
	Both	44.3	43.9
	None	4.9	5.0
YN (Single choice)	Yes	53.3	50.5
	No	18.7	23.6
	DK	27.8	25.9

Table 3.5 The percentage of each correct label in all samples. The candidate answers for the FB Q-TYPE can be varied based on its story. CO can be considered a multiple-choice or single-choice question. E.g., in "which object is above the triangle? the blue circle or the black circle?" you can consider two labels with boolean classification on each "blue circle" and "black circle" or consider it as a four-label classification: "blue circle," "black circle," "both of them," and "None of them." * **DK, None, []**, all mean none of the actual labels are correct.

expressions and arguments.

3.4 Spatial Question Answering Models

This section describes the model architectures on different Q-TYPES: FR, YN, FB, and CO. All Q-TYPES can be cast into a sequence classification task, and the three transformer-based LMs tested in this paper, BERT [18], ALBERT [59], and XLNet [147], can all handle this type of tasks by classifying the representation of [CLS], a special token prepended to each target sequence (see Figure 3.9). Depending on the Q-TYPE, the input sequence and how we do inference may be different.

FR and YN both have a predefined label set as candidate answers, and their input sequences are both the concatenation of a story and a question. While the answer to a YN question is a single label chosen from *Yes*, *No*, and *DK*, FR questions can have multiple correct answers. Therefore, we treat

```

sentence: "Medium blue square number one is touching the bottom edge of this block."
▼ spatial_description: [ ] 1 item
  ▼ 0:
    ▼ trajectory:
      phrase: "medium blue square number one"
      head: "square"
    ▼ properties:
      color: "blue"
      size: "medium"
      name: "number one"
      number: ""
      spatial_property: ""
    ▼ SOT:
      start: 167
      end: 195
    ▼ landmark:
      phrase: "the bottom edge of this block"
      head: "block"
      ► properties:
        spatial_property: "the bottom edge"
      ► SOT:
    ▼ spatial_indicator:
      phrase: "touching"
      spatial_value: "TPP"
      g_type: "Region"
      s_type: "RCC8"
      polarity: false
      FoR: "Relative"
      ► SOT:

```

Figure 3.8 SpRL annotation for an example sentence from SPARTQA.

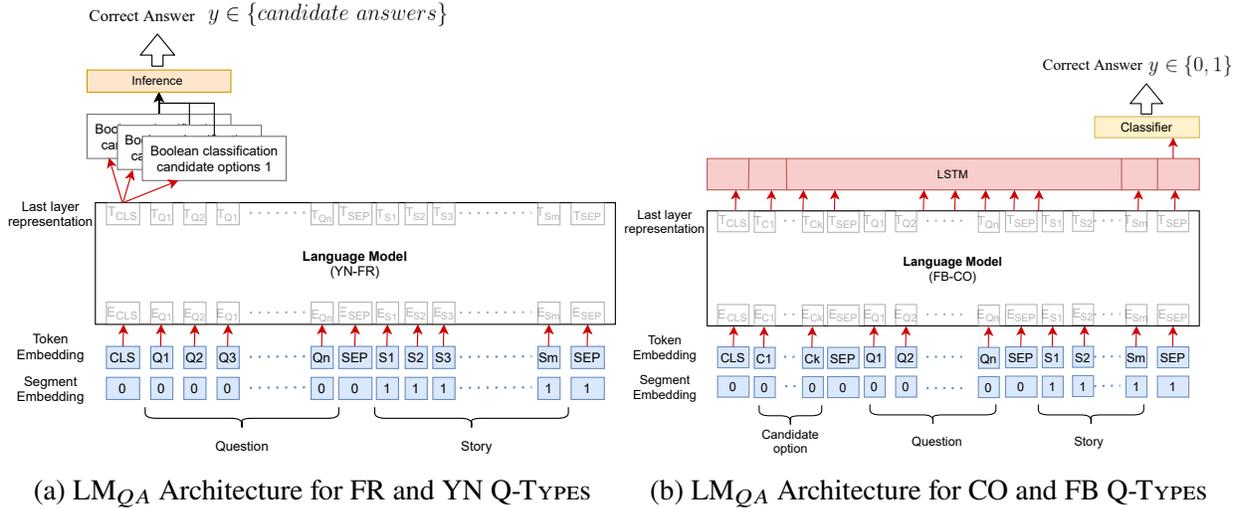


Figure 3.9 LM_{QA} for Spatial Reasoning over Text.

each candidate answer to FR as an independent binary classification problem, and take the union as the final answer. As for YN, we choose the label with the highest confidence (Fig 3.9a).

As the candidate answers to FB and CO are not fixed and depend on each story and its question the input sequences to these Q-TYPES are concatenated with each candidate answer. Since the defined YN and FR model has moderately less accurate results on FB and CO Q-TYPES, we add a LSTM [40]

layer to improve it. Hence, to find the final answer, we run the model with each candidate answer and then apply an LSTM layer on top of all token representations. Then, we use the last vector of the LSTM outputs for classification (Fig 3.9b). The final answers are selected based on Eq. (3.1).

$$\begin{aligned}
 x_i &= [s, c_i, q] \\
 \vec{T}_i &= [\vec{t}_1^i, \dots, \vec{t}_{m_i}^i] = LM(x_i) \\
 [\vec{h}_1^i, \dots, \vec{h}_{m_i}^i] &= LSTM(\vec{T}_i) \\
 \vec{y}_i &= [y_i^0, y_i^1] = \text{Softmax}(\vec{h}_{m_i}^{i^T} W) \\
 \text{Answer} &= \{c_i \mid \arg \max_j (y_i^j) = 1\}
 \end{aligned} \tag{3.1}$$

where s is the story, c_i is the candidate answer, q is the question, $[]$ indicates the concatenation of the listed vectors, and m_i is tokens' number in x_i . The parameter vector, W , is shared for all candidates.

3.4.1 Training and Inference

We train the models based on the summation of the cross-entropy losses of all binary classifiers in the architecture. For FR and YN Q-TYPES, there are multiple classifiers, while there is only one classifier used for CO and FB Q-TYPES.

We remove inconsistent answers in post-processing for FR and YN Q-TYPES during inference phase. For instance on FR, *left* and *right* relations between two objects cannot be valid at the same time. For YN, as there is only one valid answer amongst the three candidates, we select the candidate with the maximal predicted probability of being the true answer.

3.5 Experiments and Results

All experiments use the models proposed in Sec. 3.4. We use AdamW [72] with 2×10^{-6} learning rate and Focal Loss [69] with $\gamma = 2$ for training all the models.⁵ To evaluate our models, we use the Accuracy metric, which computes the correct prediction ratio to all questions.

⁵All codes are available at <https://github.com/HLR/SpartQA-baselines>

To assess the quality of the proposed datasets, we measure human performance accuracy on a sample of these datasets. The evaluation involved 100 randomly selected questions from the SPARTQA-HUMAN test set. The participants were graduate students who received training with a few examples from the dataset before answering the final questions. The high accuracy rates observed suggest that the questions are understandable and suitable for evaluation. Besides, upon examining respondents’ incorrect answers, we found that errors typically occurred when the questions required a high number of reasoning steps, leading to confusion in linking them together.

3.5.1 SPARTQA-HUMAN Evaluation

Table 3.6 shows performance on SPARTQA-HUMAN in a low-resource setting, where 0.6k QA pairs from SPARTQA-HUMAN are used for fine-tuning these LMs and 0.5k for testing (see Table 3.4 for information on this split).⁶ During our annotation, we found that the description of “near to ” and “far from” varies largely between annotators. Therefore, we ignore these two relations from FR Q-TYPE in our evaluations.

#	Model	FB	FR	CO	YN	Avg
1	Majority	28.84	24.52	40.18	53.60	36.64
2	BERT	16.34	20	26.16	45.36	30.17
3	Human	91.66	95.23	91.66	90.69	92.31

Table 3.6 Result of three baselines on SPARTQA-HUMAN. All systems are fine-tuned on the training data of SPARTQA-HUMAN, Avg: The micro-average on all four Q-TYPES.

As it is shown in Table 3.6, BERT language model, which outperforms many complex deep neural networks in other benchmarks [11], cannot perform well on SPARTQA-HUMAN. The results of this model even are lower than the Majority baseline which predicts the label that is most common in each set on each question type. This low performance indicates the complexity of the spatial reasoning (spatial question answering) task and the poor capability of powerful language models to capture that. We also can see from Table 3.6 that BERT performance falls behind human performance by a large margin.

⁶Note this low-resource setting can also be viewed as a spatial reasoning probe to these LMs [128].

#	Models	FB		FR		CO		YN	
		Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
1	Majority	48.70	48.70	40.81	40.81	20.59	20.38	49.94	49.91
2	BERT	87.13	69.38	85.68	73.71	71.44	61.09	78.29	76.81
3	ALBERT	97.66	83.53	91.61	83.70	95.20	84.55	79.38	75.05
4	XLNet	98.00	84.85	94.60	91.63	97.11	90.88	79.91	78.54
5	Human	85		90		94.44		90	

Table 3.7 Spatial reasoning is challenging. We further pretrain three transformer-based LMs, BERT, ALBERT, and XLNet, on SPARTQA-AUTO, and test their accuracy in three ways: *Seen* and *Unseen* are both from SPARTQA-AUTO, where *Unseen* has applied minor modifications to its vocabulary; to get those *Human* columns, all models are fine-tuned on SPARTQA-HUMAN’s training data. Human performance on *Seen* and *Unseen* is the same since the changes applied to *Unseen* does not affect human reasoning.

3.5.2 SPARTQA Analysis

In addition to BERT, we continue to test another two LMs, ALBERT and XLNet (Table 3.7). We further pretrain these LMs on SPARTQA-AUTO, and test them on SPARTQA-HUMAN and two held-out test sets of SPARTQA-AUTO, *Seen* and *Unseen*. Note that when a system is tested against SPARTQA-HUMAN, it is fine-tuned on SPARTQA-HUMAN’s training data following its further pretraining on SPARTQA-AUTO.

We propose an unseen test set alongside the seen test of SPARTQA-AUTO to check whether a model is using shortcuts in the language surface by describing objects and relations with new vocabularies in the samples. We use the unseen set to test to what extent the baseline models use shortcuts in the language surface. This set applies minor modifications randomly on a number of stories and questions to change the names of shapes, colors, sizes, and relationships in the vocabulary of the stories, which do not influence the reasoning steps (describing relationships using different language expressions). The modification choices are described in Table 3.8.

All models perform worst in YN across all Q-TYPES, which suggests that YN presents more complex phenomena, probably due to the additional quantifiers in the questions. XLNet performs the best on all Q-TYPES except its accuracy on SPARTQA-HUMAN’s YN section. However, the drops in *Unseen* and *human* suggest overfitting on the training vocabulary. The low accuracies on the human test set from all models show that solving this benchmark is still a challenging problem and requires more sophisticated methods like considering spatial roles and relations extraction [54, 13, 105] to

Type	Original Set	Unseen Set
Shapes	Square, Circle, Triangle	Rectangle, Oval, Diamond
Relations	Left, Right, Above, Below	Left side, Right side, Top, Under
Colors	Yellow, Black, Below	Green, Red, White
Size	Small, Medium, Big	Little, Midsize, Large

Table 3.8 Modifications on the unseen set.

understand stories and questions better.

3.5.3 Consistency Evaluation

To evaluate the reliability of the models, we also provide two extra consistency and contrast test sets.

Consistency set is made by changing a part of the question in a way that seeks for the same information [43, 124]. Given a pivot question and answer of a specific consistency set, answering other questions in the set does not need extra reasoning over the story. For instance, for the question, “What is the relation between the blue circle and the big shape? Left,” we create a similar question in the form of “What is the relation between the big shape and the blue circle? Right”. Answering these questions around a pivot question is possible for humans without the need for extra reasoning over the story and based on the main questions’ answers. Hence, the evaluation on this set shows that models understand the real underlying semantics rather than overfit on the structure of questions.

Contrast set is made by minimal modification in a question to change its answer [29]. For contrast sets, there is a need to go back to the story to find the new answer for the question’s minor variations. The consistency and contrast sets are evaluated only on the correctly predicted questions to check if the actual understanding and reasoning occur. This ensures the reliability of the models. For instance, in the question “Is the blue circle below the black triangle? Yes,” we create a contrast question “Is the blue circle below all triangles? No” by changing “the black triangle” to “all triangles”. The evaluation of this set shows the robustness of the model and its sensitivity to

semantic changes when there are minor changes in the language surface.

Models	FB	FR		CO		YN	
	Consistency	Consistency	Contrast	Consistency	Contrast	Consistency	Contrast
BERT	69.44	76.13	42.47	16.99	15.58	48.07	71.41
AlBERT	84.77	82.42	41.69	58.42	62.51	48.78	69.19
XLNet	85.2	88.56	50	71.10	72.31	51.08	69.18

Table 3.9 Evaluation of consistency and semantic sensitivity of models in Table 3.7. All the results are on the correctly predicted questions of *Seen* test set of SPARTQA-AUTO.

Based on the papers [43, 124], consistency, and contrast set should be generated manually to control the semantic change. In our case, we are probing the spatial language understanding of models, so we must change parts of the questions that affect spatial understanding, which can be implemented automatically by some static rules.

Table 3.9 shows the result of this evaluation on four Q-TYPES of SPARTQA-AUTO, where we can see, for another time, that the high scores on the *Seen* test set are likely due to overfitting on training data rather than correctly detecting spatial terms and reasoning over them.

All of the mentioned experiments clearly indicate that current SOTA and powerful language models cannot perform spatial reasoning on the provided benchmarks. Also, experiments showed that the high result of LMs on SPARTQA-AUTO are due to the overfitting of the models on fixed pattern and cannot be repeated after minor changes.

3.6 Related Research

Question answering is a useful format to evaluate machines’ capability of reading comprehension [30] and many recent works have been implementing this strategy to test machines’ understanding of linguistic formalisms: [37, 85, 64, 45, 97, 21]. An important advantage of QA is using natural language to annotate natural language, thus having the flexibility to get annotations on complex phenomena such as *spatial reasoning*. However, spatial reasoning phenomena have been covered minimally in the existing works.

To the best of our knowledge, Task 17 of the bAbI project [143] is the only QA dataset focused on textual spatial reasoning. However, bAbI is synthetic and does not reflect the complexity of the spatial reasoning in natural language. Solving Task 17 of bAbI typically does not require

sophisticated reasoning, which is an important capability emphasized by more recent works (e.g., [23, 50, 148, 14, 97]).

Spatial reasoning is arguably more prominent in multi-modal QA benchmarks, e.g., NLVR [123], VQA [4], GQA [43], CLEVR [46]. However, those spatial reasoning phenomena are mostly expressed naturally through images, while this paper focuses on studying spatial reasoning on natural language. Some other works on visual-spatial reasoning are based on geographical information inside maps and diagrams [42] and navigational instructions [9, 3].

As another approach to evaluate spatial reasoning capabilities of models, a dataset proposed in [32] generates a synthetic training set of spatial sentences and evaluates the models' ability to generate spatial facts and sentences containing composition and decomposition of relations on grounded objects.

3.7 Summary

Spatial reasoning is an important problem in natural language understanding. We propose the first human-created QA benchmark on spatial reasoning, and experiments show that SOTA pretrained language models (LM) do not have the capability to solve this task given limited training data, while humans can solve those spatial reasoning questions reliably [90].

We also provide a synthetic benchmark using hand-crafted grammar and spatial reasoning rules to automatically generate a large corpus of spatial descriptions and corresponding question-answer annotations. Even though LMs can solve the main test set of this benchmark, by providing various evaluation test sets we showed that these high results are due to the overfitting, and the models do not learn spatial concepts and how to do spatial reasoning over them.

CHAPTER 4

TRANSFER LEARNING FOR SPATIAL QUESTION ANSWERING

4.1 Introduction

The previous chapter showed how current SOTA language models cannot resolve spatial question answering tasks. As fine-tuning LMs has become a common baseline approach to knowledge transfer from a source dataset to a target task, including but not limited to [101, 158, 36], in this chapter, we study this method to increase the capability of spatial reasoning of modern LMs. This fine-tuning process is also known as *further pretraining*, to distinguish with the fine-tuning process on one’s target task. It is an open problem to find out better transfer learning techniques than simple further pretraining, as suggested in [35, 51], which is beyond the scope of this work.

One of the key requirements in this method is access to large training data. As we discussed before, SPARTQA-AUTO is the only available large benchmark on SQA. We use this dataset as a synthetic source of supervision to further pretrain the BERT language model and then fine-tune and evaluate the model on SPARTQA-HUMAN with a similar domain and some other target benchmarks. Experiments show that, by further pretraining on SPARTQA-AUTO, we improve LMs’ performance on SPARTQA-HUMAN by a large margin. The spatially-improved LMs also show stronger performance on two external QA datasets, bAbI and boolQ [11]: BERT further pretrained on SPARTQA-AUTO only requires half of the training data to achieve 99% accuracy on bAbI as compared to the original BERT; on boolQ’s development set, this model shows better performance than BERT, with 2.3% relative error reduction.¹

4.1.1 Improve the Coverage of Spatial Relations

The current *synthetic datasets* [90, 143, 118] often focus on specific types of relations with a small coverage of spatial semantics needed for spatial language understanding in various domains. Figure 4.1 indicates the coverage of sixteen spatial relation types (in Table 2.1) collected from existing resources [108, 144, 110]. The *human-generated datasets* [90, 55], despite helping study the

¹To the best of our knowledge, the test set or leaderboard of boolQ has not been released yet.

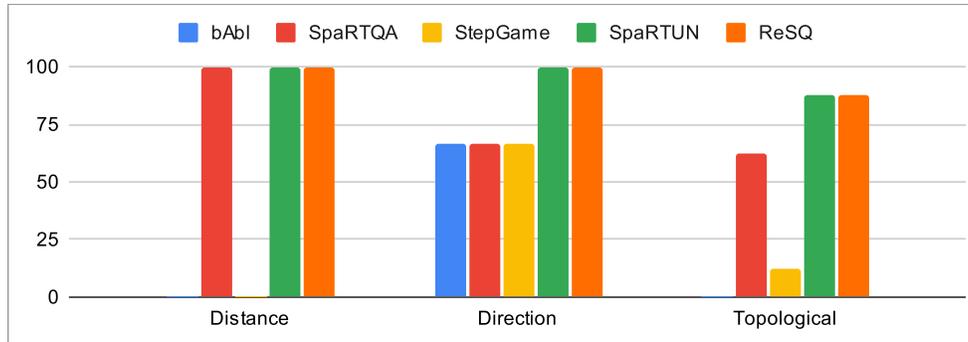


Figure 4.1 The comparative coverage of relation types based on Table 2.1 for SQA datasets.

problem as evaluation benchmarks, are less helpful for training models that can reliably understand spatial language due to their small size [90].

In this work, we build a new synthetic dataset on SQA, called SPARTUN² (Figure 4.2) to provide a source of supervision with broad coverage of spatial relation types and expressions³.

To generate SPARTUN, we follow the idea of SPARTQA benchmark and generate scene graphs from a set of images. The edges in this graph yield a set of triplets such as ABOVE(blue circle, red triangle), which are used to generate a scene description (i.e., a story).

In SPARTUN, we map the spatial relation types in triplets (e.g., ABOVE) to a variety of spatial language expressions (e.g., over, north, above) to enable the transfer learning for various data domains⁴. We also build a logical spatial reasoner to compute all possible direct and indirect spatial relations between graph nodes. Then, the questions of this dataset are selected from the indirect relations.

To evaluate the effectiveness of SPARTUN in transfer learning, we created another dataset named RESEQ⁵ (Figure 6.1). This dataset is built on mSPRL [55] corpus while we added human-generated spatial questions and answers to its real image descriptions. This dataset comparatively reflects more realistic challenges and complexities of the SQA problem.

We analyze the impact of SPARTUN as source of extra supervision on several SQA benchmarks.

²**Spatial Reasoning and role labeling for Text UNderstanding**

³We only consider explicit spatial semantics and the Metaphoric usages and implicit meaning are not covered in this work.

⁴The full list of spatial expressions used in this dataset and the dataset generation code are provided in <https://github.com/HLR/SpaRTUN>.

⁵**Real-world Spatial Questions**

Our results show that further pretraining models with SPARTUN for SQA task improves the result of previous models on RESEQ, StepGame, and SPARTQA-HUMAN benchmarks. Furthermore, studying the broad coverage of spatial relation expressions of SPARTUN in realistic domains demonstrates that this feature is a key factor for transfer learning.

The contributions of this paper can be summarized as:

(1) SPARTQA-AUTO proves to be a rich source of spatial knowledge that improved the performance of LMs on SPARTQA-HUMAN as well as on different data domains such as bAbI and boolQ.

(2) We build a new synthetic dataset to serve as a source of supervision and transfer learning for spatial language understanding tasks with broad coverage of spatial relation types and expressions (which is easily extendable).

(3) We provide a human-generated dataset to evaluate the performance of transfer learning on real-world spatial question answering.

(4) We evaluate the transferability of the models pretrained with SPARTUN on multiple SQA and SpRL benchmarks and show significant improvements in SOTA results.

4.2 SPARTUN: A Synthetic Data with Broader Relation Coverage

The *synthetic datasets* on SQA often focus on specific types of relations with a small coverage of spatial semantics needed for spatial language understanding in various domains. Figure 4.1 indicates the coverage of sixteen spatial relation types (in Table 2.1) collected from existing resources [108, 144, 110]. The *human-generated datasets*, despite helping study the problem as evaluation benchmarks, are less helpful for training models that can reliably understand spatial language due to their small size.

In this section, we build a new synthetic dataset on SQA, called SPARTUN⁶ (Figure 4.2) to provide a source of supervision with broad coverage of spatial relation types and expressions⁷.

To provide a source of supervision for spatial language understanding tasks, we generate a synthetic dataset with SQA format that contains SpRL annotation of sentences. We build this dataset

⁶Spatial Reasoning and role labeling for Text UNderstanding

⁷We only consider explicit spatial semantics and the Metaphoric usages and implicit meaning are not covered in this work.

Three boxes called one, two and three exist in an image. Box one contains a big yellow melon and a small orange watermelon. **Box two has a small yellow apple.** A small orange apple is inside and touching this box. Box one is in box three. **Box two** is to the **south** of, **far from** and to the **west** of **box three**. A **small yellow watermelon** is **inside** **box three**.

Q: Is **the yellow apple** to the **west** of the **yellow watermelon**? **Yes**

Q: Where is **box two** relative to the **yellow watermelon**? **Left, Below, Far**

Figure 4.2 SPARTUN - A synthetic large dataset provided as a source of supervision.

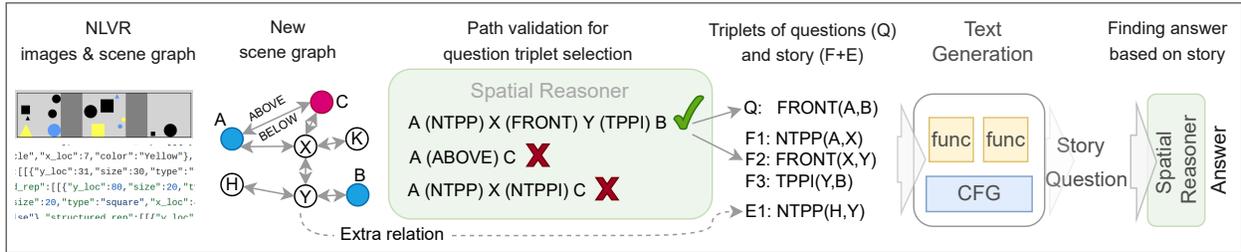


Figure 4.3 The data construction flow of SPARTUN. First, we generate scene graphs from NLVR images. Then a spatial reasoner validates each path between each pair of entities in this graph. All facts (F) in the selected *path* and some extra facts (E) from the scene graph are selected as story triplets, and the start and end nodes of the *path* are selected as question triplets. Finally, we pass all triplets to a text generation module and compute the final answer. We ignore paths with length one (e.g., $A(ABOVE)C$) and only keep questions that need multi-hop reasoning.

by expanding SPARTQA in multiple aspects. The following additional features are considered in creating SPARTUN:

F1) A broad coverage of various types of spatial relations and including rules of reasoning over their combinations (e.g. $NTPP(a, b), LEFT(b, c) \rightarrow LEFT(a, c)$) in various domains.

F2) A broad coverage of spatial language expressions and utterances used in various domains.

F3) Including extra annotations such as the supporting facts and number of reasoning steps for SQA to be used in complex modeling.

In the rest of this section, we describe the details of creating SPARTUN and the way we support the above mentioned features. Figure 4.3 depicts SPARTUN data construction flow.

Spatial Relation Computation. Following SPARTQA-AUTO, we use the NLVR scene graphs [123] and compute relations between objects in each block based on their given coordinates. NLVR is limited to 2D relation types⁸, therefore to add more dimensions (FRONT and BEHIND), we randomly change the LEFT and RIGHT to BEHIND and FRONT in a subset of examples. Moreover, there are no relations between blocks in NLVR descriptions.

To expand the types of relations, we extend this limitation and randomly assign relations⁹ to the blocks while ensuring the spatial constraints are not violated. Then, we create a new scene graph with computed spatial relations. The nodes in this graph represent the entities (objects or blocks), and the directed edges are the spatial relations.

Question Selection. There are several paths between each pair of entities in the generated scene graph. We call a path valid if at least one relation can be inferred between its start and end nodes can be inferred. For example, in Figure 4.3, $NTPP(A, X)$, $FRONT(X, Y)$, $TPPI(Y, B)$ is valid since it results in $FRONT(A, B)$ while $NTPP(A, X)$, $NTPPI(X, C)$ is not a valid path –there is no rules of reasoning that can be applied to infer new relations.

To verify the validity of each path, we pass its edges, represented as triplets in the predicate-arguments form to a logical spatial reasoner (implemented in Prolog) and query all possible relations between the pair. The number of triplets in each path represents the number of reasoning steps for inferring the relation.

We generate the question triplets from the paths with the most steps of reasoning (edges). This question will ask about the spatial relationship between the head and tail entity of the selected path. The triplets in this path are used to generate the story and are annotated as supporting facts. Additionally, the story will include additional information (extra triplets) unnecessary for answering the question to increase the complexity of the task.

Spatial Reasoner. We implement several rules (in the form of Horn clauses shown in Table 6.2) in Prolog, which express the logic between the relation types (described in Table 2.1) in various formalisms and model the logical spatial reasoning computation. Given the facts and queries in

⁸The relations types included in NLVR are: DC, EC, *PP relations, LEFT, RIGHT, BELOW, and ABOVE.

⁹All relation in Table 2.1 except EQ

Prolog format, the spatial reasoner can carry out the reasoning process and provide an answer to any given query. The reasoner matches variables in the program with concrete values and a backtracking search to explore different possibilities for each rule until a solution is found. Compared to previous tools [144], we are the first to include the spatial, logical computation between multiple formalisms. This reasoner validates the question/queries based on the given facts. For instance, by using the Combination rule in Table 6.2 over the set of facts $\{NTPP(A, X), FRONT(X, Y), TPPI(Y, B)\}$, the reasoner returns *True* for the query $FRONT(A, B)$ and *False* for $FRONT(B, A)$ or $BEHIND(A, B)$.

We consider the logic rules mentioned in Table 6.2 and in the form of the Horn clauses. we collect the different combinations of spatial relations mentioned in Table 2.1 and implement the logic-based spatial reasoner. Figure 4.4a shows an example of some parts of our code on *LEFT* relation. In Figure 4.4b, on the left, some facts are given, and the query “*ntppi(room, X)*” asks about all objects that existed in the room. Below each query, there are all possible predictions for them.

Text generation. The scene description is generated from the selected story triplets in question selection phase and using a publicly available context-free grammar (CFG) provided in SPARTQA-AUTO. However, we increase the variety of spatial expressions by using a vocabulary of various entity properties and relation expressions (e.g., above, over, or north for ABOVE relation type) taken from existing resources [27, 80, 71, 122, 39] We map the relation types and the entity properties to the lexical forms in our collected vocabulary (check Figures 4.5 and 4.6).

For the question text, we generate the entity description and relation expression for each question triplet. The entity description is generated based on a subset of its properties in the story. For instance, an expression such as “a black object” can be generated to refer to both “a big black circle” and “a black rectangle”. We generate two question types, YN (Yes/No) questions that ask whether a specific relation exists between two entities, and FR (Find Relations) questions that ask about all possible relations between them. To make YN questions more complex, we add quantifiers (“all” and “any”) to the entities’ descriptions.

Our text generation method can flexibly use an extended vocabulary to provide a richer corpus

```

left_2(X,Y) :-
    left_2(X,Y,[ ]).

left_2(X,Y,_) :-
    left_1(X,Y).

left_2(X,Z,Visited) :-
    left_1(X,Y),
    Y \= X,
    \+ member(Y, Visited),
    left_2(Y,Z, [Y|Visited]),
    Y \= Z,
    X \= Z.

left_2_check(X,Y) :- left_2(X,Y),

right_2(X,Z) :- left_2(Z,X).

left_3(X,Z) :-
    left_2_and_eq(X,Z);

left_3__(X,Z) :-
    has_pp_rel(X,Y),
    X \= Y,
    left_2_and_eq(Y,H),
    Y \= H,
    has_ppi_rel(H,Z),
    H \= Z;

left_2_and_eq(X,Z) :-
    left_2(X,Z);

eq_2(X,Y),
left_2_check(Y,Z);

eq_2(Y,Z),
left_2_check(X,Y).

left_2_and_eq(X,H),
X \= H,
has_ppi_rel(H,Z),
H \= Z.

```

(a) Example of implemented rule clauses in Prolog.

```

%Facts:
left(box1, box2).
above(box1, box2).
below(box3, box2).

ntpp(apple, box1).
ntpp(pear, box3).
ntppi(box2, grapes).
ntppi(room, box1).
ntppi(room, box2).
dc(box2, box1).
eq(apple1, apple).
eq(apple2, apple1).

%Query:
?- ntpi(room, X).
X = box1 ;
X = box2 ;
X = apple ;
X = grapes ;
X = apple1 ;
X = apple2 ;
X = apple1 ;
X = box1 ;
false.

?- left(X, grapes).
X = apple ;
X = apple1 ;
X = apple2 ;
X = apple1 ;
X = box1 ;
false.

```

(b) Example of Facts, Query, and answer of implemented model

Figure 4.4 Logic-based spatial reasoner.

to supervise new target tasks when required.

Finding Answers. We search all entities in the story based on the entity descriptions (e.g., all circles, a black object) in each question and use the spatial reasoner to find the final answer.

SpRL Annotations. Along with generating the sentences for the story and questions, we automatically annotate the described spatial configurations with spatial roles and relations (trajectory, landmark, spatial indicator, spatial type, triplet, entity ids). These annotations are based on a

Formalism	Type		Cardinals	Clocks
Directional	LEFT	"to the left of", "on the left side of", "to the left-hand side of"	"west of", "to the west of"	"at 9:00 position relative to", "at 9:00 position regarding to", "at 9 o'clock position regarding to"
	RIGHT	"to the right of", "on the right side of", "to the right-hand side of"	"east of", "to the east of"	"at 3:00 position relative to", "at 3:00 position regarding to", "at 3 o'clock position regarding to"
	BELOW	"above", "over"	"north of", "to the north of"	"at 12:00 position relative to", "at 12:00 position regarding to", "at 12 o'clock position regarding to"
	ABOVE	"below", "under"	"south of", "to the south of"	"at 6:00 position relative to", "at 6:00 position regarding to", "at 6 o'clock position regarding to"
	BEHIND FRONT	"behind", "on the back of" "in front of", "after"		
Distances	FAR	"far from", "farther from", "away from"		
	NEAR	"near to", "close to"		
Topological	DC	disconnected from		
	EC	"touch[es]"		
	PO	"overlap[s]"		
	EQ	-		
	TPP	"covered by", "inside and touching"		
	TPPI	"cover[s]"		
	NTPP	"in", "inside", "within"		
	NTPPI	"ha[s/ve]", "contain[s]"		

Figure 4.5 List of relation expression supported in SPARTUN. This can easily be extended based on the target task.

Properties	
block	Block, box
blocks	Blocks, boxes
object_general_name	thing, object, shape, fruit
objects_general_name	things, objects, shapes, fruits
block_name	AAA, BBB, CCC, DDD, EEE, JJJ, HHH, JJJ, LLL, KKK, one, two, three.
color	yellow, black, blue, green, red, orange, grey, white, purple
size	small, big, medium, midsize, large, tiny, little
type	circle, oval, square, rectangle, dimond, star, triangle, hexagon, pentagon, watermelon, apple, melon.
types	circles, ovals, squares, rectangles, dimonds, stars, triangles, hexagons, pentagons, watermelons, apples, melons.

Figure 4.6 List of entities properties supported in SPARTUN.

previously proposed annotation scheme of SpRL and provide free annotations for the SpRL task.

4.2.1 Statistic Information:

Each example in SPARTUN contains a story that describes the spatial relation between entities and some questions which ask about indirect relations between entities. On average, each story contains eight sentences and 91 tokens, which describe ten relations on average.

To generate SPARTUN, we use 6.6k NLVR scene graphs for training and 1k for each dev and test set. We follow SPARTQA for dataset split. Thus, we collect 20k training, 3k dev, and 3k test examples for each FR and YN question (see Table 4.1)¹⁰. On average, each story of SPARTUN contains eight sentences and 91 tokens that describe, on average ten relations between different mentions of entities.

54% of the YN questions have "Yes" answers and the rest (46%) have "No." In FR questions which can have multiple answers, the label are true as below: { left : 10%, right:10%, above: 27%, below: 26%, behind: 19%, front: 10%, near: 2%, far: 15%, dc: 26%, ec: 7%, po: 0.2%, tpp: 2%, ntp: 10%, tppi: 3%, and ntpi: 8% }.

4.2.2 Version Updates

We introduce an updated version of the dataset wherein we have eliminated elements contributing to coreference and pronoun complexity, as well as quantifiers in the Yes/No questions. This modification has been made to ensure that the evaluation of the models is focused solely on their capacity to execute multi-hop spatial reasoning tasks. To facilitate this, each variable within the dataset is assigned a unique name/identifier, which is then consistently utilized across all corresponding sentences and questions. These changes minimize ambiguity, thereby allowing for a more accurate assessment of the models' spatial reasoning capabilities.

4.3 Spatial Question Answering in Real-World Domain

To evaluate the effectiveness of SPARTUN in transfer learning, we created another dataset named RESQ¹¹ (Figure 6.1). This dataset is built on mSPRL [55] corpus while we added human-generated

¹⁰All data are provided in the English language.: The corpus is in English.

¹¹Real-world Spatial Questions

spatial questions and answers to its real image descriptions. This dataset comparatively reflects more realistic challenges and complexities of the SQA problem.

A grey car is parking **in front of** a grey house **with brown window frames** and **plants on the balcony**.
Q: Are the **plants in front of the car**? No
Q: Are the **plants in the house**? Yes

Figure 4.7 RESQ - A human-generated dataset for probing the models on realistic SQA.

We created this dataset to reflect the natural complexity of real-world spatial descriptions and questions. We asked three volunteers (English-speaking undergrad students) to generate at least four Yes/No questions for contexts in mSPRL dataset that contains complex human-generated sentences. The questions require at least one step of reasoning. The advantage of RESQ is that the human-generated spatial descriptions and their spatial annotations already exist in the original dataset. On average, the annotators spent 20 minutes generating questions for each group of sentences, which, in total, cost 210 hours to generate the whole data. After gathering the data, another undergrad student checked the questions and removed the incorrect ones. The train set is provided on the train set of mSPRL, and since mSPRL does not have a dev set, we split the 32% of test data (equal to 20% of the training set) and keep it as the dev set. 50% of questions in this dataset have "Yes," and 50% have "No" answers. The statistics of this dataset are provided in Table 4.1.

One of the challenges of the RESQ, which is not addressed here, is that the questions require spatial commonsense knowledge in addition to capturing the spatial semantics. For example, by using commonsense knowledge from the sentence, “a lamp hanging on the ceiling”, we can infer that the lamp is above all the objects in the room.

4.3.1 Version Updates

In the updated version of this dataset, we have implemented several enhancements to improve its quality and utility for research in spatial reasoning. These updates include:

1. Clarification of Question Text: We have reviewed and eliminated ambiguous questions or ones that contained incorrect entity mentions. This step ensures that the questions in the dataset are unambiguous, facilitating more accurate evaluations.
2. Verification and Correction of Answers: The answers to the questions in the dataset have been thoroughly double-checked and corrected where necessary. This update ensures the reliability and accuracy of the dataset.
3. Introduction of "non-commonsense" and "commonsense" Labels: We have added labels to the questions to indicate whether they require "non-commonsense" or "commonsense" spatial reasoning. This update helps to have a finer granularity in evaluating and analyzing the performance of models on different types of reasoning tasks.
4. Expansion with Additional Non-Commonsense Questions: We have enriched the dataset by adding extra questions that specifically focus on evaluating models' multi-hop spatial reasoning capabilities while disregarding their commonsense understanding.

Overall, these updates enhance the dataset's quality, making it a more reliable and effective tool for evaluating the spatial reasoning capabilities of various models.

4.4 Experiments and Results

Dataset	Train	Dev	Test
bAbI	8992	992	992
SPARTQA-AUTO (YN)	26152	3860	3896
SPARTQA-AUTO (FR)	25744	3780	3797
SPARTQA-HUMAN (YN)	162	51	143
SPARTQA-HUMAN (FR)	149	28	77
RESQ	1008	333	610
StepGame	50000	1000	10000
SPARTUN (YN)	20334	3152	3193
SPARTUN (FR)	18400	2818	2830

Table 4.1 Size of SQA benchmarks.

In this section, we analyze the impact of our synthetic benchmarks, SPARTQA-AUTO and SPARTUN, as an extra source of supervision on SQA tasks using the PLM-based models described in Section 3.4. In all experiments, we compare the performance of models *fine-tuned with the target datasets* with and without *further pretraining on synthetic supervision (SynSup)*. We select bAbI

tasks 17 and 19, StepGame, and boolQ alongside our proposed benchmarks SPARTQA-HUMAN and RESQ as our target datasets.

The baselines for all experiments include a majority baseline (MB) which predicts the most repeated label as the answer to all questions, and a pretrained language model, that is, BERT here. We also report the human accuracy in answering the questions for the human-generated datasets¹². For all experiments, to evaluate the models, we measure the accuracy which is the percentage of correct predictions in the test sets.

4.4.1 Experiments Configuration

Since the datasets that we use contain different question types, we supervise the models based on the same question type as the target task¹³.

We use the huggingFace¹⁴ implementation of pretrained BERT which has 768 hidden dimensions. All models are trained on the training set, evaluated on the dev set, and reported the result on the test set. For training, we train the model until no changes happen on the dev set and then store and use the best model on the dev set. We use AdamW ([72]) optimizer on all models and modules. We use Focal Loss [69] with $\gamma = 2$.

$$Loss = \sum \text{CrossEntropyLoss}(p', y') + \text{BCELoss}(p, y) \quad (4.1)$$

Besides, The result of BERT model trained and tested on SPARTUN and SPARTUN are provided in Table 4.2. SPARTUN-Simple only contains one spatial expression for each relation type, and SPARTUN-Clock contains all relation expressions plus clock expressions (Column 5 in Figure 4.5) for relation types. All codes are publicly available at <https://github.com/HLR/Spatial-QA-tasks>.

¹²All human results gathered by scoring the human answers over a subset of the test set.

¹³StepGame only has FR question types. Hence, we use the model trained on FR questions for both FR and YN target tasks.

¹⁴https://huggingface.co/transformers/v2.9.1/model_doc/bert.html

Dataset	YN	FR
SPARTUN	92.83	93.66
SPARTUN-Simple	90.30	93.66
SPARTUN-Clock	-	87.13
SPARTQA	82.05	94.17

Table 4.2 Result of BERT (SQA) model trained and test on two synthetic supervision data.

4.4.2 Further Pretraining on SPARTQA-AUTO Improves Spatial Reasoning

Table 4.3 shows performance on SPARTQA-HUMAN in a low-resource setting, where 0.6k QA pairs from SPARTQA-HUMAN are used for fine-tuning these LMs and 0.5k for testing.

#	Model	FB	FR	CO	YN	Avg
1	Majority	28.84	24.52	40.18	53.60	36.64
2	BERT	16.34	20	26.16	45.36	30.17
3	BERT (Stories only; MLM)	21.15	16.19	27.1	51.54	32.90
4	BERT (SPARTQA-AUTO; MLM)	19.23	29.54	32.71	47.42	34.88
5	BERT (SPARTQA-AUTO)	62.5	46.66	32.71	47.42	47.25
6	Human	91.66	95.23	91.66	90.69	92.31

Table 4.3 Further pretraining BERT on SPARTQA-AUTO improves accuracies on SPARTQA-HUMAN. All models are fine-tuned on the training data of SPARTQA-HUMAN, but line 3-5 are also further pretrained in different ways. line 3: further pretrained on the stories from SPARTQA-AUTO as a masked language model (MLM) task. line 4: further pretrained on both stories and QA annotations as MLM. line 5: the proposed model that is further pretrained on SPARTQA-AUTO as a QA task. Avg: The micro-average on all four Q-TYPES.

In Table 4.3, line 5, BERT (SPARTQA-AUTO), is the proposed method of further pretraining BERT on SPARTQA-AUTO. We can see that line 2, the original BERT, performs consistently lower than BERT(SPARTQA-AUTO), indicating that having SPARTQA-AUTO as a further pretraining task improves BERT’s spatial understanding.

In addition, we implement another two baselines. BERT (Stories only; MLM): further pretraining BERT only on the stories of SPARTQA-AUTO as a masked language model (MLM) task; BERT (SPARTQA-AUTO; MLM): we convert the QA pairs in SPARTQA-AUTO into textual statements and further pretrain BERT on the text as an MLM (see Fig. 4.8 for an example conversion).

To convert each question and its answer into a sentence, we utilize static templates for each question type which removes the question words and rearranges other parts into a sentence.

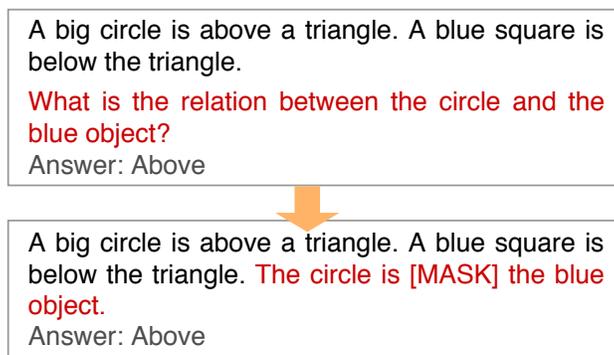


Figure 4.8 Convert a triplet of (paragraph, question, answer) into a single piece of text for the MLM task.

We can see that line 3 slightly improves over original BERT, an observation consistent with many prior works that seeing more text generally helps an LM (e.g., [34]). The significant gap between line 3 and the proposed line 5 (BERT (SPARTQA-AUTO)) indicates that supervision signals come more from our annotations in SPARTQA-AUTO rather than from seeing more unannotated text. Line 4 is another way to make use of the annotations in SPARTQA-AUTO, but it is shown to be not as effective as further pretraining BERT on SPARTQA-AUTO as a QA task.

While the proposed BERT (SPARTQA-AUTO) overall performs better than the other three baseline models, one exception is its accuracy on YN, which is lower than that of line 3. Since all models' YN accuracies are also lower than the majority baseline, we hypothesize that this is due to imbalanced data. To verify it, we compute the F_1 score for YN Q-TYPE in Table 4.4, where we see all models effectively achieve better scores than the majority baseline. However, further pretraining BERT on SPARTQA-AUTO still does not beat other baselines, which implies that straightforward pretraining is not necessarily helpful in capturing the complex reasoning phenomena required by YN questions.

Model	F_1
Majority	35
BERT	50
BERT (Stories only; MLM)	53
BERT (SPARTQA-AUTO; MLM)	48
BERT (SPARTQA-AUTO)	48

Table 4.4 Switching from accuracy in Table 4.3 to F_1 shows that the models are all performing better than the majority baseline on YN Q-TYPE.

The human performance is evaluated on 100 random questions from each SPARTQA-AUTO and SPARTQA-HUMAN test set. The respondents are graduate students that were trained by some examples of the dataset before answering the final questions. We can see from Table ?? that all models’ performances fall behind human performance by a large margin.

4.4.2.1 Extrinsic Evaluation

In this subsection, we show further pretraining BERT on SPARTQA-AUTO can achieve better performance on two extrinsic evaluation datasets, namely bAbI and boolQ.

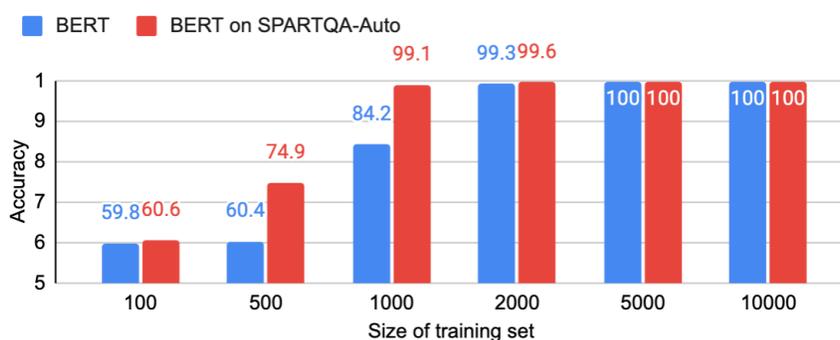


Figure 4.9 Learning curve of BERT and BERT further pretrained on SPARTQA-AUTO on bAbI.

We draw the learning curve on bAbI, using the original BERT as a baseline and BERT further pretrained on SPARTQA-AUTO (Fig. 4.9). Although both models achieve perfect accuracy given large enough training data (i.e., 5k and 10k), BERT (SPARTQA-AUTO) is showing better scores given less training data. Specifically, to achieve an accuracy of 99%, BERT (SPARTQA-AUTO) requires 1k training examples, while BERT requires twice as much. We also notice that BERT (SPARTQA-AUTO) converges faster in our experiments.

As another evaluation dataset, we chose boolQ [11] (Check Figure 4.10) for two reasons. First, we needed a QA dataset with Yes/No questions. To our knowledge boolQ is the only available one used in the recent work. This dataset is provided on Wikipedia and is designed to be answerable by reading the corresponding passage of text and do not require any external knowledge. Second, indeed, SPARTQA and boolQ are from different domains, however, boolQ needs multi-step reasoning in which we wanted to see if SPARTQA helps.

Q: Has the UK been hit by a hurricane?
P: The Great Storm of 1987 was a violent extratropical cyclone which caused casualties in England, France and the Channel Islands ...
A: Yes. [An example event is given.]

Q: Does France have a Prime Minister and a President?
P: ... The extent to which those decisions lie with the Prime Minister or President depends upon ...
A: Yes. [Both are mentioned, so it can be inferred both exist.]

Figure 4.10 An example of boolQ dataset.

Model	Accuracy
Majority baseline	62.2
Recurrent model (ReM)	62.2
ReM fine-tuned on SQuAD	69.8
ReM fine-tuned on QNLI	71.4
ReM fine-tuned on NQ	72.8
BERT (our setup)	71.9
BERT (SPARTQA-AUTO)	74.2

Table 4.5 Models performances on the dev set of boolQ (since the test set is not available to us). Top: numbers reported in [11]. Bottom: numbers from our experiments. BERT (SPARTQA-AUTO): further pretraining BERT on SPARTQA-AUTO as a QA task.

Table 4.5 shows that further pretraining BERT on SPARTQA-AUTO yields a better result than the original BERT and those reported numbers in clark2019boolq, which also tested on various distant supervision signals such as SQuAD [107], Google’s Natural Question dataset NQ [57], and QNLI from GLUE [138].

We observe that many of the boolQ examples answered correctly by the BERT further pretrained on SPARTQA-AUTO require multi-step reasoning. Our hypothesis is that since solving SPARTQA-AUTO questions needs multi-step reasoning, fine-tuning BERT on SPARTQA-AUTO generally improves this capability of the base model.

4.4.3 Impact of Broader Relation Coverage on Transfer Learning

The following experiments demonstrate the impact SPARTUN with broader coverage of spatial relation types and expressions on transfer learning for SQA benchmarks compared to different supervisions.

Model	SynSup	17^{1k}	19⁵⁰⁰
MB	-	51.9	10.6
BERT	-	87.39	34.53
BERT	SPARTQA-AUTO	90.42	100
BERT	StepGame	87.39	99.89
BERT	SPARTUN-Simple	92.43	98.99
BERT	SPARTUN	90.02	99.89

Table 4.6 Impact of using synthetic supervision on the bAbI tasks. All the models are further fine-tuned on the training set of task 17 (size = 1k) and 19 (size = 500), and test on bAbI test sets.

bAbI(task 17 and 19): Task 17 is on spatial reasoning and contains binary Yes/No questions. Task 19 is on pathfinding and contains FR questions with answers in {LEFT, RIGHT, ABOVE, BELOW} set. The original dataset contains west, east, north, and south, which we mapped to their corresponding relative relation type. Due to the simplicity of **bAbI** dataset, our experiments show that PLM can solve this benchmark with 100% accuracy when trained on the full training data(10k). Hence we run our experiment on only 1k and 500 training examples of task 17 and task 19, respectively.

Table 4.6 demonstrates the impact of synthetic supervision on both tasks of bAbI. The results with various synthetic data are fairly similar for these two tasks. However, pretraining the model with the simple version of SPARTUN, named SPARTUN-Simple, performs better than other synthetic datasets on task 17. This can be due to the fewer relation expressions in SPARTUN-Simple, which follows the same structure as task 17.

Model	SynSup	YN	FR
MB	-	53.60	24.52
BERT	-	49.65	18.18
BERT	SPARTQA-AUTO	39.86	48.05
BERT	StepGame	44.05	11.68
BERT	SPARTUN-Simple	44.75	37.66
BERT	SPARTUN	48.25	50.64
Human	-	90.69	95.23

Table 4.7 Transfer learning on SPARTQA-HUMAN. SPARTQA-A stands for SPARTQA-AUTO.

SPARTQA-HUMAN: In the next experiment, we investigate the impact of SPARTUN on **SPARTQA-HUMAN** result. Comparing the results in Table 4.7, we find that even though the classification layer for SPARTQA-AUTO and SPARTQA-HUMAN are the same, the model trained on SPARTUN has a better

Story:

- 0:"B is south east of J."
- 1:"X is under E."
- 2:"K is to the left of Z and is on the **same horizontal plane**."
- 3:"If L is the **center of a clock face**, E is **located between 10 and 11**."
- 4:"S is positioned above Q."
- 5:"Q is **diagonally to the bottom right** of L."
- 6:"C and S are horizontal and C is to the left of S."
- 7:"I is **above** B with a small gap between them."
- 8:"E is **above** N and **to the left** of N."
- 9:"Q is **below and to the right** of B."
- 10:"X is to the left of C with a small gap between them."

question:"What is the relation of the agent L to the agent J? "lower-right"

Figure 4.11 StepGame. An example of questions that need ten steps of reasoning.

transferability. It achieves 2.6% better accuracy on FR and 9% better accuracy on YN questions compared to SPARTQA-AUTO. YN is, yet, the most challenging question type in SPARTQA-HUMAN and none of the PLM-based models can reach even the simple majority baseline.

Model	SynSup	k steps of reasoning									
		1	2	3	4	5	6	7	8	9	10
TP-MANN	-	85.77	60.31	50.18	37.45	31.25	28.53	26.45	23.67	22.52	21.46
BERT	-	98.44	94.77	91.78	71.7	57.56	50.34	45.17	39.69	35.41	33.62
BERT	SPARTQA-AUTO	98.63	94.95	91.94	77.74	68.37	61.67	57.95	50.82	46.86	44.03
BERT	SPARTUN-Simple	98.70	95.21	92.46	77.93	69.53	62.14	57.37	48.79	44.67	42.72
BERT	SPARTUN	98.55	95.02	92.04	79.1	70.34	63.39	58.74	52.09	48.36	45.68

Table 4.8 Result of models with and without extra synthetic supervision on StepGame.

StepGame [118]: is a synthetic SQA dataset (See Figure 4.11) containing FR questions which need k reasoning steps to be answered ($k = 1$ to 10). This paper was published in 2022 with the same contribution as SPARTQA-AUTO. However, it contains more relation expressions and less relation types and rules. The answer to each question is one relation in {left, right, below, above, lower-left, upper-right, lower-right, upper-left} set. This dataset was provided one year after SPARTQA with the same purpose. However, it contains fewer relation types and rules.

Table 4.8 demonstrates our experiments on **StepGame**. BERT without any extra supervision significantly outperforms the best reported model in shi2022stepgame, TP-MANN, which is based

on a neural memory network. As expected, all the PLM-based models almost solve the questions with one step of reasoning (i.e. where the answer directly exists in the text). However, with increasing the steps of reasoning, the performance of the models decreases. Comparing the impact of different synthetic supervision, SPARTUN achieves the best result on $k > 3$. For questions with $k \leq 3$, SPARTUN-Simple achieves competitive similar results compared to SPARTUN. Overall, the performance gap in SPARTUN-Simple, SPARTQA-AUTO and SPARTUN shows that more coverage of relation expressions in SPARTUN is effective.

Model	SynSup	Accu
MB	-	50.21
BERT	-	57.37
BERT	SPARTQA-AUTO	55.08
BERT	StepGame	60.14
BERT	SPARTUN-Simple	58.03
BERT	SPARTUN	63.60
Human	-	90.38

Table 4.9 Results with and without extra supervision on ReSQ. The Human accuracy is the performance of human on answering a subset of test set.

ReSQ: In the next experiment, we show the influence of SPARTUN on real-world examples, which contain more types of spatial relations and need more rules of reasoning to be solved. Table 4.9 shows the result of transfer learning on **ReSQ**. To compute the human accuracy, we asked two volunteers (one from those who created the questions and one new volunteer) to answer 100 questions from the test set of **ReSQ**. Finally, a third student graded their answers and computed the accuracy. This result shows that the limited coverage of spatial relations and expression in SPARTQA-AUTO impacts the performance of BERT negatively. However, further pretraining BERT on SPARTUN-Simple improves the result on **ReSQ**. This can be due to the higher coverage of relation types in SPARTUN-Simple than SPARTQA-AUTO. Using SPARTUN for further pretraining BERT has the best performance and improves the result by 5.5%, indicating its advantage for transferring knowledge to solve real-world spatial challenges.

In conclusion, our experiments show the efficiency of SPARTUN in improving the performance

of models on different benchmarks due to the flexible coverage of relation types and expressions.

4.5 Related Research

Requiring large amounts of annotated data is a well-known issue in training complex deep neural models [161] that is extended to spatial language processing tasks. In our study, we noticed that all available large datasets on SQA task including bAbI [143], SPARTQA-AUTO, and StepGame [118] are, all, synthetic.

bAbI is a simple dataset that covers a limited set of relation types, spatial rules, and vocabulary. StepGame focuses on a few relation types but with more relation expressions for each and considers multiple reasoning steps. SPARTQA-AUTO, comparatively, contains more relation types and needs complex multi-hop spatial reasoning. However, it contains a single linguistic spatial expression for each relation type. All of these datasets are created based on controlled toy settings and are not comparable with real-world spatial problems in the sense of realistic language complexity and coverage of all possible relation types. SPARTQA-HUMAN is a human-generated version of SPARTQA-AUTO with more spatial expressions. However, this dataset is provided for probing purposes and has a small training set that is not sufficient for effectively training deep models.

Transfer learning has been used effectively in different NLP tasks to further fine-tune the PLMs [109, 2, 77]. Besides transfer learning, several other approaches are used to tackle the lack of training data in various NLP areas, such as providing techniques to label the unlabeled data [24], using semi-supervised models [132, 65] or data augmentation with synthetic data [67, 86]. However, transfer learning is a simple way of using synthetic data as an extra source of supervision at no annotation cost. Compared to the augmentation methods, the data in the transfer learning only needs to be close to the target task/domain [76] and not necessarily the same.

Using logical reasoning for building datasets that need complex reasoning for question answering is used before in building QA datasets [12, 113]. More recent efforts even use the path of reasoning and train models to follow that [126]. However, there are no previous works to model spatial reasoning as we do here with the broad coverage of spatial logic.

4.6 Summary

To improve LMs’ capability on spatial reasoning, we use a synthetic dataset, SPARTQA-AUTO, as a source of supervision for transfer learning for spatial question answering (SQA) which contains a large corpus of spatial descriptions and corresponding question-answer annotations and show that further pretraining LMs on this extra supervision significantly enhances their spatial language understanding and reasoning. We also show that a spatially-improved LM can have better results on two extrinsic datasets (bAbI and boolQ).

We also create a new synthetic dataset and demonstrate that expanding the coverage of relation types and combinations and spatial language expressions can provide a more robust source of supervision for pretraining and transfer learning [89]. As a result, this data improves the models’ performance in many experimental scenarios on both tasks when tested on various evaluation benchmarks. This data includes rules of spatial reasoning and the chain of logical reasoning for answering the questions that can be used for further research in the future.

Moreover, we provide a human-generated dataset on a realistic SQA task that can be used to evaluate the models and methods for spatial language understanding related tasks in real-world problems.

CHAPTER 5

SPATIAL ROLE LABELING MODELS

5.1 Introduction

As we described in Section 2.1.1, Spatial Role Labeling (SpRL), which can also be considered as spatial information extraction, is the task of identifying and classifying the spatial roles and their relations. Since this task is fundamental and one of the key points to spatial language understanding, in this section, we provide models to solve this task. We call the former **spatial role extraction** and the latter **spatial relation¹ extraction** (Figure 5.1).

The SpRL model contains two pipeline models, one for **spatial role extraction** and another for **spatial relation² extraction** (Figure 5.1). The spatial relation extraction model is implemented for two joint tasks, spatial triplet validation, and spatial relation type classification. We show how this model outperforms previous SOTAs on mSPRL benchmark.

Next, we analyze the impact of SPARTUN as a source of extra supervision on several SpRL benchmarks. To the best of our knowledge, we are the first to use synthetic supervision for the SpRL task. Our results show that the auto-generated data successfully improves the SOTA results on mSPRL and SPARTQA-HUMAN, which are annotated for SpRL task.

5.2 Models

In this section, we describe the implemented SpRL models. Several neural models have been proposed to solve spatial role [83, 74, 15]. We take a similar approach to prior research [119] for the extraction of spatial roles (entities (Trajector/Landmark) and spatial indicators).

First, we separately tokenize each sentence in the context and use a PLM (which is BERT here) to compute the tokens representation. Next, we apply a BIO tagging layer on tokens representations using (O, B-entity, I-entity, B-indicator, I-indicator) tags. A Softmax layer on BIO tagger output is used to select the spatial entities and spatial indicators with the highest probability. For training, we use CrossEntropy loss given the spatial annotation.

¹In different works like [54], the triplet and relation are used interchangeably.

²In different works like [54], the triplet and relation are used interchangeably.

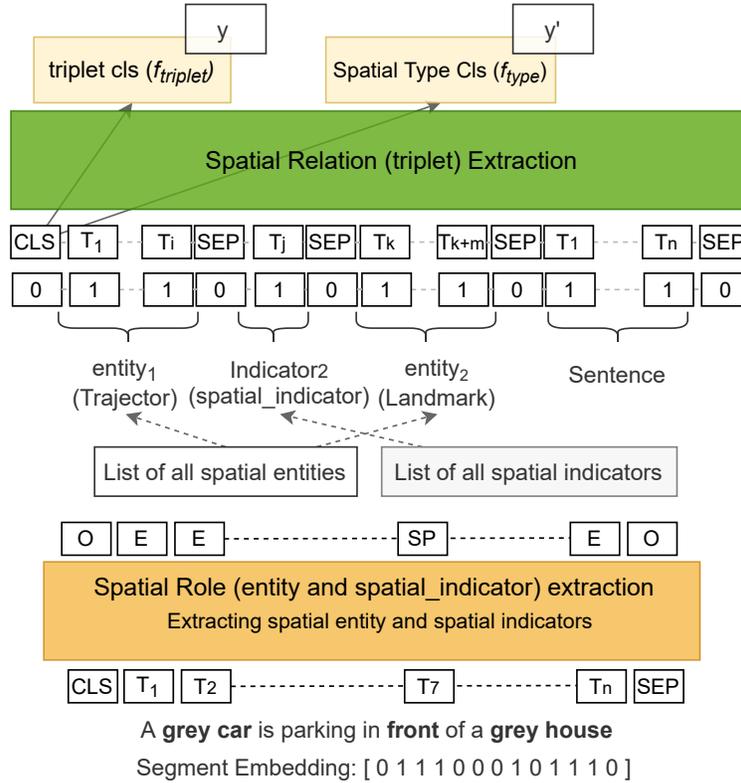


Figure 5.1 Spatial role labeling model includes two separately trained modules. E: entity, SP: spatial_indicators. As an example, triplet (a grey house, front , A grey car) is correct and the “spatial_type = FRONT”, and (A grey car, front, a grey house) is incorrect, and the “spatial_type = NaN”.

For the spatial relation extraction model, similar to [150, 119], we use BERT and a classification layer to extract correct triplets. Given the output of the spatial role extraction model, for each combination of (spatial entity(tr), spatial_indicator(sp), spatial entity(lm)) in each sentence we create an input³ and pass it to the BERT model. To indicate the position of each spatial role in the sentence, we use segment embeddings and add 1 if it is a role position and 0 otherwise.

The [CLS] output of BERT will be passed to a one-layer MLP that provides the probability for the triplet. Compared to the prior research, we predict the spatial type for each triplet as an auxiliary task for spatial relation extraction. To this aim, we apply another multi-class classification layer⁴ on the same [CLS] token.

For spatial argument extraction, we use cross-entropy loss for BIO-tagging, and for spatial relation

³[CLS, tr , SEP, sp , SEP, lm , SEP, sentence, SEP]

⁴The classes are relation types in Table 2.1 alongside a NaN class for incorrect triplets.

Formalism (General Type)	Specific value	Spatial type/Spatial value)	Expressions (e.g.)
Topological	RCC8	DC (disconnected) EC (Externally Connected) PO (Partially Overlapped) EQ (Equal) TPP (Tangential Proper Part) NTPP (Non-Tangential Proper Part) TPPI (Tangential Proper Part inverse) NTPPI (Non-Tangential Proper Part inverse)	disjoint touching overlapped equal covered by in, inside covers has
Directional	Relative	LEFT, RIGHT BELOW, ABOVE BEHIND, FRONT	left of, right of under, over behind, in front
Distance	Qualitative	Far, Near	far, close

Table 5.1 Spatial relation types and examples of spatial language expressions.

extraction, we use the summation of loss for each spatial relation and relation type classification part.

$$\begin{aligned}
 Loss = \sum \text{CrossEntropyLoss}(p', y') \\
 + \text{BCELoss}(p, y)
 \end{aligned}
 \tag{5.1}$$

5.3 Experiments and Results

The pipeline model provided in Section 5.2, contains three main parts, a model for spatial role extraction (SRole), a model for spatial relation extraction (SRel), and a model for spatial relation type classification (SType) which we analyze separately (Check Table 6.4). We use Macro F1-score (mean of F1 for each class) to evaluate the performance of the SRole, SRel, and SType models.

Here, we first analyze the superiority of our proposed model compared to the previous SOTA on mSpRL model. Next, we analyze the influence of the extra synthetic supervision on SpRL task when evaluated on human-generated datasets, mSpRL and SPARTQA-HUMAN. To do this, we further pretrain the BERT module in these models and then fine-tune it on the target domain. Table 5.2 shows the number of sentences in each SpRL benchmark.

As shown in Table 6.4, all SpRL submodules achieve high results on synthetic datasets, SPARTQA and SPARTUN. The reason behind this good performance is that these datasets may contain less ambiguity coming from the natural language expressions. Therefore, the BERT-base models can

Dataset	Train	Dev	Test
SPARTQA-AUTO (story)	25755	16214	16336
SPARTQA-AUTO (question)	23584	15092	15216
SPARTQA-HUMAN (story)	176	99	272
SPARTQA-HUMAN (question)	155	127	367
SPARTUN (story)	48368	7031	7191
SPARTUN (question)	38734	5970	6023
mSPRL	481	-	461

Table 5.2 Number of sentences of SpRL benchmarks. To train the SPARTQA-AUTO, we only use the 3k training examples (23 - 25k sentences).

Model	Dataset	SRole	SRel	SType
R-Inf*	mSPRL	80.92	68.78	-
BERT		88.59	69.12	19.79
BERT (SPARTQA-AUTO)		88.41	68.84	
BERT (SPARTUN)		88.03	71.23	23.65
BERT* (SPARTUN)		87.74	70.18	26.02
BERT	SPARTQA-HUMAN	55.8	S: 57.43 Q: 52.55	43.79 39.34
BERT (SPARTQA-AUTO)		57.28	S: 58.32 Q: 55.17	
BERT (SPARTUN)		72.53	S: 60.24 Q: 61.53	48.74 48.07
BERT	SPARTQA-AUTO	99.92	S: 99.72 Q: 98.36	99.05 98.62
BERT	SPARTUN	99.96	S: 99.18 Q: 97.68	98.57 98.11

Table 5.3 Extraction modules used in PISTAQ. Q: question and S: stories. Certain sentences in the mSPRL datasets are considered as conveying no spatial information by the original SpRL dataset annotators. These sentences are included in experiments where specified by an asterisk (*) and are excluded from all other experiments.

easily capture the spatial language’s syntactic patterns that are needed for extracting the roles and direct relations from the large training set.

Table 6.4 also demonstrates the influence of synthetic supervision in spatial role extraction evaluated on mSPRL and SPARTQA-HUMAN.

We compare the result of the SRole model with the previous SOTA, “R-Inf” [78], on mSPRL dataset. R-Inf uses external multi-modal resources and global inference. All of the BERT-based SRole models outperform the R-Inf, which shows the power of PLMs for this task. However, since

the accuracy of the SRole is already very high, using synthetic supervision shows no improvements compared to the model that only trained with mSPRL training set for the SRole. In contrast, on SPARTQA-HUMAN, using synthetic supervision helps the model perform better. Especially, using SPARTUN increases the performance of the SRole model dramatically by 15%.

In Table 6.4, we show the result of the SRel model (containing spatial relation extraction and spatial relation type classification) for spatial relation extraction, with and without extra supervision from synthetic data. Same as the SRole model, extra supervision from SPARTUN achieves the best result when tested on SPARTQA-HUMAN.

For mSPRL, we compared the SRel model with R-Inf on spatial relation extraction. As table 6.4 demonstrates we improve the SOTA by 2.6% on F1 measure using SPARTUN as synthetic supervision. Also, the model further pretrained on SPARTQA-AUTO gets lower results than the model with no extra supervision due to the limited relation expressions used in this data.

In conclusion, our experiments show the efficiency of SPARTUN in improving the performance of models on different benchmarks due to the flexible coverage of relation types and expressions.

5.4 Related Research

For the SpRL task, mSPRL and SpaceEval (SemEval-2015 task 8) [102] are two available datasets with spatial roles and relation annotations. These are small-scale datasets for studying the SpRL problem. From the previous works which tried transfer learning on SpRL task, [94] only used it on word embedding of their SpRL model, and [119] used PLM without any specifically designed dataset for further pretraining. These issues motivated us to create SPARTUN for further pretraining and transfer learning for SQA and SpRL.

5.5 Summary

We demonstrate that our proposed SpRL model surpasses the previous SOTA model on mSPRL benchmark [89]. Besides, we show that our generated synthetic dataset can be helpful as a source of supervision for transfer learning for spatial role labeling (SpRL) tasks. We show that expanding the coverage of relation types and combinations and spatial language expressions can provide a more robust source of supervision for pretraining and transfer learning. As a result, this data improves

the models' performance in many experimental scenarios on both tasks when tested on various evaluation benchmarks.

CHAPTER 6

DISENTANGLEING EXTRACTION AND REASONING IN MULTI-HOP SPATIAL REASONING

6.1 Introduction

Despite the high performance of recent pretrained language models on question-answering (QA) tasks, solving questions that require multi-hop reasoning is still challenging [82]. In chapter 3, we have investigated the use of general end-to-end deep neural models such as pretrained language models (PLM) [91] in spatial question answering (SQA). PLMs show reasonable performance on the SQA problem and can implicitly learn spatial rules from a large set of training examples. However, the black-box nature of PLMs makes it unclear whether these models are making the abstractions necessary for spatial reasoning or their decisions are based solely on patterns observed in the data.

As a solution for better multi-hop reasoning, recent research has investigated the impact of using fine-grained information extraction modules such as Named Entity Recognition (NER) [93, 84], gated Entity/Relation [157] or semantic role labels (SRL) [116, 25] on the performance of models.

On a different thread, cognitive studies [121, 20] show when the given information is shorter, humans also find spatial abstraction and use spatial rules to infer implicit information. Figure 6.1 shows an example of such extractions. Building upon these findings, we aim to address the limitations of end-to-end models and capitalize on the advantages of fine-grained information extraction in solving SQA. Thus, we propose models which disentangle the *language understanding* and *spatial reasoning* computations as two separate components. Specifically, we first design a pipeline model that includes trained neural modules for extracting direct fine-grained spatial information from the text and performing symbolic spatial reasoning over them.

The second model is simply an end-to-end PLM that uses annotations used in extraction modules of pipeline model in the format of *extra QA* supervision. This model aims to demonstrate the advantages of using separate extraction modules compared to a QA-based approach while utilizing the same amount of supervision. Ultimately, the third model is an end-to-end PLM-based model on relation extraction tasks that has explicit latent layers to disentangle the extraction and reasoning

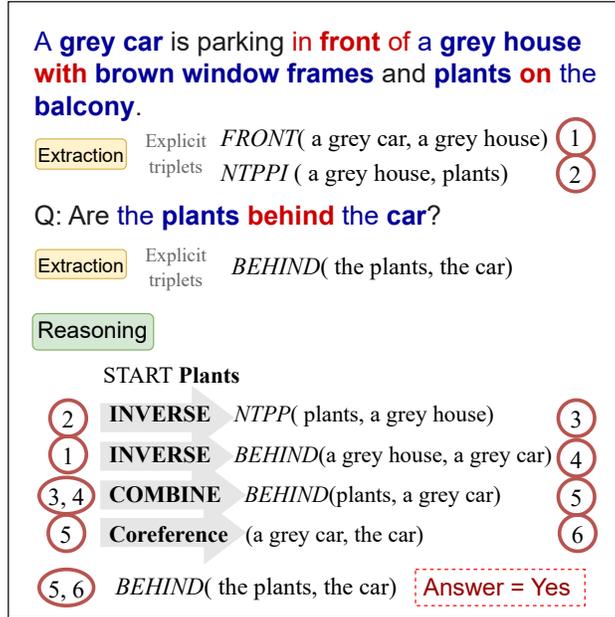


Figure 6.1 An example of steps of spatial reasoning on RESEQ dataset. We begin by searching for *the plants* from the question triplet within the text, enabling us to extract explicit triplets (1,2). Next, we apply rules such as INVERSE to deduce implicit triplets (3,4,5). Then, utilizing triplets 5 and 6 we determine the final answer, 'Yes'. NTPP: Non-Tangential Proper Part (Table 6.1).

Formalism (General Type)	Spatial Type	Expressions (e.g.)
Topological (RCC8)	DC (disconnected) EC (Externally Connected) PO (Partially Overlapped) EQ (Equal) TPP NTPP TPPI NTPPI	disjoint touching overlapped equal covered by in, inside covers has
Directional (Relative)	LEFT, RIGHT BELOW, ABOVE BEHIND, FRONT	left of, right of under, over behind, in front
Distance	Far, Near	far, close

Table 6.1 List of spatial relation formalism and types.

inside the model. This model incorporates a neural spatial reasoner, which is trained to identify all spatial relations between each pair of entities.

We evaluate the proposed models on multiple SQA datasets, demonstrating the effectiveness of the disentangling extraction and reasoning approach in controlled and realistic environments. Our pipeline outperforms existing SOTA models by a significant margin on benchmarks with a

controlled environment (toy tasks) while utilizing the same or fewer training data. However, in real-world scenarios with higher ambiguity of natural language for extraction and more rules to cover, our end-to-end model with explicit layers for extraction and reasoning performs better.

These results show that disentangling extraction and reasoning benefits deterministic spatial reasoning and improves generalization in realistic domains despite the coverage limitations and sensitivity to noises in symbolic reasoning. These findings highlight the potential of leveraging language models for information extraction tasks and emphasize the importance of explicit reasoning modules rather than solely depending on black-box neural models for reasoning.

6.2 Related Research

End-to-end model on SQA: To solve SQA tasks, recent research evaluates the performance of different deep neural models such as Memory networks [118, 125], Self-attentive Associative Memory [61], subsymbolic fully connected neural network [160], and Recurrent Relational Network (RRN) [100]. In Chapter 3, we used transfer learning and provided large synthetic supervision that enhances the performance of PLMs on spatial question answering. However, the results show a large gap between models and human performance on human-generated data. Besides, none of these models use explicit spatial semantics to solve the task. The only attempt towards integrating spatial semantics into the spatial QA task is a baseline model we introduced, which uses rule-based spatial semantics extraction for reasoning on bAbI (task 17) and achieves 100% accuracy without using any training data.

Extraction and Reasoning: While prior research has extensively explored the use of end-to-end models for learning the reasoning rules [87, 103], there is limited discussion on separating the extraction and reasoning tasks. [98] utilizes LMs to generate new sentences and extract facts while using some symbolic rules to ensure consistency between generated sentences. Similarly, ThinkSum [99] uses LMs for knowledge extraction (Think) and separate probabilistic reasoning (Sum), which sums the probabilities of the extracted information. However, none of these works are on multi-step or spatial Reasoning.

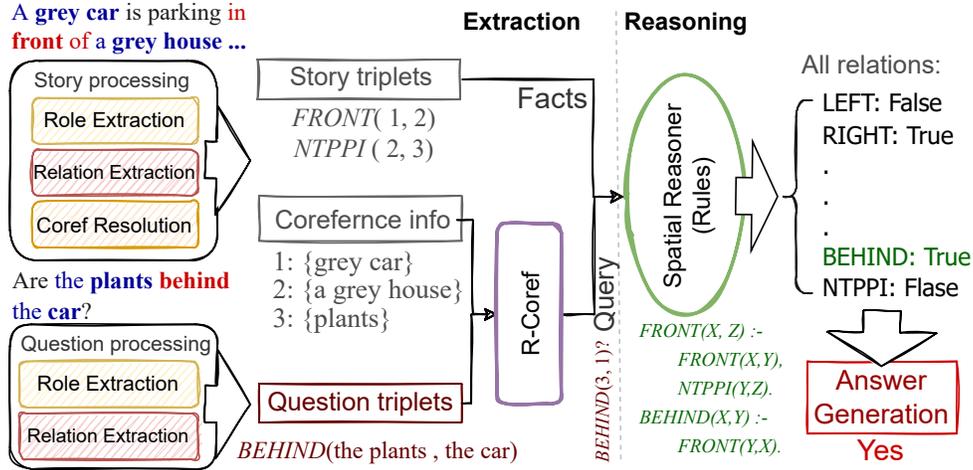


Figure 6.2 PISTAQ pipeline based on disentangled extraction and reasoning. In this model, facts, e.g., $FRONT(\text{grey car}, \text{grey house})$, are extracted from the story and linked by coreference modules. The R-Coref equates ‘the car’ from the question with ‘a grey car’ in the story and forms a query. This query, along with facts, is sent to the spatial reasoner. Finally, the spatial reasoner employs $FRONT$ and $BEHIND$ rules and returns True as the answer.

6.3 Proposed Models

To understand the effectiveness of disentangling the extraction and reasoning modules, we provide three groups of models for spatial question answering (SQA) task. The first model is a pipeline of extraction and symbolic reasoning (§6.3.1), the second model is an end-to-end PLM that uses the same spatial information supervision but in a QA format (§6.3.2), and the third model is an end-to-end neural model with explicit layers of extraction and reasoning (§6.3.3). We elaborate each of these models in the subsequent sections.

6.3.1 Pipeline of Extraction and Reasoning

Here, we describe our suggested pipeline model designed for spatial question answering task, referred to as **PISTAQ**¹. As shown in the extraction part of Figure 6.2, the spatial information is extracted first and forms a set of triplets for a story (Facts) and a question (Query). Then a coreference resolution module is used to connect these triplets to each other. Given the facts and queries, the spatial reasoner infers all implicit relations. The answer generator next conducts the final answer. Below we describe each module in more detail.

Spatial Role Labeling (SpRL) is the task of identifying and classifying the *spatial roles* of phrases

¹PIpipeline model for SpaTiAl Question answering

Not	$\forall(X, Y) \in Entities$	$R \in \{Dir \vee PP\}$	IF $R(X, Y)$	$\Rightarrow NOT(R_reverse(X, Y))$
Inverse	$\forall(X, Y) \in Entities$	$R \in \{Dir \vee PP\}$	IF $R(Y, X)$	$\Rightarrow R_reverse(X, Y)$
Symmetry	$\forall(X, Y) \in Entities$	$R \in \{Dis \vee (RCC - PP)\}$	IF $R(Y, X)$	$\Rightarrow R(X, Y)$
Transitivity	$\forall(X, Y, Z) \in Entities$	$R \in \{Dir \vee PP\}$	IF $R(X, Z), R(Z, Y)$	$\Rightarrow R(X, Y)$
Combination	$\forall(X, Y, Z, H) \in Entities$	$R \in Dir, *PP \in PP$	IF $*PP(X, Z), R(Z, H), *PPi(Z, Y)$	$\Rightarrow R(X, Y)$

Table 6.2 Designed spatial rules [89]. *Dir*: Directional relations (e.g., LEFT), *Dis*: Distance relations (e.g., FAR), *PP*: all Proper parts relations (NTPP, NTPPI, TPPI, TPP), *RCC – PP*: All RCC8 relation except proper parts relations. **PP*: one of TPP or NTPP. **PPi*: one of NTPPi or TPPi.

within a text (including the Trajector, Landmark, and Spatial Indicator) and formalizing their *relations* [54]. Here, we use the same SpRL modules as in Chapter 5. This model first computes the token representation of a story and its question using a BERT model. Then a BIO tagging layer is applied on the tokens representations using (O, B-entity, I-entity, B-indicator, and I-indicator) tags. Finally, a softmax layer on the BIO tagger output selects the spatial entities² (e.g., ‘grey car’ or ‘plants’ in Figure 6.2) and spatial indicators (e.g., ‘in front of’ in Figure 6.2).

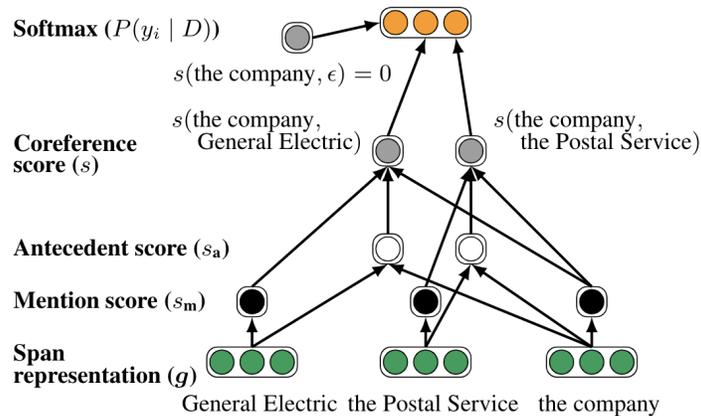
Given the output of the spatial role extraction module, for each combination of (Trajector, Spatial Indicator, Landmark) in each sentence, we create a textual input³ and pass it to a BERT model. The [CLS] output of BERT will be passed to a one-layer MLP that provides the probability for each triplet. To apply the logical rules on the triplets, we need to assign a relation type to each triplet. To this aim, we use another multi-classification layer on the same [CLS] token to identify the spatial types of the triplet. The classes are relation types in Table 6.1 alongside a class NaN for triplet with no spatial meaning. For instance, in Figure 6.2, (grey car, in front of, grey house) is a triplet with *FRONT* as its relation type while (grey house, in front of, grey car) is not a triplet and its relation type is *NaN*. We use a joint loss function for triplet and relation type classification to train the model.

Coreference Resolution Linking the extracted triplets from the stories is another important step required in this task, as different phrases or pronouns may refer to the same entity. To make such connections, we implement a coreference resolution model based on [62] and extract all antecedents for each entity and assign a unique *id* to them. In contrast to previous work, we have extended

²Trajector/Landmark

³[CLS, *traj*, SEP, *indic*, SEP, *land*, SEP, *sentence*, SEP]

the model to support plural antecedents (e.g., two circles). We first collect all mentions of each predicted entity from spatial role extraction or role annotations, then assign an “id” to the same mentions and include that id in each triplet. For example, for BELOW(a cat, a grey car), Front(the car, a church), id 1= a cat, 2 = a grey car, the car, and 3 = a church. So we create new triplets in the form of BELOW(1, 2) and Front(2, 3). To train the model, we pair each mention with its previous



(a) The coreference resolution model structure.

$$s_m(i) = \mathbf{w}_m \cdot \text{FFNN}_m(\mathbf{g}_i)$$

$$s_a(i, j) = \mathbf{w}_a \cdot \text{FFNN}_a([\mathbf{g}_i, \mathbf{g}_j, \mathbf{g}_i \circ \mathbf{g}_j, \phi(i, j)])$$

(b) The formula for computing the coreference scores

Figure 6.3 The coreference resolution model [62].

antecedent and use cross-entropy loss to penalize the model if the correct pair is not chosen. For singletons and starting mention of objects, the model should return class 0, which is the [CLS] token. Since the previous model does not support the plural antecedent (e.g., two circles), we include that by considering shared entities in pairs like both (two circles, the black circle) and (two circles, the blue circle) are true pairs.

As an instance of the importance of coreference resolution in spatial reasoning, consider this context “block A has one black and one green circle. The black circle is above a yellow square. The yellow square is to the right of the green circle. Which object in block A is to the left of a yellow square?” The reasoner must know that the ‘one green circle’ in NTPPI(block A, one green circle)

and the ‘the green circle’ in RIGHT(the yellow square, the green circle) are the same to be able to connect them using transitivity rule and find the answer.

To find the mentions of the question entities in the story and create the queries, we use a Rule-based Coreference (R-Coref) based on exact/partial matching. In Figure 6.2, ‘the car’ in the question has the same id as ‘the grey car’ from the story’s triplets.

Logic-based Spatial Reasoner To do symbolic spatial reasoning, we use the reasoner from Chapter 3. This reasoner is implemented in Prolog and utilizes a set of rules on various relation types, as illustrated in Table 6.2. Given the facts and queries in Prolog format, the spatial reasoner can carry out the reasoning process and provide an answer to any given query. The reasoner matches variables in the program with concrete values and a backtracking search to explore different possibilities for each rule until a solution is found. As shown in Figure 6.2, the reasoner uses a FRONT and a BEHIND rules over the facts and generates the True response for the query.

6.3.2 PLMs Using SpRL Annotations

To have a fair comparison between the QA baselines and models trained on SpRL supervision, we design **BERT-EQ**⁴. We convert the SpRL annotation into extra YN questions⁵ asking about explicit relations between a pair of entities. To generate extra questions, we replace triplets from the SpRL annotation into the “Is [Trajector] [Relation*] [Landmark]?” template. The [Trajector] and [Landmark] are the entity phrases in the main sentence ignoring pronouns and general names (e.g., “an object/shape”). The [Relation*] is a relation expression (examples presented in Table 6.1) for the triplet relation type. To have equal positive and negative questions, we reverse the relation in half of the questions. We train BERT-EQ using both original and extra questions by passing the “question+story” into a BERT with answers classification layers.

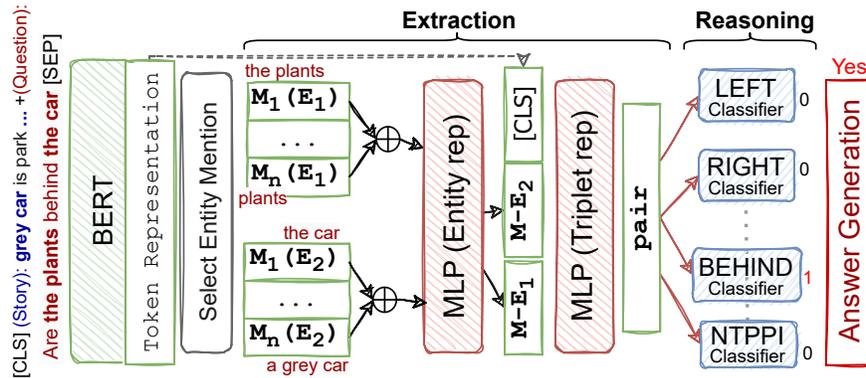
6.3.3 PLMs with Explicit Extractions

As another approach, we aim to explore a model that disentangles the extraction and reasoning parts inside a neural model. Here, rather than directly predicting the answer from the output of

⁴BERT+Extra Question

⁵This augmentation does not apply to FR type since it inquires about all relations between the two asked entities.

PLMs (as typically done in the QA task), we introduce explicit layers on top of PLM outputs. These layers are designed to generate representations for entities and pairs of entities, which are then passed to neural layers to identify all relations. We call this model **SREQA**⁶, which is an end-to-end spatial relation extraction model designed for QA. Figure 6.4 illustrates the structure of this model.



(a) Model structure. First, entity mentions such as ‘plants’ and ‘grey car’ are selected from the BERT output and the entity representation is formed. Next, triplets like (‘plants’, ‘car’, [CLS]) are generated and fed into the reasoning component. The collective output of all relation classifiers determines the relationships between each pair. *All hatched parts are trained end-to-end. The rest of the data is obtained from annotations or off-the-shelf modules.

Source of	Entity Mentions	Relation Supervision
All story relations (Training step#1)	Entity + Coref annotation	SpRL + Coref annotation -> Spatial Reasoner
Question relation (Training step#2)	Trained entity extraction + R-Coref	QA Answer

(b) The source of supervision in each step of training. In step#1, we train the model on all story relations, and in step#2, we only train it on question relations. These modules and data are the same as the ones used in PISTAQ.

Figure 6.4 The SREQA model with explicit neural layers to disentangle extraction and reasoning part.

In this model, we first select the entity mentions ($M_j(E_1)$) from the BERT tokens representation and pass it to the extraction part shown in Figure 6.4a. Next, the model computes entity representation ($M(E_1)$) by summing the BERT token representations of all entity’s mentions and passing it to an MLP layer. Then, for each pair of entities, a triplet is created by concatenating the pair’s entities representations and the BERT [CLS] token representation. This triplet is passed through an MLP layer to compute the final pair representations. Next, in the reasoning part in Figure 6.4a, for

⁶Spatial Relation Extraction for QA

each relation type in Table 6.1, we use a binary 2-layer MLP classifier to predict the probability of each relation between the pairs. We remove the inconsistent relations by selecting one with a higher probability at inference time, e.g., LEFT and RIGHT cannot be true at the same time. The final output is a list of all possible relations for each pair. This model is trained using the summation of Focal loss [69] of all relation classifiers.

We train SREQA in two separate steps. In the first step, the model is trained on a relation extraction task which extracts *all direct and indirect relations* between *each pair of entities* only from stories. The top row of Figure 6.4b shows the annotation and modules employed in this step to gather the necessary supervision. We use the entity and coreference annotation to select the entity mentions from the BERT output. To compute the relations supervision for each pair of entities, we employ the spatial reasoner from PISTAQ and apply it to the direct relations (triplets) from the SpRL annotation, which are connected by coreference annotations. This training step is only feasible for datasets with available SpRL and coreference annotations.

In the next step, we further train SREQA on extracting *questions relation* using QA supervision. As shown in the bottom row of Figure 6.4b, we employ the trained spatial role extraction model used in PISTAQ to identify the entities in the question and use R-Coref to find the mentions of these entities in the story. To obtain the relation supervision, we convert the question answers to relation labels. In FR, the label is similar to the actual answer, which lists all relations. In YN, the question relation is converted to a label based on the Yes/No answer. For example, in Figure 6.4a, the question relation is ‘BEHIND,’ and the answer is Yes, so the label for the BEHIND classifier is 1.

We evaluate the SREQA model’s performance in predicting the accurate answers of the test set’s questions, the same as training step 2.

6.4 Experiments

We evaluate the proposed models on the datasets presented in Chapter 3, SPARTQA, SPARTUN, and RESQ. The first two datasets, SPARTQA and SPARTUN, are pre-annotated with both SpRL and coreference annotations. For the RESQ dataset, we utilize the provided SpRL annotations from mSPRL.

6.4.1 Model Configurations & Baselines

We compare the models described in section 6.3 with the following baselines.

Majority Baseline: This baseline selects the most frequent answer(s) in each dataset.

GT-PISTAQ: This model uses ground truth (GT) values of all involved modules in PISTAQ to eliminate the effect of error propagation in the pipeline. This baseline is used to evaluate the alignments between the questions and story entities and the reasoning module in solving the QA task. It also gives an upper bound for the performance of the pipeline model, as the extraction part is perfect.

BERT: We select the same BERT described in Chapter 3 as a candidate PLM that entangles the extraction and reasoning steps. In this model, the input of the “question+story” is passed to the BERT, and the [CLS] representation is used for the answer classification.

GPT-3.5: GPT-3.5 [8] baselines (GPT-3.5 text-davinci-003) is selected as a candidate of generative larger language models which already passes many SOTAs in reasoning tasks [5, 53]. We use *Zero_shot* and *Few_shot* (In-context learning with few examples) settings to evaluate this model on the human-generated benchmarks. We also evaluate the Chain-of-Thoughts (CoT) prompting method [142] to extend the prompts with manually-written reasoning steps. We provide more details of the performance of LLMs on spatial language understanding tasks in Chapter 7.

When applicable, we further train models on SPARTUN synthetic data shown by “*”. We use the huggingFace⁷ implementation of pretrained BERT base models, which have 768 hidden dimensions. All models are trained on the training set, evaluated on the dev set, and reported the result on the test set. For training, we train the model until no changes happen on the dev set and then store and use the best model on the dev set. We use AdamW ([72]), and learning rates from 2×10^{-6} , 2×10^{-5} (depends on the task and datasets) on all models and modules. We used the same configuration and setting for the extraction modules and the baselines as previous works [89]. For SREQA models we use learning rates of 2×10^{-5} , 4×10^{-6} for SREQA(story) and SREQA(question) respectively. To run the models, we use a machine with an Intel Core i9-9820X (10 cores, 3.30 GHz) CPU and Titan

⁷https://huggingface.co/transformers/v2.9.1/model_doc/bert.html

RTX with NVLink as GPU.

For GPT-3.5, we use Instruct-GPT, *davinci-003*⁸. The cost for running GPT-3.5 on the human-generated benchmarks was 0.002\$ per 1k tokens. All codes are publicly available at <https://github.com/RshNk73/PistaQ-SREQA>.

6.5 Results and Discussion

Here, we discuss the influence of disentangling extraction and reasoning manifested in PISTAQ and SREQA models compared to various end-to-end models with no explicit design for these modules, such as BERT, BERT-EQ, and GPT-3.5. Table 6.3 shows the list of these models with the sources of their supervision as well as extra off-the-shelf or rule-based modules employed in them.

Model	Supervisions	Rule-based Modules
BERT	QA	-
GPT-3.5 ^{zero_shot}	-	-
GPT-3.5 ^{few_shot}	QA(8 ex)	-
GPT-3.5 ^{few_shot} +CoT	QA(8 ex) + CoT	-
BERT-EQ	QA +SpRL(S)	-
SREQA	QA +SpRL(all)+Coref	Reasoner, R-Coref
SREQA*	QA + SpRL(Q)	R-Coref
PISTAQ	SpRL(all) + Coref	Reasoner, R-Coref
PISTAQ ^{zero_shot}	-	Reasoner, R-Coref

Table 6.3 The list of annotations from the target benchmarks and rule-based modules employed in each model. We use a quarter of SpRL annotations to train the modules on auto-generated benchmarks. S: Stories, Q: Questions, All: Stories+Questions.

Since the performance of extraction modules, Spatial Role Labeling (SpRL) and Coreference Resolution (Coref), directly contribute to the final accuracy of the designed models, we have evaluated these modules and reported the results in Table 6.4. We choose the best modules on each dataset for experiments. For a detailed discussion on the performance of these modules, see Section 6.5.3.

6.5.1 Result on Controlled Environment

Table 6.5 shows the performance of models on two auto-generated benchmarks, SPARTUN and SPARTQA-AUTO. We can observe that PISTAQ outperforms all PLM baselines and SREQA.

⁸from <https://beta.openai.com>

Dataset	Coref	SRole	SRel	SType
mSpRL	-	88.59	69.12	19.79
mSpRL*	-	88.03	71.23	23.65
HUMAN	82.16	55.8	S: 57.43 Q: 52.55	43.79 39.34
HUMAN*	81.51	72.53	S: 60.24 Q: 61.53	48.74 48.07
SPARTQA	99.83	99.92	S: 99.72 Q: 98.36	99.05 98.62
SPARTUN	99.35	99.96	S: 99.18 Q: 97.68	98.57 98.11

Table 6.4 Performance of the extraction modules. Q: question. S: stories. HUMAN: SPARTQA-HUMAN. SPARTQA: SPARTQA-AUTO. *Further pretraining modules on SPARTUN. We report macro F1 for SpRL and the accuracy of the Coref modules.

#	Models	SPARTUN		SPARTQA-AUTO	
		YN	FR	YN	FR
1	Majority baseline	53.62	14.23	51.82	44.35
2	GT-PISTAQ	99.07	99.43	99.51	98.99
3	BERT	91.80	91.80	84.88	94.17
4	BERT-EQ	90.71	N/A	85.60	N/A
5	SREQA	88.21	83.31	85.11	86.88
6	PISTAQ	96.37	94.52	97.56	98.02

Table 6.5 Results on auto-generated datasets. We use the accuracy metric for both YN and FR questions.

This outcome first highlights the effectiveness of the extraction and symbolic reasoning pipeline compared to PLMs in addressing deterministic reasoning within a controlled environment. Second, it shows that disentangling extraction and reasoning as a pipeline works better than explicit neural layers in SQA with a controlled environment. The complexity of these environments is more related to conducting several reasoning steps and demands accurate, logical computations where a rule-based reasoner excels. Thus, the result of PISTAQ with a rule-based reasoner module is also higher than SREQA with a neural reasoner.

The superior performance of PISTAQ over BERT suggests that SpRL annotations are more effective in the PISTAQ pipeline than when utilized in BERT-EQ in the form of QA supervision. Note that the extraction modules of PISTAQ achieve perfect results on auto-generated benchmarks while trained only on a quarter of the SpRL annotations as shown in Table 6.5. However, BERT-EQ uses all the original dataset questions and extra questions created from the full SpRL annotations.

#	Models	YN	FR		
		Acc	P	R	F1
1	Majority baseline	52.44	29.87	14.28	6.57
2	GT-PISTAQ	79.72	96.38	66.04	75.16
3	BERT	51.74	30.74	30.13	28.17
4	BERT*	48.95	60.96	49.10	50.56
5	GPT-3.5 ^{Zero_shot}	45.45	40.13	22.42	16.51
6	GPT-3.5 ^{Few_shot}	60.13	45.20	54.10	44.28
7	GPT-3.5 ^{Few_shot} +CoT	62.93	57.18	37.92	38.47
8	BERT-EQ	50.34	-	-	-
9	BERT-EQ*	45.45	-	-	-
10	SREQA	53.23	15.68	13.85	13.70
11	SREQA*	46.96	18.70	25.79	24.61
12	PISTAQ	75.52	72.11	35.93	46.80

Table 6.6 Results on SPARTQA-HUMAN. We use accuracy on YN questions and average Precision (P), Recall (R), and Macro-F1 on FR question types. *Using SPARTUN supervision for further training.

Table 6.6 demonstrates the results of models on SPARTQA-HUMAN with a controlled environment setting. As can be seen, our proposed pipeline, PISTAQ, outperforms the PLMs by a margin of 15% on YN questions, even though the extraction modules, shown in Table 6.4, perform low. This low performance is due to the ambiguity of human language and smaller training data. We also evaluate PISTAQ on SPARTQA-HUMAN FR questions using Macro_f1 score on all relation types. PISTAQ outperforms all other baselines on FR questions, except for BERT*.

There are two main reasons behind the inconsistency in performance between YN and FR question types. The first reason is the complexity of the YN questions, which goes beyond the basics of spatial reasoning and is due to using quantifiers (e.g., all circles, any object). While previous studies have demonstrated that PLMs struggle with quantifiers [91], the reasoning module in PISTAQ can adeptly handle them without any performance loss. Second, further analysis indicates that PISTAQ predicts ‘No’ when a relationship is not extracted, which can be correct when the answer is ‘No’. However, in FR, a missed extraction causes a false negative, which decreases the F1 score.

6.5.2 Results on Real-world Setting

We select RESQ as an SQA dataset with realistic settings and present the result of models on this dataset in Table 6.7.

To evaluate PISTAQ on RESQ, we begin by adapting its extraction modules through training on

#	Models	Accuracy
1	Majority baseline	50.21
2	BERT	57.37
3	BERT* <i>Zero_shot</i>	49.18
4	BERT*	63.60
5	GPT-3.5 ^{<i>Zero_shot</i>}	60.32
6	GPT-3.5 ^{<i>Few_shot</i>}	65.90
7	GPT-3.5 ^{<i>Few_shot</i>} +CoT	67.05
8	BERT-EQ	56.55
9	BERT-EQ* <i>Zero_shot</i>	51.96
10	BERT-EQ*	61.47
11	SREQA	53.15
12	SREQA* <i>Zero_shot</i>	53.32
13	SREQA*	69.50
14	PiSTAQ ^{mSpRL}	41.96
15	PiSTAQ ^{SPARTUN+mSpRL}	47.21
16	Human	90.38

Table 6.7 Result on RESQ. *Further training on SPARTUN. The *Zero_shot* refers to evaluation without further training on RESQ or mSpRL training data.

the corresponding dataset. We train the SpRL modules on both mSpRL and SPARTUN, and the performance of these models is presented in Table 6.4. As the mSpRL dataset lacks coreference annotations, we employ the model trained on SPARTUN for this purpose. Rows 14 and 15 in Table 6.7 show the performance of the PiSTAQ on RESQ is inferior compared to other baselines. We analyze the first 25 questions from the RESQ test set to find the reason. We find that 18 out of 25 (72%) questions required spatial commonsense information and cannot be answered solely based on the given relations in the stories. From the remaining seven questions, only two can be answered using the SpRL annotations provided in the mSpRL dataset (8%). Our analysis shows that the SpRL annotations of mSpRL are not fully practical in PiSTAQ due to three main reasons:

1. **No Commonsense:** Even though the commonsense knowledge is required to answer PiSTAQ examples, this information is out of the scope of annotations in mSpRL dataset.
2. **Missed annotations:** As shown in Figure 6.5, some of the prepositions were not considered as spatial concepts by the SpRL annotators but were needed by our reasoning procedure for finding the answers, e.g., NTPP(bar, with, chair).
3. **No coreference :** The coreference annotation is not supported in this dataset, e.g., “L2: it” and “T2: a bench” are the same entity with different mentions, but they are mentioned with

different ids. These missed coreferences result in fewer connections between entities and fewer inferred relations.

Story: behind it a bar with chairs and two people , and a bench with one person lying on it . Upper level with doors and a blue rail.	
mSpRL annotation: Triplet: Behind (a bar (id: t1), behind, it (id: l1)) Triplet: Behind (a bench (id: t2), behind, it (id: l1)) Triplet: EC (one person (id: t3), on, it (id: l2))	
Question 1: Are the people behind the bar?	Answer: Yes
Predicted answer based on mSpRL annotations: No	
Question 2: Is the door above the bar?	Answer: Yes
Predicted answer based on relation in text: No	
Predicted answer based on the commonsense (upper level is above the main level): Yes	

Figure 6.5 An example of the limitation of mSpRL and coreference annotation to answer REsq question. The answer to the questions was predicted wrongly due to two main reasons. First, the missed commonsense knowledge in question 2, and second, the limited coverage of ground truth annotation in mSpRL in question 2.

Hence, the low performance of PISTAQ is attributed to first the absence of integrating commonsense information in this model and, second, the errors in the extraction modules, which are propagated to the reasoning modules.

As shown in Table 6.7, the best result on REsq is achieved by SREQA* model. Compared to SREQA, SREQA* is trained on SPARTUN instead of mSpRL⁹ in the first step of the training. mSpRL lacks some SpRL and coreference annotations to answer REsq questions. In the absence of this information, collecting the supervision for the first phase of training results in a significant number of missed relations. Therefore, as shown in row 11 of Table 6.7, employing mSpRL in the first training phase decreases the performance while replacing it with SPARTUN in SREQA* significantly enhances the results.

SREQA* surpasses the PLMs trained on QA and QA+SpRL annotation, showcasing the advantage of the design of this model in utilizing QA and SpRL data within explicit extraction layers and the data preprocessing. Also, the better performance of this model compared to PISTAQ

⁹As mentioned, we use the mSpRL annotation for REsq dataset.

demonstrates how the end-to-end structure of SREQA can handle the errors from the extraction part and also can capture some rules and commonsense knowledge from RESQ training data that are not explicitly supported in the symbolic reasoner.

In *conclusion*, compared to PLMs, disentangling extraction and reasoning as a pipeline indicates superior performance in deterministic spatial reasoning within controlled settings. Moreover, explicitly training the extraction module proves advantageous in leveraging SpRL annotation more effectively compared to using this annotation in QA format in the end-to-end training. Comparison between disentangling extraction and reasoning as a pipeline and incorporating them within an end-to-end model demonstrates that the end-to-end model performs better in realistic domains even better than PLMs. The end-to-end architecture of this model effectively enhances the generalization in the real-world setting and addresses some of the limitations of rule coverage and commonsense knowledge.

6.5.3 Evaluating Extraction and Reasoning Modules

Here, we discuss each module used in PISTAQ and their performance, including the *Spatial Role Labeling* (SpRL), *Coreference Resolution*, and *Spatial reasoner*.

6.5.3.1 Spatial Role Labeling (SpRL)

The SpRL module is divided into three sub-modules, namely, spatial role extraction (SRole), spatial relation extraction (SRel)¹⁰, and spatial type classification (SType). We only use these modules on sentences that convey spatial information in each benchmark. To measure the performance of SpRL modules, we use the macro average of F1 measure for each label. These modules are evaluated on three datasets that provide SpRL annotations, mSPRL, SPARTQA, and SPARTUN. When training the SpRL module on auto-generated benchmarks, we achieved a performance of 100% using only a quarter of the training data, therefore, we stopped further training.

As shown in Table 6.4, all SpRL sub-modules achieve high performance on synthetic datasets, SPARTQA and SPARTUN. This good performance is because these datasets may contain less

¹⁰Since the questions(Q) and stories(S) have different annotations (questions have missing roles), we separately train and test the SRel and SType modules

ambiguity in the natural language expressions. Therefore, the BERT-base models can easily capture the syntactic patterns needed for extracting the roles and direct relations from the large training set.

6.5.3.2 Coreference Resolution (Coref) in Spatial Reasoning

To evaluate the coreference resolution module (Coref in Table 6.4), we compute the accuracy of the pairs predicted as Corefs. The Coref model achieves a high performance on all datasets. The performance is slightly lower on the SPARTQA-HUMAN dataset when SPARTUN is employed for additional pre-training. However, we observe many errors in the annotations in SPARTQA-HUMAN, and the pre-trained model is, in fact, making more accurate predictions than what is reflected in the evaluation.

6.5.3.3 Logic-based Spatial Reasoner

Datasets	Q-TYPE	Total	A	C	R
SPARTQA-AUTO	YN	18	7	10	1
	FR	38	5	20	13
SPARTUN	YN	13	4	9	0
	FR	35	0	35	0
SPARTQA-HUMAN	YN	29	20	6	3

Table 6.8 Analyzing wrong predictions in GT-PISTAQ. A: Missing/errors in Annotation, C: rule-based Coreference issues in connecting extracted information, R: Shortcomings of the Reasoner.

To solely evaluate the performance of the logic-based reasoner, we use the “GT-PISTAQ”. We look into the errors of this model and categorize them based on the source of errors. The categories are *missing/wrong ground truth direct annotations* (A), *rule-based Coreference Error* (C) in connecting the extracted information before passing to the reasoner, and *the low coverage of spatial concepts in the reasoner* (R). As is shown in Table 6.8, spatial Reasoner causes no errors for SPARTUN since the same reasoner has been used to generate it. However, the reasoner does not cover spatial properties of entities (e.g., right edge in “touching right edge”) in SPARTQA and causes wrong predictions in those cases.

6.5.4 SREQA on All Story Relations

Table 6.9 displays the results of the SREQA model trained and tested solely on all the story’s relation extraction parts (step 1). During the evaluation, we also possess the same data preprocessing

Datasets	F1 on SREQA
SPARTUN	96.37
SPARTQA-AUTO	97.78
SPARTQA-HUMAN	23.79
mSPRL (Used in RESQ)	16.59

Table 6.9 The result of SREQA model only trained and tested on all story relations of the SQA datasets.

and gather annotations of all relations between stories’ entities and select the best model based on performance on the development set.

Notably, the performance on the human-generated datasets, SPARTQA-HUMAN and RESQ, is significantly lower compared to the auto-generated datasets. As discussed in , the mSPRL datasets contain missed annotations, resulting in the omission of several relations from the stories’ entities and incomplete training data for this phase. Similarly, the SPARTQA-HUMAN SpRL annotation also exhibits some noise, particularly in coreference annotation, leading to similar issues as observed in mSPRL regarding annotation of all story relations.

Consequently, this reduced performance in all story relation extraction impacts the overall performance of the main SREQA model trained using two steps; however, as illustrated in the results of SREQA* in Table6.7, which utilizes SPARTUN instead of mSPRL for training on all story’s relations, the performance substantially improves on the RESQ dataset.

6.6 Summary

We investigate the benefits of disentangling the processes of extracting spatial information and reasoning over them [112]. To this end, we devised a series of experiments utilizing PLMs for spatial information extraction coupled with a symbolic reasoner for inferring indirect relations. The outcomes of our experiments provide noteworthy insights: (1) Our observations in controlled experimental conditions demonstrate that disentangling extraction and symbolic reasoning, compared to PLMs, enhances the models’ reasoning capabilities, even with comparable or reduced supervision. (2) Despite the acknowledged fragility of symbolic reasoning in real-world domains, our experiments highlight that employing explicit extraction layers and utilizing the same symbolic reasoner in data preprocessing enhances the reasoning capabilities of models. These results emphasize the advantage

of disentangling the extraction and reasoning in spatial language understanding.

CHAPTER 7

EXPLORING SPATIAL REASONING WITH LARGE LANGUAGE MODELS

7.1 Introduction

Recent research highlights the high performance of Large Language Models (LLMs) on numerous tasks using Zero_shot [5] or Few_shot prompting, a.k.a in-context learning, methods [142]. Zero_shot prompting refers to the ability of a model to perform a task without any task-specific training examples, relying solely on its pretrained knowledge. On the other hand, Few_shot prompting involves providing the model with a small number of task-specific examples to help it adapt to the task and improve its performance. These prompting methods demonstrate the LLMs' ability to generalize and adapt to various problems with minimal task-specific information [10, 8] and perform well even without extensive training data for specific tasks.

However, despite their success in many domains, LLMs have limitations on multi-hop reasoning tasks [41, 131]. For instance, [5] indicates that ChatGPT (GPT-3.5-Turbo) struggles to perform well on the SQA (multi-hop Spatial Question Answering) task when evaluated using Zero_shot settings on human-generated test cases. This highlights the challenges LLMs face in reasoning tasks that require more than just pattern recognition and necessitate a deeper understanding of the context and relationships within the data. To address these challenges, new prompting techniques such as Chain-of-Thought (CoT) [142] or Least-to-Most prompting [159] have been introduced. These promptings involve instructing the LLM to generate a step-by-step solution or explanation for a given problem before arriving at the final answer, encouraging the model to think more systematically. By leveraging this method, LLMs can overcome some of the limitations observed in Few_shot evaluations and achieve better performance on complex multi-hop reasoning tasks.

In this chapter, we first evaluate LLMs on SQA tasks using Few_shot and Chain-of-Thought (CoT) prompting strategies and compare them with Zero_shot evaluation. Afterward, we introduce an approach in multi-hop reasoning by leveraging the superior performance of LLMs in information extraction tasks [137, 135, 146]. Specifically, we employ an LLM as the extraction module within the proposed pipeline model of extraction and symbolic reasoning, detailed in Chapter 6. As

discussed in Section 6.5.1, the efficacy of this pipeline is prominent when the extraction module functions optimally. Hence, our objective in this experiment is to evaluate the impact of LLMs as a potent extraction module within this framework.

```
"('two single beds', 'a pillow')": {  
  "relation_lists": [  
    "covers"  
  ],  
  "probability": {  
    "covers": 0.21900160947318847,  
    "near": 0.17055862495161717,  
    "front": 0.12478334318791685,  
    "left": 0.10026623307188026,  
    "above": 0.09871174921121996  
  }  
},
```

Figure 7.1 An example of the GPT-3.5 output for a pair of entities. GPT-3.5 output the first one with the highest probability as the final output.

When incorporating language models for information extraction within the extraction and reasoning pipeline, we disregard the inherent uncertainty in their predictions and instead select the predictions based on the highest probability (uncertainty score). However, incorporating this uncertainty can enable a more informed and context-aware utilization of language model outputs. Hence, in another line of experiments, we aim to evaluate LLMs in the same extraction and symbolic reasoning pipeline while employing the uncertainty of their prediction in probabilistic logical reasoning rather than a deterministic method. In probabilistic reasoning, first, we select the top five spatial relation predictions of LLM, along with corresponding probability values, for each pair of entities (e.g., Figure 7.1). Next, all this pertinent information is conveyed to a probabilistic reasoner to address the SQA task.

As discussed in Section 6.5.2, another challenge associated with using symbolic reasoning in the pipeline model is its weakness in covering all existing spatial rules, including real-world commonsense knowledge, e.g., the concept that a roof is always above a house. Thus, in our last experiment, we capitalize on the comprehensive commonsense knowledge of LLMs about general space. In this experiment, we utilize LLMs to extract spatial commonsense knowledge that may not be explicitly present in the text. This information is passed to a probabilistic spatial reasoner

alongside the direct information to solve the SQA task. This experiment highlights the potential of LLMs to provide a more holistic understanding of the spatial world used in the pipeline of extraction and reasoning.

The outcomes of our experimental evaluation demonstrate the following insights:

- LLMs exhibit better spatial reasoning performance when utilizing Few_shot or CoT prompting compared to Zero_shot evaluation. However, this performance still is less than human performance in this task.
- Our case studies and experimental results from concurrent research in controlled SQA environments reveal the impact of LLMs as extraction modules in providing precise information within the pipeline of extraction and reasoning. This pipeline model outperforms SOTA baselines, particularly LLMs, with in-context learning. However, our exploration of real-world SQA tasks unveils a different outcome. The complexity inherent in spatial language hinders the proficiency of LLMs in extracting spatial information from real-world text. Hence, on real-world SQA task, using LLMs in the pipeline model demonstrates inferior performance compared to end-to-end LLMs. Despite this observation, our experiments provide valuable insights:

1. Employing LLM as an extraction module in this pipeline provides more precise information for symbolic spatial reasoning compared to BERT-based modules.
2. Utilizing LLM’s probabilistic outputs in probabilistic logical reasoning leads to better results than deterministic reasoning.
3. Leveraging LLMs’ commonsense knowledge can enhance the pipeline model’s performance by enclosing some rules and relations that the designed symbolic reasoning module may not cover.

7.2 Large Language Models (LLM)

In this study, we select GPT-3.5 as a candidate of LLMs and use the framework developed by OpenAI as the foundation for our experiments. GPT-3.5, with 175 billion parameters, is a state-of-the-art LLM that boasts several advantages over other available open-source LLMs. GPT-3.5 outperforms LLaMA and PaLM on MMLU and GSM8K benchmarks as shown in Figure 7.2 [129].

This suggests that GPT-3.5 has a slight edge over LLaMA and PaLM performance on question answering task. GPT4 already surpasses all results, but we limited our usage of GPT4 to a small set of test examples due to budgetary constraints. One of the primary benefits of GPT-3.5 is its

Benchmark (shots)	GPT-3.5	GPT-4	PaLM	PaLM-2-L	LLAMA 2
MMLU (5-shot)	70.0	86.4	69.3	78.3	68.9
TriviaQA (1-shot)	–	–	81.4	86.1	85.0
Natural Questions (1-shot)	–	–	29.3	37.5	33.0
GSM8K (8-shot)	57.1	92.0	56.5	80.7	56.8
HumanEval (0-shot)	48.1	67.0	26.2	–	29.9
BIG-Bench Hard (3-shot)	–	–	52.3	65.7	51.2

Figure 7.2 Comparison of different LLMs on Natural Language processing benchmarks.

extensive knowledge base, which encompasses a wide range of topics and domains. This extensive knowledge allows the model to provide more accurate and contextually relevant responses even with Few_shot prompting, making it particularly effective for our tasks.

7.2.1 OpenAI API

The OpenAI API provides an interface for interacting with OpenAI’s Large Language Models, including GPT-3.5 and GPT-4. To use this API, we send a request to the API endpoint, passing the prompt and any desired parameters as part of the request body. The API will return a JSON response containing the model’s output. The following provides a list of parameters we employed to customize the OpenAI framework in our experiments.

1. **Prompt:** This parameter specifies the input prompt for the model. It can be a string or a list of strings, and it sets the context for the model’s response.
2. **Temperature:** This parameter controls the randomness of the model’s output. A higher value (e.g., 0.8) makes the output more random and creative, while a lower value (e.g., 0.2) makes it more deterministic and focused.
3. **Max_tokens:** This parameter sets the maximum length of the generated text to prevent overly long outputs.
4. **Log_probs:** This parameter specifies the number of most likely tokens for which the model should return the log probabilities.

5. **Top_k**: This parameter selects the top-k most probable tokens. It controls the diversity of the output by restricting the set of tokens considered during generation.

We use GPT-3.5 *davinci-text-003*¹ in our experiments of evaluating LLMs on spatial reasoning task. The cost of running this model on the human-generated benchmarks is 0.002\$ per 1k tokens. To use GPT-3.5 as information extraction, we use *turbo-text-instruct* (a.k.a ChatGPT) with a cost of 0.0001\$ per 1k tokens. We also use the GPT4 playground in our experiments.

7.2.2 Prompt Engineering

Selecting and providing pertinent prompts is crucial, as the prompt can significantly influence the model’s performance. We employ various prompting techniques in our experiments, including Zero_shot, Few_shot, and Chain-of-Thought (CoT) prompting, to evaluate the model’s performance under different conditions.

In the Zero_shot setting, the prompt given as input to the model is formatted as “Context: story. Question: question?” and the model returns the answer to the question. In the Few_shot setting, we add two random examples from the training data containing the story, all its questions, and their answers. Figure 7.3 depicts a prompt example for SPARTQA-HUMAN YN questions passed to GPT-3.5.

In Few_shot+CoT, we use the same idea as [142] and manually write the reasoning steps for eight questions (from two random stories). The input then is formatted as “Context: story. Question: CoT. Answer. Asked Context: story. Question: question?”. Figure 7.4 shows an example of these reasoning steps for RESQ dataset.

7.2.3 Engineering Over LLM Output

Output engineering is essential in extracting meaningful information from the LLM’s output. In our experiments, we implement various output processing techniques, such as filtering and ranking over the model’s predictions. This step transfers the raw output of the LLM into structured data that can be used for final evaluation or as input to other modules. Given that LLMs excel at imitating observed formats, we instructed them to generate predictions in a structured format, as specified in

¹from <https://beta.openai.com>

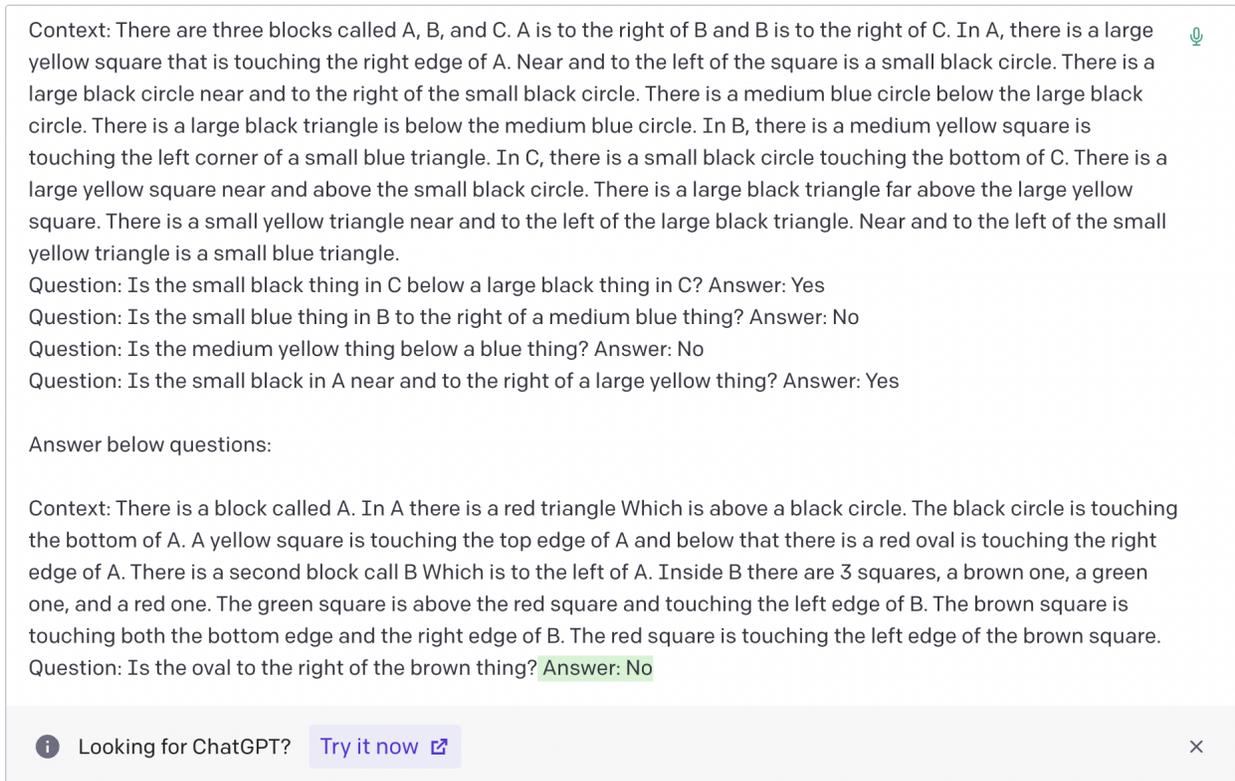


Figure 7.3 Example of the input for GPT-3.5 (Few_shot). The Zero_shot setting is the same just do not have the first training example.

the prompt (e.g., illustrated in Figure 7.5). This approach significantly simplified the engineering process over the string output of LLM.

7.3 Probabilistic Spatial Reasoning

As discussed in Chapter 6, one of the primary reasons for the lower performance of the PISTAQ model on real-world data is the error propagation from the information extraction modules to the reasoning module. One potential solution to address this issue is to pass the uncertain information by including the probabilities of the extracted information from LLM to a probabilistic reasoner instead of a deterministic logical reasoner. Employing deterministic reasoning has two primary drawbacks. First, the probability of the predicted token with the highest probability may not be significantly high; for example, in Figure 7.1, the probability of “covers” is 0.219, while the probability of the second-best token, “near” is 0.17. Ultimately, the model chooses “covers” as the final output while “near” has a close probability. Second, the next best predictions might also offer valuable insights.

Chain-of-Thoughts Reasoning Steps.

Consider Below examples:

There are three blocks called A, B, and C. A is to the right of B and B is to the right of C. In A, there is a small blue square far to the right of a medium black circle. In B, there is a large black circle far below a medium yellow circle. The medium yellow circle is far below a medium black triangle. The medium black triangle is near and below a small blue square. In C, there is a large yellow square near and above a medium blue square. There is also a medium yellow square far to the left of the large yellow square.

Is the medium blue thing near and below a large yellow thing?

The chain of reasoning to answer this question is: The large yellow square is near and above a medium blue square. Using the inverse rule, the medium blue square is near and below the large yellow square. The answer is Yes.

Is the small blue thing in B below a medium yellow thing?

The chain of reasoning to answer this question is: The small blue circle in B is above the black triangle which is above the medium yellow circle. So the small blue thing is above a medium yellow thing. The answer is No.

Is the small blue thing in A to the right of a medium blue thing?

The chain of reasoning to answer this question is: block A is right of block B and block B is right of block C, so block A is right of block C. The medium blue thing is in block C. All objects in block A are right of objects in block C, so the small blue thing in A is right of the medium blue thing in C. The answer is Yes.

Is the medium black thing in A above a small blue thing?

The chain of reasoning to answer this question is: There is a small blue thing in block A and block B. It is not mentioned if block B is below or above block A or not. Also it is not mentioned if medium black thing in A is above the small blue thing in A. So the answer is Don't Know(No).

Answer below questions:

Figure 7.4 Example of the input for GPT-3.5(Few_shot+Cot) with human-written Chain-of-Thoughts.

For example, in Figure 7.1, if we inquire whether “pillow” is near the “beds”, considering only the top-1 token prediction, the probability of the “near” is 0, and the final answer is “No”. However, if we select the top-2 prediction, the probability of “near” becomes 0.17, resulting in the “Yes” answer. Hence, employing the top-k (here we select k= 5) predictions of the LLMs with their probability value helps to solve the SQA task efficiently.

To implement the probabilistic logical reasoner, we leverage probabilistic Prolog, known as Problog[17]. **Problog** is a probabilistic logic programming language that extends traditional logic programming with the ability to reason about uncertainty and probability. It allows users to write programs that define relationships between events or conditions and assign probabilities to those relationships. In Problog, probabilities can be assigned to logical rules, and queries can be made about the probability of different events or conditions given a set of evidence or assumptions. In Figures 7.6 and 7.7, you can see examples of predictions with and without probability computation. The 0.8 and 0.6 are the predicted probabilities of spatial relation extraction, and the rules are directly from the proposed spatial reasoner in Chapter 4. Problog computes the probabilities of the query based on all possible ways the query can be true. As a result, $above(a, c)$ can be *True* with 0.48

Extraction		Prompt Example
Story	Entity	Context 1: there are two social workers in the foreground . one wearing a red pullover and brown pants is bending over to access the blue paint . the other one in a red tee-shirt and black jeans is watching her . at the back of the room there is another worker wearing a white tee-shirt and blue jeans , acutally painting . there are many newspapers on the ground to protect the kindergarten floor . all three workers are wearing a mask . entities = ['two social workers', 'the foreground', 'one', 'a red pullover', 'brown pants', 'the blue paint', 'the other one', 'a red tee-shirt', 'black jeans', 'her', 'the back', 'the room', 'another worker', 'a white tee-shirt', 'blue jeans', 'many newspapers', 'the ground', 'the kindergarten floor', 'three workers', 'a mask']
	Relation	- a man in white shirt , black jacket , grey pants and black shoes is sitting on a wooden chair and talking on the phone. relation triplets: [("a man", "in", "white shirt"), ("a man", "in", "black jacket"), ("a man", "in", "grey pants"), ("a man", "in", "black shoes"), ("A man", "sitting on", "a wooden chair"), ("a man", "talking on", "the phone")] - on the right a wooden bed with white bedcovers. relation triplets: [("a wooden bed", "on", "the right"), ("a wooden bed", "with", "white bedcovers")]
	Relation Type	If the relation set is: Relation type set: LEFT = to the left of another object, DC= Disconnected, disconnected from other object, (the wall, behind, the tourists), relation type is: ['BEHIND'] (Lots of locals, in front of, a blue building) , relation type is: ['FRONT'] (pictures, on, the wall), relation type is: ['FRONT', 'EC'] (a clock, above, the blackboard), relation type is:['ABOVE']
	Coreference	Context 1: Three women are sitting on a wooden bench in front of an about one metre high , red brick wall . they are all wearing skirts and jumpers ... If the list of all entities is: list_of_noun_phrases = ["three women", "they", "two of them", "a wooden bench", "an about one metre high red brick wall", "the wall", "skirts", ...] The below list shows which noun phrases in the "list_of_noun_phrases" refers to which same phrase: {"Three women": ["Three women", "they", "two of them"], they: ["Three women", "they", "two of them"], two of them: ["two of them"], a wooden bench: ["a wooden bench",],}
Question	Relation and Type	If the relation set is: Are the lamps behind the building?: list_of_dictionary = [{"triplet": ("the lamps", "behind", "the building"), "relation type": ["BEHIND"]}] Is the camera in front of the all kids?: list_of_dictionary = [{"triplet": ("the camera", "in front of", "the all kids"), "relation type": ["FRONT"]}] Is a flag to the left of the stairs?: list_of_dictionary = [{"triplet": ("a flag", "to the left of", "the stairs"), "relation type": ["LEFT"]}]
Question entity to Story Mentions		This should consider the exact or partially matching based on the phrase root. For examples,"{0: "small window", 2: "large window", 5: "three windows"}" all can be matched with "the window" since the root is window here. Also the output should be in the form of only a python dictionary like {"the window": [0,2,5]}.

Figure 7.5 The example of prompts used for LLMs (GPT-3.5-Turbo) in information extraction.

probability.

7.3.1 Extract probabilistic Facts From LLM

To extract the probabilities of tokens from the GPT-3.5, we use the “log_probs” parameter in the API, which specifies the number of most likely tokens for which the model returns the

```

1 above(a,b).
2 below(c,b).
3
4
5 below(X,Y) :- above(Y,X).
6 above(X,Y) :- below(Y,X).
7 above(X,Y) :- above(X,Z), above(Z,Y).
8
9 query(above(a,c)).
10
11 probability = 1

```

Figure 7.6 The result of Problog without any probabilistic prediction.

```

1 0.8::above(a,b).
2 0.6::below(c,b).
3
4
5 below(X,Y) :- above(Y,X).
6 above(X,Y) :- below(Y,X).
7 above(X,Y) :- above(X,Z), above(Z,Y).
8
9 query(above(a,c)).
10
11 probability = 0.48

```

Figure 7.7 The result of Problog with probabilistic facts.

log probabilities. Following this, from the “log_probability” output of the LLM, we identify additional prediction alternatives and their associated probabilities for the relation’s token as shown in Figure 7.8. Besides, to identify the relation token from the LLM output, a specific pattern was defined within the prompt of the GPT-3.5. For instance, as shown in Figure 7.9, the relation tokens are always inside brackets.

7.4 Experiments and Results

In this section, we evaluate GPT-3.5 on two human-generated SQA benchmarks, SPARTQA-HUMAN and RESQ. We compare the performance of GPT-3.5 against a suite of models including BERT, BERT*², PISTAQ, and SREQA described in Chapter 6. Both BERT and BERT* serve as candidate PLMs in which the input of “question+story” is passed to a BERT with answer classification layers. The PISTAQ model is a pipeline that disentangles extraction and symbolic reasoning, while SREQA is an end-to-end neural model featuring explicit layers for extraction

²BERT further fine-tuned on SPARTUN supervisions.

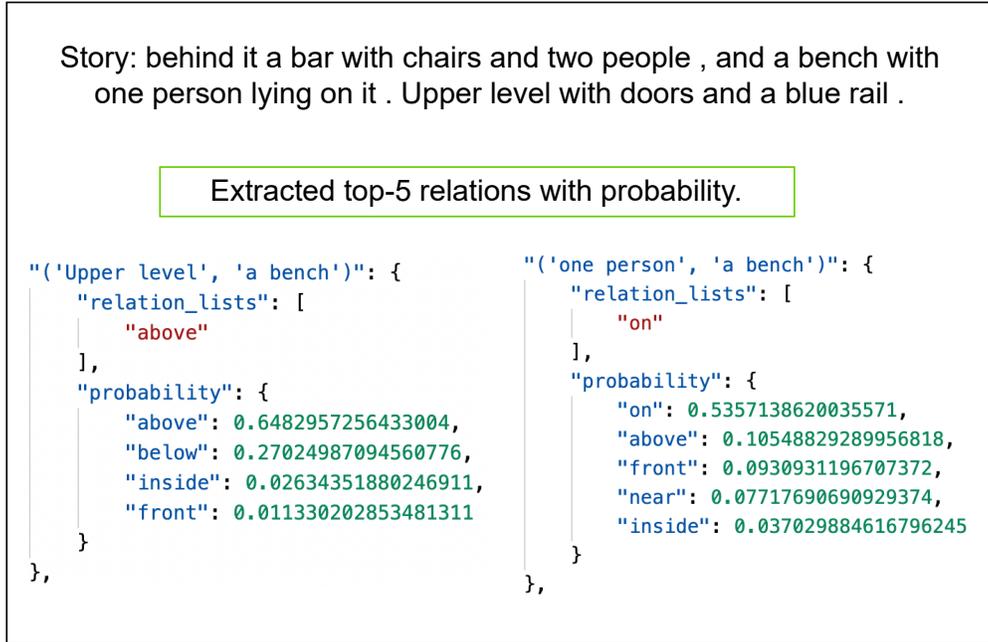


Figure 7.8 Example of probability of top-5 tokens extracted from the GPT-3.5.

and reasoning. Additionally, we assess the performance of using LLM as the extraction module within PISTAQ, referred to as LLM_PISTAQ. In cases where the probabilistic output of the LLM is integrated into this model, we call the model LLM_PISTAQ^{prob}.

7.4.1 LLM with Few_shot Prompting

Recent research shows the high performance of LLMs with Zero/Few_shot setting on many tasks [10, 8]. However, [5] shows that ChatGPT (GPT-3.5-Turbo) with Zero_shot evaluation cannot perform well on SPARTQA-HUMAN test cases. Similarly, our experiments, as shown in Tables 6.6 and 6.7, show the lower performance of GPT-3.5 (davinci) with Zero/Few_shot settings compared to human and our models PISTAQ and SREQA. Figure 7.10, shows an example of three LLMs, GPT-3.5, GPT4 and PaLM2 on SPARTQA-HUMAN example. Although [142] shows that using CoT prompting improves the performance of PaLM on multi-step reasoning task, its spatial reasoning capabilities still does not meet the expectation.

7.4.1.1 LLMs as Extraction Module in PISTAQ

Recent studies [117, 137] show that LLMs have a promising performance in information retrieval. Building upon this, first, we employ GPT-3.5 with Few_shot prompting to extract information from

Prompts For Finding The Probability

Let's find out the "direct" spatial relations of trajectory toward the landmark from the given text from below list and return a dictionary:
list_of_spatial_relations = [null, above, below, front, behind, left, right, far, near, disconnected, Externally Connected, overlapped, equal, inside, contains, covered by, covers]

The relation should be directly mentioned in the text and you should find all of them. The entities should be selected from the list of the entities which also are given based on the context.

Examples:

Context1: "tourists are standing in the classroom of a school in front of the blackboard. there is a clock and the writing Dios es Amor above the blackboard."

all_entity_on_the_context = ["tourists", "the classroom of a school", "the blackboard", "a clock", "the writing Dios es Amor"]

Dictionary of only all direct Spatial Relations is:

```
{("tourists", "the classroom of a school"): ["in", "inside"],  
 ("tourists", "the blackboard"): ["front"],  
 ("a clock", "the blackboard"): ["above"],  
 ("the writing Dios es Amor above", "the blackboard"): ["above"]}
```

Be concise and based on the given context and entities list find the direct spatial relation and return a dictionary.

list_of_spatial_relations = [null, above, below, front, behind, left, right, far, near, disconnected, Externally Connected, overlapped, equal, inside, contains, covered by, covers]

Figure 7.9 Example of the pattern repeated in the prompt of the LLM model. Hence, the relation tokens are always inside brackets ('[]'), which helps find the relation tokens.

a set of SPARTQA-HUMAN and RESQ examples. The extraction process contains Entity, Relation, Relation Type, coreference extractions from the story, and entity and relation extraction from the question. Additionally, LLM is employed to identify mentions of question entities within the text. We construct multiple manually crafted prompt examples for each extraction task, as depicted in Figure 7.5. The extracted information is subsequently utilized within the PISTAQ framework called LLM_PISTAQ.

We conduct a case study, illustrated in the last row of Figure 7.11, which highlights how combining LLM extraction and symbolic reasoning enables answering questions that LLMs struggle to address. Furthermore, Figure 7.12 provides a comparison between the trained BERT-based spatial information extraction modules and GPT-3.5 with Few_shot prompting in PISTAQ. GPT-3.5 extracts more accurate information, leading to a correct answer that end-to-end LLMs cannot solve. Furthermore, a concurrent study by Yang et al. (2023) [149] showcases the efficacy of LLMs as extraction modules coupled with symbolic reasoning within a controlled environment of StepGame dataset. As depicted in Figure 7.13, LLM excels in the extraction process when the synthetic

SpaRTOA-Human	Story	There is a block called A. Inside there's a red oval that is above a blue triangle. The blue triangle is touching the right edge of A and the top of a large black square. Touching the bottom edge of A is a small orange square. There is a block called B Which is to the right and touching A. Inside B there are 5 triangles, a red one, a green one, a large yellow one, a brown one, and an orange one. The red and orange triangles are touching the left edge of B and the orange triangle is above the red triangle. The large yellow triangle is touching the top edge of B. The green triangle is touching the right edge of B. The brown triangle is below all of the other triangles and is touching the bottom of B.
	Question 1:	What is the relation between blue triangles and orange triangles? Answer: left, near
	Question 2:	Is the green thing above the yellow thing? Answer: No
	Question 3:	Does B have all of the triangles? Answer: No
GPT3.5 (Prediction)	Zero-shot	Q1: Don't Know Q2: No , there is no green thing mentioned in the description. Q3: No
	CoT	Q1: The blue triangle is touching the right edge of block A and the orange triangle is touching the left edge of block B. Block B is to the right and touching block A so the orange triangle is right of the blue triangle. The answer is [Right] . Q2: The green triangle is to the right of the large yellow triangle and the large yellow triangle is touching the top edge of B. Using inverse rule, the green triangle is below the yellow triangle. The answer is No . Q3: There are five triangles mentioned in block B, a red one, a green one, a large yellow one, a brown one, and an orange one. All of these triangles are in block B, so the answer is Yes .
GPT4 (Prediction)	Zero-shot	Q1: Don't Know Q2: The description does not provide enough information to determine if the green triangle is above the yellow triangle. It only states that the green triangle is touching the right edge of block B and the large yellow triangle is touching the top edge of B. The relative vertical positions of the green and yellow triangles are not specified. Q3: Yes , B has all of the triangles mentioned in the description. Inside block B, there are 5 triangles: a red one, a green one, a large yellow one, a brown one, and an orange one.
	CoT	Q1: The blue triangle is in block A, and the orange triangle is in block B. Block B is to the right and touching block A. So, the blue triangle is left of the orange triangle. The answer is [Left] . Q2: The green triangle is touching the right edge of B, and the large yellow triangle is touching the top edge of B. It is not mentioned if the green triangle is above or below the yellow triangle. The answer is Don't Know . Q3: There is a blue triangle in block A, and there are 5 triangles in block B. Since there is a triangle in block A, not all triangles are in block B. The answer is No .
PaLM2 (Prediction)	Zero-shot	Q1: The answer is near . The blue triangle and the orange triangle are both in the block B. The blue triangle is above the orange triangle. Q2: Yes , the green thing is above the yellow thing. The green thing is a triangle that is touching the right edge of block B, and the yellow thing is a triangle that is touching the top edge of block B. Q3: Yes , B has all of the triangles. The question asks if B has all of the triangles in the block. The answer is yes because the passage says that "Inside B there are 5 triangles, a red one, a green one, a large yellow one, a brown one, and an orange one." This means that B has all of the triangles in the block.
	CoT	Q1: The blue triangle is in block A and the orange triangle is in block B. Block A is left of block B so the blue triangle is left of the orange triangle. The answer is [Left] . Q2: The green triangle is touching the right edge of B and the large yellow triangle is touching the top edge of B. The right edge of B is below the top edge of B. So the green triangle is below the yellow triangle. The answer is No . Q3: There are 5 triangles in B. There is a red triangle in B. There is a green triangle in B. There is a large yellow triangle in B. There is a brown triangle in B. There is an orange triangle in B. The answer is Yes .

Figure 7.10 An example of three LLMs, GPT-3.5 (davinci-text-003), GPT4, and PaLM2 prediction on SQA task evaluated with Zero_shot and CoT prompting.

context follows a fixed pattern³, and consequently, the disentangled model, incorporating GPT-3 for extraction and symbolic reasoning (using Answer Set Programming (ASP)), demonstrates superior performance compared to end-to-end models like BERT* (SynSup) or GPT-3.5+CoT.

In this experiment, we aim to evaluate the efficacy of LLM_PISTAQ on the real-world SQA benchmark, RESQ. Spatial information (SpRL) from real-world data is significantly more complicated than fixed patterns of synthetic data, as it involves natural language's ambiguity and intricacy. For

³As the StepGame dataset lacks explicit SpRL supervision, the evaluation of this model's performance on the extraction task remains unassessed.

SpaRTOA-Human	Story: There is a block called A. Inside there's a red oval that is above a blue triangle ... There is a block called B which is to the right and touching A. Inside B there are 5 triangles, a red one, a green one, a large yellow one, a brown one, and an orange one ... The large yellow triangle is touching the top edge of B. The green triangle is touching the right edge of B...
	Question 1: Is the green thing above the yellow thing? Answer: No
	Question 2: Does B have all of the triangles? Answer: No
GPT3.5 (Prediction)	Q1: The green triangle is to the right of the large yellow triangle and the large yellow triangle is touching the top edge of B. Using inverse rule, the green triangle is below the yellow triangle. The answer is No .
	Q2: There are five triangles mentioned in block B, a red one, a green one, a large yellow one, a brown one, and an orange one. All of these triangles are in block B, so the answer is Yes .
GPT4 (Prediction)	Q1: The green triangle is touching the right edge of B, and the large yellow triangle is touching the top edge of B. It is not mentioned if the green triangle is above or below the yellow triangle. The answer is Don't Know .
	Q2: There is a blue triangle in block A, and there are 5 triangles in block B. Since there is a triangle in block A, not all triangles are in block B. The answer is No .
PaLM2 (Prediction)	Q1: The green triangle is touching the right edge of B and the large yellow triangle is touching the top edge of B. The right edge of B is below the top edge of B. So the green triangle is below the yellow triangle. The answer is No .
	Q2: There are 5 triangles in B. There is a red triangle in B. There is a green triangle in B. There is a large yellow triangle in B. There is a brown triangle in B. There is an orange triangle in B. The answer is Yes .
PiSTaQ	Q2: Extracted Facts (GPT3, Turbo): NTPP(blue triangle, A), TPP(red triangle, left edge of B), TPP(orange triangle, left edge of B), ... Extracted Queries (GPT3, Turbo): NTPPI/TPPI(B, blue triangle), NTPPI/TPPI(B, red triangle), NTPPI/TPPI(B, orange triangle), ... Symbolic Reasoning (Rule-based Reasoner): False & True & True & ... → Answer = No

Figure 7.11 An example of Large Language Models (LLMs) prediction on SQA task evaluated with CoT prompting. The last row shows an example of using GPT3.5-Turbo for information extraction in LLM_PiSTaQ.

Story:	a photo of a room with white walls , two single beds with a night table in between and a picture on the wall above the beds .	
Question:	Are the beds below the picture? Answer: Yes	
Story Facts:	BERT	0: ['a picture', 'the beds'], 2:['a'], 1: ['a picture', 'the wall'] Facts: right(2, 1), below(2, 0), near(2, 0)
	GPT3	3: ['two single beds', 'the beds'], 5: ['a picture'], 6: ['the wall', 'the beds'] Facts: above(5, 3), above(5, 6) ...
Queries:	BERT	below(0, 0)? or below(0, 1)?
	GPT3	below(3, 5)? or below(3, 6)?
Reasoning:	BERT	below(0, 0) = False, below(0, 1) = False → Answer = No
	GPT3	below(3, 5) = True, below(3, 6) = False → Answer = Yes

Figure 7.12 An example of using BERT-based SpRL and GPT-3.5 as information extraction in PiSTaQ on a RESQ example.

example, correctly extracting the relation (a large window, on the left, a room) from the sentence “a room with walls made of wood and a large window with white curtains on the left.” is challenging for even potent LLMs.

We first evaluate the performance of GPT-3.5 in information extraction within the context of RESQ. Since RESQ derives its textual context from the mSpRL dataset, which includes spatial information annotations, we evaluate GPT-3.5 on this dataset. Table 7.1, shows GPT-3.5 performance on entity extraction. The results reveal a higher Recall for GPT-3.5 compared to other models, while

Method	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
RN	22.6	17.1	15.1	12.8	11.5	11.1	11.5	11.2	11.1	11.3
RRN	24.1	20.0	16.0	13.2	12.3	11.6	11.4	11.8	11.2	11.7
UT	45.1	28.4	17.4	14.1	13.5	12.7	12.1	11.4	11.4	11.7
STM	53.4	36.0	23.0	18.5	15.1	13.8	12.6	11.5	11.3	11.8
TPR-RNN	70.3	46.0	36.1	26.8	24.8	22.3	19.9	15.5	13.0	12.7
TP-MANN	85.8	60.3	50.2	37.5	31.3	28.5	26.5	23.7	22.5	21.5
SynSup	98.6	95.0	92.0	79.1	70.3	63.4	58.7	52.1	48.4	45.7
Few-Shot (d3)	55.0	37.0	25.0	30.0	32.0	29.0	21.0	22.0	34.0	31.0
CoT (d3)	61.0	45.0	30.0	35.0	35.0	27.0	22.0	24.0	23.0	25.0
GPT-3(c1)+ASP	44.7	38.8	40.5	58.8	62.4	57.4	56.2	58.0	56.5	54.1
GPT-3(d2)+ASP	92.6	89.9	89.1	93.8	92.9	91.6	91.2	90.4	89.0	88.3

Figure 7.13 Results from [149] over StepGame with different number of steps of reasoning (k). GPT-3+ASP is a model that disentangles the extraction and reasoning. GPT-3 is used for extraction, and ASP is the symbolic reasoner. Synsup is BERT* here.

Entity Extraction (mSPRL)	Recall	Precision	Macro-F1
R-Inf [78]	71.99	75.24	73.56
BERT-Based	88.52	75.89	81.72
BERT-Based (SPARTUN)	89.52	74.65	81.06
GPT-3.5-based	91.63	49.72	64.47

Table 7.1 The result of models on entity extraction task. Since the annotation in mSPRL does not contain all information, recall is considered the most crucial metric.

Relation Extraction (mSPRL)	Recall	Precision	Macro-F1
R-Inf [78]	68.25	70.23	69.23
BERT-Based	81.31	60.11	69.12
BERT-Based (SPARTUN)	89.13	59.32	71.23
GPT-3.5-based	34.09	16.48	22.22

Table 7.2 The result of models on relation extraction task.

the Precision is considerably lower. This discrepancy is attributed to the numerous false-positive predictions that are accurately made by GPT-3.5 but are not covered by mSPRL annotations (Figure 6.5 shows some of these missed annotations in mSPRL.). Hence, GPT-3.5 can extract information that is not even annotated in mSPRL but is required to solve RESQ questions.

Table 7.2 compares GPT-3.5 and other benchmarks on spatial relation extraction task. In this table, GPT-3.5 exhibits inferior performance compared to other baselines; however, it is essential to acknowledge that this result may not precisely reflect the true capabilities of GPT-3.5. Figure 7.14 presents an example of the complexity of evaluating GPT-3.5 extraction performance based on the

```

Story: a brownish red house with a white decorated entrance . there are lamps and windows with white bars on the ground level ,
with a man with a ladder , a car , pedestrians and a street in front of it .

mSpRL annotation:
{'trajector': 'lamps',      'landmark': 'the ground level', 'spatial_indicator': 'on',      'rel_type': 'TPP'},
{'trajector': 'windows',   'landmark': 'the ground level', 'spatial_indicator': 'on',      'rel_type': 'TPP'},
{'trajector': 'a man',     'landmark': 'it',               'spatial_indicator': 'in front of', 'rel_type': 'FRONT'},
{'trajector': 'a car',     'landmark': 'it',               'spatial_indicator': 'in front of', 'rel_type': 'FRONT'}

LLM_Extraction:
{'trajector': 'a brownish red house', 'landmark': 'a white decorated entrance', 'spatial_indicator': 'with',      'rel_type': ['EC', 'TPPI']},
{'trajector': 'lamps and windows',   'landmark': 'the ground level',         'spatial_indicator': 'on',        'rel_type': ['FRONT', 'EC']},
{'trajector': 'a man',               'landmark': 'a ladder',                 'spatial_indicator': 'with',      'rel_type': ['EQ', 'NTPP']},
{'trajector': 'a car',               'landmark': 'it',                       'spatial_indicator': 'in front of', 'rel_type': 'FRONT'},
{'trajector': 'pedestrians',         'landmark': 'it',                       'spatial_indicator': 'in front of', 'rel_type': 'FRONT'},
{'trajector': 'a street',            'landmark': 'it',                       'spatial_indicator': 'in front of', 'rel_type': 'FRONT'}

```

Figure 7.14 An example of relation extraction using GPT-3.5.

mSpRL annotations. First, GPT-3.5 predicts more correct spatial relations or relation types than those annotated in mSpRL. Second, the model correctly identifies the first two relations in mSpRL. However, due to errors in entity extraction, where "lamps and windows" are not separated into two distinct entities, this prediction is labeled as incorrect. Thus, evaluating GPT-3.5 relation extraction performance using mSpRL annotations may not be a fair comparison. We compare the performance of these models based on their performance within the pipeline framework.

As depicted in Table 7.3, LLM_PiSTAQ surpasses PiSTAQ, which employs BERT-based extraction modules. This outcome highlights the high capability of LLMs in information extraction when compared to BERT-based modules. However, LLM_PiSTAQ still underperforms any end-to-end language models. This performance gap is primarily attributed to 1) the inherent complexities involved in the extraction task within real-world environments, 2) errors in output engineering to structure the string outputs generated by GPT-3.5, 3) the limitations of the designed symbolic reasoner to cover real-world commonsense rules. It is noteworthy to mention that, as discussed in Chapter 6, out of 25 sampled questions from RESQ, only seven can be solved without relying on spatial commonsense information. Our analysis shows that LLM_PiSTAQ can solve these seven questions.

In further exploration, we aim to integrate LLMs as neural spatial reasoners within a pipeline structure of extraction and reasoning. The process, illustrated in Figure 7.15, involves augmenting

<p>Story: a man in white shirt , black jacket , grey pants and black shoes is sitting on a wooden chair and talking on the phone . on the right a wooden bed with white bedcovers . on the left (before the man) a wooden desk and a vase with flowers . there is a black brief-case in front of the chair , and there is also a picture hanging on the wall above the bed .</p> <p>Relation_in_story= [("a man", "in", "white shirt"), ("a man", "in", "black jacket"), ("a man", "in", "grey pants"), ("a man", "in", "black shoes"), ("A man", "sitting on", "a wooden chair"), ("a man", "talking on", "the phone"), ("a wooden bed", "on", "the right"), ("a wooden bed", "with", "white bedcovers"), ("a wooden desk", "on", "the left"), ("a wooden desk", "before", "the man"), ("a vase", "on", "the left"), ("a vase", "before", "the man"), ("a vase", "with", "flowers"), ("a black brief-case", "in front of", "the chair"), ("a picture", "hanging on", "the wall"), ("a picture", "above", "the bed")]</p>
<p>Question: Is the man touching the chair? (the man, touching, the chair),</p> <p>Chain of reasoning: ("A man", "sitting on", "a wooden chair") which is externally connected and above. The answer is Yes</p>
<p>Question: Is the bedcover below the picture?(the bedcover, below, the picture),</p> <p>Chain of reasoning: ("a wooden bed", "with", "white bedcovers") which means bedcover has the same position as bedcover, ("a picture", "above", "the bed"), so (the picture, above, the bedcover), so (the bedcover, below, the picture). The answer is Yes.</p>

Figure 7.15 We employ LLMs in both extraction and reasoning tasks, but in a disentangled manner. Initially, we extract information using LLMs, and subsequently incorporate this extracted information into the prompt alongside written CoTs based on the extracted data.

the prompt of a GPT-3.5 with both the extracted information from LLM and the composed Chain of Thoughts (CoTs) based on this information. Contrary to expectations, the obtained result is even lower (62.62%) when compared to GPT-3.5-CoT with the main text (67.05%) during evaluation on the RESQ dataset. This outcome emphasizes the capability of LLMs to better comprehend information from natural language in contrast to structured data without further fine-tuning.

7.4.2 LLM with Probabilistic logical Reasoning

As illustrated in Figures 7.1 and 7.8, considering the top-5 relations’ token from LLM prediction can yield more insights and provide additional information for each pair. In this experiment, we integrate the probability of extracted information from GPT-3.5 with the probabilistic symbolic reasoner in the PISTAQ pipeline and call the model LLM_PISTAQ^{prob}. We select the top-5 relation predictions made by the GPT-3.5 along with their corresponding probability for each pair of entities with direct relations in the text. Subsequently, the extracted relations are fed into the probabilistic

#	Models	Accuracy
1	Majority baseline	50.21
2	BERT	57.37
3	BERT* <i>Zero_shot</i>	49.18
4	BERT*	63.60
5	GPT-3.5 ^{<i>Zero_shot</i>}	60.32
6	GPT-3.5 ^{<i>Few_shot</i>}	65.90
7	GPT-3.5 ^{<i>Few_shot</i>} +CoT	67.05
8	SREQA	53.15
9	SREQA* <i>Zero_shot</i>	53.32
10	SREQA*	69.50
11	PiSTAQ(BERT)	47.21
12	LLM_PiSTAQ	50.01
13	LLM_PiSTAQ + Commonsense	51.48
14	LLM_PiSTAQ ^{<i>prob</i>}	52.12
15	LLM_PiSTAQ ^{<i>prob</i>} (Only Commonsense)	49.50
16	LLM_PiSTAQ ^{<i>prob</i>} + Commonsense	54.76
17	Human	90.38

Table 7.3 Result on RESQ. *Further training on SPARTUN. The Zero_shot refers to evaluation without further training on RESQ or mSPRL training data.

logical reasoner. The results of this evaluation are presented in Table 7.3.

<p>Story: a king size bed with a night table on each side. phone on the right table.</p> <p>Question: Is the table under the phone? Answer: Yes</p> <p>-----</p> <p>Deterministic Reasoning</p> <p>Facts:1.0::right_('phone', 'the right table')</p> <p>Query:below(the table, the phone)? Answer: No (prob: 0.0)</p> <p>-----</p> <p>Probabilistic Reasoning</p> <p>Facts:0.293::right_('phone', 'the right table'). 0.208::above_('phone', 'the right table'). 0.195::front_('phone', 'the right table'). 0.015::below_('phone', 'the right table').</p> <p>Query:below(the table, the phone)? Answer: Yes (prob: 0.208)</p>

Figure 7.16 Comparing probabilistic and deterministic reasoning over an example of RESQ. The threshold to generate the Yes/No answers is 0.1, obtained by try and trials.

Comparing lines 12 and 14 in Table 7.3 shows that when we consider the uncertainty of the extracted relations (facts) in the probabilistic reasoning of LLM_PiSTAQ^{*prob*}, the outcomes are superior compared to scenarios where certain facts are used in deterministic reasoning of

LLM_PISTAQ. Figure 7.16 illustrates an example of this comparison. The ‘above’ and ‘below’ relations are missed in the deterministic facts extraction, while in the probabilistic one, the top second and fourth relations are ‘above and below.’ As a result, LLM_PISTAQ^{prob} correctly predicts the ‘Yes’ answer. These results highlight the critical insight that the prediction with the highest probability does not necessarily equate to the most accurate result. Moreover, considering the top-5 predictions enables us to consider multiple potential relations between entities, providing a more comprehensive and subtle understanding of the relationships within the data.

Despite the demonstrated improvement in LLM_PISTAQ^{prob} compared to LLM_PISTAQ when employing probabilistic reasoning, it is noteworthy that the performance of this model still falls short of that achieved by the other language models and SREQA models. While the transparent structure of the PISTAQ framework facilitates understanding the prediction process, interpreting the superior performance of language models on this task remains challenging due to their black-box nature. This higher performance may be attributed to better spatial and commonsense reasoning capability of LLMs or simply result from random predictions for Yes/No questions.

7.4.3 LLM as Spatial Commonsense Knowledge Extractor

One advantage of using LLMs as an end-to-end model, compared to LLM_PISTAQ, is their notable commonsense reasoning capabilities in generating responses for RESQ examples. This capability poses a considerable challenge for the PISTAQ or LLM_PISTAQ^{prob} model, which relies on explicit information in the text and covers limited spatial rules. We propose a novel methodology to leverage the commonsense knowledge embedded in LLMs within our model. In this approach, we provide tuples of all entities present in the text (extracted using LLMs) without any context as an input to the LLMs and ask for spatial commonsense relations. By excluding the main story text, the LLM returns the most common relation between pairs of entities based on its generalized knowledge of space. These relations are then incorporated with the direct facts from the text into the probabilistic spatial reasoner.

Comparing the outcomes presented in lines 14 and 16 of Table 7.3, we observe the enhancement in the performance of LLM_PISTAQ^{prob} when commonsense knowledge is utilized. Figure 7.17

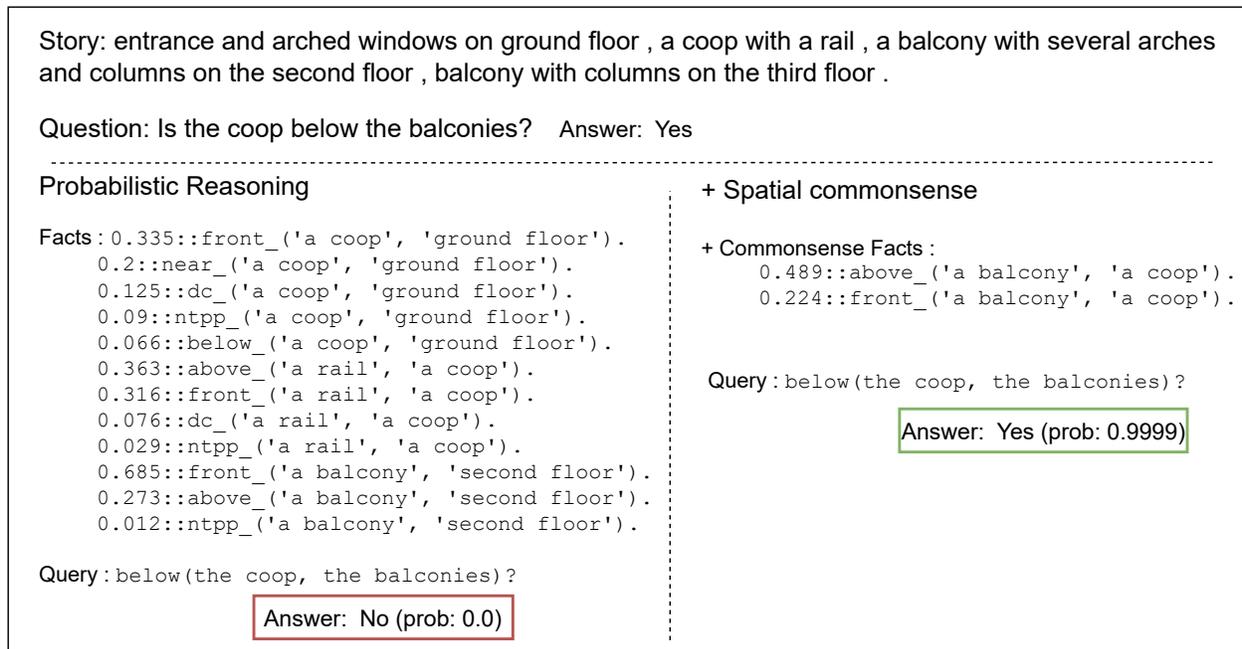


Figure 7.17 An illustration of the advantages of employing commonsense knowledge in probabilistic reasoning. Without the extra commonsense knowledge, the question cannot be answered.

depicts an example highlighting the benefits of using commonsense knowledge in probabilistic reasoning. The relation between ‘the balcony’ and ‘the coop’ is not stated in any sentences from the story. However, based on a common understanding that balconies are typically on upper floors while coops are on the ground, it can be inferred that a balcony should be above a coop. This commonsense knowledge is reflected in the LLM output with a probability of 0.48. This information is then utilized in the probabilistic reasoning, resulting in a "Yes" answer.

Furthermore, an additional experiment assesses the impact of incorporating commonsense knowledge into deterministic reasoning. As illustrated in lines 12 and 13 of Table 7.3, integrating commonsense knowledge deterministically also improves results. However, employing commonsense knowledge in probabilistic reasoning surpasses deterministic reasoning by a 3.3% gap, highlighting the added value of probabilistic reasoning in this context. Similar to previous experiments, incorporating such knowledge from GPT-3.5 into the pipeline model cannot outperform the GPT-3.5 or GPT-3.5+CoT models. While this method increases the extracted information compared to the primary PISTAQ model, the remaining errors from the extraction part prevent it from surpassing the end-to-end LLMs.

7.5 Summary

In conclusion, our research has elucidated several key findings regarding the role of LLMs in spatial reasoning tasks. We have shown that even though utilizing Chain-of-Thought prompting can enhance the performance of LLMs on SQA task, they still have shortcomings and are far behind the human performance. Our case studies show that, despite the inherent limitations of LLMs in spatial reasoning, capitalizing on their potential for information extraction within a disentangled structure of extraction and reasoning can yield some benefits. However, our experimental results in a real-world environment indicate that this high performance is more robust in controlled environments with a fixed pattern of spatial descriptions and is challenging in real-world scenarios. GPT-3.5 exhibits limitations in extracting complex spatial information from natural language that directly influences the symbolic reasoner, resulting in performance lower than end-to-end LLM models.

Besides, we propose approaches to enhance the performance of the pipeline model with LLM as an extraction module. Our findings demonstrate that leveraging probabilistic outputs from LLMs in probabilistic logical reasoning produces superior results compared to deterministic reasoning. This emphasizes the advantages of considering uncertainties of language models' outputs for reasoning tasks. Furthermore, incorporating LLMs' commonsense knowledge significantly improves the pipeline model's performance by encapsulating some commonsense rules and relations that the symbolic reasoning module may not cover. However, in the end, due to errors from the extraction modules, the pipeline model still underperforms the end-to-end language models in realistic domains.

CHAPTER 8

CONCLUSION AND FUTURE DIRECTIONS

This chapter summarizes the research presented in this thesis and emphasizes the outcomes of each contribution. Furthermore, we outline some interesting future directions based on the foundation of this work.

8.1 Summary of Contributions

In the first contribution, this thesis addresses the lack of evaluation benchmarks for multi-hop spatial reasoning by proposing three novel textual corpora for spatial question answering (SQA). These new benchmarks represent a significant contribution to the field, enabling more robust evaluation and comparison of different models in spatial reasoning tasks. The first benchmark, `SPARTQA-HUMAN`, offers a human-generated dataset of spatial questions and answers based on controlled visual scenes. This contribution allows researchers to evaluate language models using data that reflects human perception and interpretation of spatial relationships. The second benchmark, `SPARTQA-AUTO`, is an automatically generated corpus using hand-crafted grammar and spatial reasoning rules. This contribution provides a large dataset for training and evaluating models, which can be especially useful for early-stage experiments and simulations. The third benchmark, `RESQ`, is a realistic human-curated SQA dataset derived from spatial role labeling (SpRL). This contribution extends existing benchmarks with spatial semantic annotations, offering a unique resource for evaluating language models in real-world contexts. Through extensive experiments, we demonstrated that state-of-the-art pre-trained language models struggle with spatial reasoning tasks on human-generated text, emphasizing the need for improved models. Our findings also highlighted that perfect results on auto-generated data might indicate overfitting due to limited vocabulary and grammatical structures, underscoring the importance of diverse and realistic benchmarks.

The second main contribution is enhancing the spatial reasoning capability of neural models through transfer learning. By creating a new synthetic dataset, `SPARTUN`, which covers a broad range of spatial relation types, combinations, and spatial language expressions, we provide a comprehensive source of supervision for pretraining and transfer learning. This dataset allows for

extensive pretraining of language models (LMs), leading to significantly improved performance in spatial question answering and spatial role labeling tasks. Moreover, our experiments show that language models pretrained with SPARTUN exhibit superior results on several external datasets, indicating the robustness of our approach. The enhanced spatial reasoning in these models suggests that SPARTUN offers a richer source of supervision for pretraining due to its expanded coverage of spatial relation types and combinations compared to other synthetic data sources. This broader scope contributes to the field by providing a more effective foundation for developing spatially-aware neural models.

As our main third contribution, we investigate the benefits of disentangling the processes of extracting spatial information and reasoning over them. To this end, we devised a series of experiments utilizing PLMs for spatial information extraction coupled with a symbolic reasoner for inferring indirect relations. The outcomes of our experiments provide noteworthy insights: (1) Our observations in controlled experimental conditions demonstrate that disentangling extraction and symbolic reasoning, compared to PLMs, enhances the models' reasoning capabilities, even with comparable or reduced supervision. (2) Despite the acknowledged fragility of symbolic reasoning in real-world domains, our experiments highlight that employing explicit extraction layers and utilizing the same symbolic reasoner in data preprocessing enhances models' reasoning capabilities. These outcomes highlight the importance of distinguishing the processes of extraction and reasoning, offering a pathway for building more robust models in spatial language understanding. Our results suggest that this approach can lead to better-performing models in various applications on complex spatial reasoning tasks.

As our fourth and last contribution, our investigation has uncovered insights into the function and efficacy of Large Language Models (LLMs) in spatial reasoning tasks. We demonstrate that LLMs, particularly those employing Few_shot or Chain-of-Thought prompting techniques, still face challenges solving spatial reasoning tasks. Our case studies and experiments reveal that while LLMs face challenges in spatial reasoning, harnessing their strengths in information extraction within a pipeline of Extraction and Reasoning can lead to improvements, especially in

controlled environments. However, in a real-world setting, this approach falls short of outperforming end-to-end language models due to the limitations of GPT-3.5 in extracting complicated spatial information from natural language. Moreover, our experimental outcomes of using LLM in this pipeline model indicate that employing the probabilistic outputs from LLMs in probabilistic logical reasoning consistently outperforms deterministic reasoning. This result highlights the benefits of incorporating uncertainties in reasoning tasks. Finally, using commonsense knowledge from LLMs that comprehend commonsense rules and relations enhances the performance of the pipeline model. Even though these methods enhance knowledge acquisition from the context during the extraction phase, none can mitigate the extraction errors from the LLMs. Hence, the pipeline model still underperforms the end-to-end language models.

8.2 Future Directions

In this section, we outline several promising directions for future research that build upon the findings and methodologies established in this thesis. While our work has made significant contributions to the field of spatial reasoning, there is ample scope for extending and refining our approach to explore new areas and address potential limitations. Below we detail potential avenues for future work.

- **Methods For Prompt Engineering:** In the pipeline of extraction and reasoning, one of the shortcomings of employing LLMs lies in their subpar performance in extracting spatial relations from text. A hypothesis to solve this issue is refining prompt engineering, potentially improving the in-context learning of LLMs for spatial information tasks. Investigating automated methods with LLM in the loop to discover optimal prompts for this specific task could be an intriguing avenue for future research. For instance, we can maintain a collection of prompts and select the optimal prompts based on the test example to enhance performance.
- **Exploring Other Reasoning Types:** In this thesis, we focus on spatial reasoning, which contains both language understanding and multi-step reasoning. An extension would be to evaluate our approaches to other types of reasoning that require logical rules, such as temporal reasoning and arithmetic reasoning, to provide valuable insights into the versatility

and effectiveness of these methods. For instance, the rules over RCC-8 relations, which are designed in our rule-based spatial reasoner, can also be practical in temporal reasoning. Consequently, incorporating an off-the-shelf model for temporal information extraction within the context of a disentangled extraction and reasoning framework could be beneficial.

- **Expanding to Other Languages:** The current work primarily targets English language, but the methodology can be easily adapted to other languages that share similar grammatical structures. An intriguing avenue for future work is applying our approach to different languages and assessing the benefits of disentangling extraction and reasoning. This would require developing customized extraction modules while utilizing the same spatial reasoner in our designed pipeline model.
- **Optimizing LLMs for Spatial Reasoning:** Our research demonstrates that LLMs may not excel at spatial reasoning tasks when acting alone but show improvement when used with symbolic reasoning within a disentangled model. Additionally, we observe that the performance of LLMs diminishes when extracted information is presented in the prompt of in-context learning as structured data. A promising direction for future research would be to investigate strategies that allow LLMs to utilize such data without need for further fine-tuning.

PUBLISHED PAPERS

1. Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjamshidi. 2021. *SPARTQA: A Textual Question Answering Benchmark for Spatial Reasoning*. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4582–4598, Online. Association for Computational Linguistics.
2. Roshanak Mirzaee and Parisa Kordjamshidi. 2022. *Transfer Learning with Synthetic Corpora for Spatial Role Labeling and Reasoning*. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 6148–6165, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
3. Roshanak Mirzaee and Parisa Kordjamshidi. 2023. *Disentangling Extraction and Reasoning in Multi-hop Spatial Reasoning*. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore. Association for Computational Linguistics.
4. Roshanak Mirzaee and Parisa Kordjamshidi. 2023. *Dual-Phase Models for Extracting Information and Symbolic Reasoning: A Case-Study in Spatial Reasoning*. In Proceedings STRL-23 Second International Workshop on Spatio-Temporal Reasoning and Learning, Macao, S.A.R. CEUR Workshop Proceedings.

BIBLIOGRAPHY

- [1] Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics.
- [2] Reem Alrashdi and Simon O’Keefe. *Automatic Labeling of Tweets for Crisis Response Using Distant Supervision*, page 418–425. Association for Computing Machinery, New York, NY, USA, 2020.
- [3] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683, 2018.
- [4] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.
- [5] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity, 2023.
- [6] John A Bateman, Joana Hois, Robert Ross, and Thora Tenbrink. A linguistic ontology of space for natural language processing. *Artificial Intelligence*, 174(14):1027–1071, 2010.
- [7] Robert Bossy, Wiktor Golic, Zorana Ratkovic, Philippe Bessieres, and Claire Nédellec. Bionlp shared task 2013—an overview of the bacteria biotope task. In *Proceedings of the BioNLP shared task 2013 workshop*, pages 161–169, 2013.
- [8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [9] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. TOUCHDOWN: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12538–12547, 2019.
- [10] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

- [11] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, 2019.
- [12] Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as soft reasoners over language. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3882–3890. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.
- [13] Soham Dan, Parisa Kordjamshidi, Julia Bonn, Archana Bhatia, Zheng Cai, Martha Palmer, and Dan Roth. From spatial relations to spatial configurations. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5855–5864, Marseille, France, May 2020. European Language Resources Association.
- [14] Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. Quoref: A reading comprehension dataset with questions requiring coreferential 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 5925–5932, 2019.
- [15] Surabhi Datta and Kirk Roberts. A hybrid deep learning approach for spatial trigger extraction from radiology reports. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2020, page 50. NIH Public Access, 2020.
- [16] Surabhi Datta, Yuqi Si, Laritza Rodriguez, Sonya E Shooshan, Dina Demner-Fushman, and Kirk Roberts. Understanding spatial language in radiology: Representation framework, annotation, and spatial relation extraction from chest x-ray reports using deep learning. *Journal of biomedical informatics*, 108:103473, 2020.
- [17] Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. Problog: A probabilistic prolog and its application in link discovery. In *IJCAI*, volume 7, pages 2462–2467. Hyderabad, 2007.
- [18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.
- [19] Kaustubh Dhole and Christopher D. Manning. Syn-QG: Syntactic and shallow semantic rules for question generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 752–765, 2020.
- [20] Emmanuelle-Anna Dietz, Steffen Hölldobler, and Raphael Höps. A computational logic approach to human spatial reasoning. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 1627–1634. IEEE, 2015.

- [21] Xinya Du and Claire Cardie. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.
- [22] Xinya Du, Junru Shao, and Claire Cardie. Learning to Ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, 2017.
- [23] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378, 2019.
- [24] Saman Enayati, Ziyu Yang, Benjamin Lu, and Slobodan Vucetic. A visualization approach for rapid labeling of clinical notes for smoking status extraction. In *Proceedings of the Second Workshop on Data Science with Human in the Loop: Language Advances*, pages 24–30, Online, June 2021. Association for Computational Linguistics.
- [25] Hossein Rajaby Faghihi, Parisa Kordjamshidi, Choh Man Teng, and James Allen. The role of semantic parsing in understanding procedural text. *arXiv preprint arXiv:2302.06829*, 2023.
- [26] Yixing Fan, Xiaohui Xie, Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang, Jiafeng Guo, et al. Pre-training methods in information retrieval. *Foundations and Trends® in Information Retrieval*, 16(3):178–317, 2022.
- [27] John Freeman. The modelling of spatial relations. *Computer graphics and image processing*, 4(2):156–171, 1975.
- [28] Dayne Freitag, John Cadigan, Robert Sasseen, and Paul Kalmar. Valet: rule-based information extraction for rapid deployment. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 524–533, 2022.
- [29] Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models’ local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.
- [30] Matt Gardner, Jonathan Berant, Hannaneh Hajishirzi, Alon Talmor, and Sewon Min. Question Answering is a Format; when is it useful? *ArXiv*, abs/1909.11291, 2019.
- [31] Jeremy Getman, Joe Ellis, Zhiyi Song, Jennifer Tracey, and Stephanie M Strassel. Overview of linguistic resources for the tac kbp 2017 evaluations: Methodologies and results. In *TAC*, 2017.

- [32] Mehdi Ghanimifard and Simon Dobnik. Learning to compose spatial relations with grounded neural language models. In *IWCS 2017-12th International Conference on Computational Semantics-Long papers*, 2017.
- [33] M Grubinger and C Leung. A benchmark for performance calibration in visual information search. In *Proceedings of The 2003 International Conference on Visual Information Systems (VIS 2003)*, pages 414–419, 2003.
- [34] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don’t Stop Pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, 2020.
- [35] Hangfeng He, Qiang Ning, and Dan Roth. QuASE: Question-answer driven sentence encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8743–8758, Online, July 2020. Association for Computational Linguistics.
- [36] Hangfeng He, Mingyuan Zhang, Qiang Ning, and Dan Roth. Foreshadowing the benefits of incidental supervision. *arXiv preprint arXiv:2006.05500*, 2020.
- [37] Luheng He, Mike Lewis, and Luke Zettlemoyer. Question-Answer Driven Semantic Role Labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, 2015.
- [38] Michael Heilman and Noah A Smith. Question generation via overgenerating transformations and ranking. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA LANGUAGE TECHNOLOGIES INST, 2009.
- [39] A. Herskovits. *Language and Spatial Cognition: An Interdisciplinary Study of the Prepositions in English*. Cambridge University Press, 1986.
- [40] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [41] Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*, 2022.
- [42] Zixian Huang, Yulin Shen, Xiao Li, Yu’ang Wei, Gong Cheng, Lin Zhou, Xinyu Dai, and Yuzhong Qu. GeoSQA: A benchmark for scenario-based question answering in the geography domain at high school level. 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 5866–5871, 2019.
- [43] Drew A Hudson and Christopher D Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6700–6709, 2019.
- [44] Gilles Jacobs and Véronique Hoste. Sentivent: enabling supervised information extraction of company-specific events in economic and financial news. *Language Resources and Evaluation*, 56(1):225–257, 2022.

- [45] Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. TempQuestions: A benchmark for temporal question answering. In *Companion Proceedings of the The Web Conference 2018*, pages 1057–1062, 2018.
- [46] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017.
- [47] Maged N Kamel Boulos, Guochao Peng, and Trang VoPham. An overview of geoi applications in health and healthcare. *International journal of health geographics*, 18(1):1–9, 2019.
- [48] Bastian Karstaedt and Birgit Wendholt. Towards semantic interpretations of spatial information in adaptive smart homes. In *Forum Bauinformatik*, page 8. Citeseer, 2011.
- [49] William G Kennedy, Magdalena D Bugajska, Matthew Marge, William Adams, Benjamin R Fransen, Dennis Perzanowski, Alan C Schultz, and J Gregory Trafton. Spatial representation and reasoning for human-robot collaboration. In *AAAI*, volume 7, pages 1554–1559, 2007.
- [50] Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking Beyond the Surface:a challenge set for reading comprehension over multiple sentences. In *Proceedings of North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.
- [51] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. UnifiedQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, 2020.
- [52] Hyounghun Kim, Abhaysinh Zala, Graham Burri, Hao Tan, and Mohit Bansal. ArraMon: A joint navigation-assembly instruction interpretation task in dynamic environments. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3910–3927, Online, November 2020. Association for Computational Linguistics.
- [53] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.
- [54] Parisa Kordjamshidi, Marie-Francine Moens, and Martijn van Otterlo. Spatial Role Labeling: Task definition and annotation scheme. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 413–420. European Language Resources Association (ELRA), 2010.
- [55] Parisa Kordjamshidi, Taher Rahgooy, Marie-Francine Moens, James Pustejovsky, Umar Manzoor, and Kirk Roberts. Clef 2017: Multimodal spatial role labeling (msprl) task overview. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 367–376. Springer, 2017.

- [56] Parisa Kordjamshidi, Martijn Van Otterlo, and Marie-Francine Moens. Spatial role labeling: Towards extraction of spatial relations from natural language. *ACM Transactions on Speech and Language Processing (TSLP)*, 8(3):1–36, 2011.
- [57] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural Questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.
- [58] Igor Labutov, Sumit Basu, and Lucy Vanderwende. Deep questions without deep understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 889–898, 2015.
- [59] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.
- [60] Christian Landsiedel, Verena Rieser, Matthew Walter, and Dirk Wollherr. A review of spatial reasoning and interaction for real-world robotics. *Advanced Robotics*, 31(5):222–242, 2017.
- [61] Hung Le, Truyen Tran, and Svetha Venkatesh. Self-attentive associative memory. In *International Conference on Machine Learning*, pages 5682–5691. PMLR, 2020.
- [62] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.
- [63] Jumphon Lertlakkhanakul, Jin Won Choi, and Mi Yun Kim. Building data model and simulation platform for spatial interaction management in smart home. *Automation in Construction*, 17(8):948–957, 2008.
- [64] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-Shot relation extraction via reading comprehension. In *CONLL*, pages 333–342, 2017.
- [65] Dongdong Li, Zhaochun Ren, Pengjie Ren, Zhumin Chen, Miao Fan, Jun Ma, and Maarten de Rijke. Semi-supervised variational reasoning for medical dialogue generation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 544–554, 2021.
- [66] Jianfu Li, Qiang Wei, Omid Ghiasvand, Miao Chen, Victor Lobanov, Chunhua Weng, and Hua Xu. A comparative study of pre-trained language models for named entity recognition in clinical trial eligibility criteria from multiple corpora. *BMC Medical Informatics and Decision Making*, 22(3):1–10, 2022.
- [67] Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. A logic-driven framework for consistency of neural models. *arXiv preprint arXiv:1909.00126*, 2019.

- [68] Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. A unified mrc framework for named entity recognition. *arXiv preprint arXiv:1910.11476*, 2019.
- [69] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
- [70] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [71] Kate Lockwood, Ken Forbus, D Halstead, and Jeffrey Usher. Automatic categorization of spatial prepositions. In *Proceedings of the 28th annual conference of the cognitive science society*, pages 1705–1710, 2006.
- [72] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [73] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. *arXiv preprint arXiv:1808.09602*, 2018.
- [74] Oswaldo Ludwig, Xiao Liu, Parisa Kordjamshidi, and Marie-Francine Moens. Deep embedding for spatial role labeling. *arXiv preprint arXiv:1603.08474*, 2016.
- [75] Kai Ma, Yongjian Tan, Miao Tian, Xuejing Xie, Qinjun Qiu, Sanfeng Li, and Xin Wang. Extraction of temporal information from social media messages using the bert model. *Earth Science Informatics*, 15(1):573–584, 2022.
- [76] Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering. In *35th AAAI Conference on Artificial Intelligence*, 2021.
- [77] Arjun Magge, Davy Weissenbacher, Abeed Sarker, Matthew Scotch, and Graciela Gonzalez-Hernandez. Deep neural networks and distant supervision for geographic location mention extraction. *Bioinformatics*, 34(13):i565–i573, 06 2018.
- [78] Umar Manzoor and Parisa Kordjamshidi. Anaphora resolution for improving spatial relation extraction from text. In *Proceedings of the First International Workshop on Spatial Language Understanding*, pages 53–62, 2018.
- [79] Rui Mao, Qian Liu, Kai He, Wei Li, and Erik Cambria. The biases of pre-trained language models: An empirical study on prompt-based sentiment analysis and emotion detection. *IEEE Transactions on Affective Computing*, 2022.
- [80] David M Mark et al. *Languages of spatial relations: Researchable questions & NCGIA research agenda*. National Center for Geographic Information and Analysis Santa Barbara . . . , 1989.

- [81] Wouter Massa, Parisa Kordjamshidi, Thomas Provoost, and Marie-Francine Moens. Machine reading of biological texts: bacteria-biotope extraction. In *Proceedings of the 6th international conference on bioinformatics models, methods and algorithms*, pages 55–64. SCITEPRESS, 2015.
- [82] Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. A survey on multi-hop question answering and generation. *arXiv preprint arXiv:2204.09140*, 2022.
- [83] Alexey Mazalov, Bruno Martins, and David Matos. Spatial role labeling with convolutional neural networks. In *Proceedings of the 9th Workshop on Geographic Information Retrieval*, pages 1–7, 2015.
- [84] Ana Cristina Mendes, Luísa Coheur, and Paula Vaz Lobo. Named entity recognition in questions: Towards a golden collection. In *LREC*, 2010.
- [85] Julian Michael, Gabriel Stanovsky, Luheng He, Ido Dagan, and Luke Zettlemoyer. Crowdsourcing question-answer meaning representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 560–568, 2018.
- [86] Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Online, July 2020. Association for Computational Linguistics.
- [87] Pasquale Minervini, Sebastian Riedel, Pontus Stenetorp, Edward Grefenstette, and Tim Rocktäschel. Learning reasoning strategies in end-to-end differentiable proving. In *International Conference on Machine Learning*, pages 6938–6949. PMLR, 2020.
- [88] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, 2009.
- [89] Roshanak Mirzaee and Parisa Kordjamshidi. Transfer learning with synthetic corpora for spatial role labeling and reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6148–6165, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.
- [90] Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjamshidi. SPARTQA: A textual question answering benchmark for spatial reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4582–4598, Online, June 2021. Association for Computational Linguistics.
- [91] Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjamshidi. SPARTQA: A textual question answering benchmark for spatial reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*:

- Human Language Technologies*, pages 4582–4598, Online, June 2021. Association for Computational Linguistics.
- [92] Swaroop Mishra, Matthew Finlayson, Pan Lu, Leonard Tang, Sean Welleck, Chitta Baral, Tanmay Rajpurohit, Oyvind Tafjord, Ashish Sabharwal, Peter Clark, and Ashwin Kalyan. Lila: A unified benchmark for mathematical reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.
- [93] Diego Mollá, Menno Van Zaanen, Daniel Smith, et al. Named entity recognition for question answering. 2006.
- [94] Alaeddine Moussa, Sébastien Fournier, Khaoula Mahmoudi, Bernard Espinasse, and Sami Faiz. Spatial role labeling based on improved pre-trained word embeddings and transfer learning. *Procedia Computer Science*, 192:1218–1226, 2021.
- [95] Alaeddine Moussa, Sebastien Fournier, Khaoula Mahmoudi, Bernard Espinasse, and Sami Faiz. Mixing static word embeddings and roberta for spatial role labeling. *Procedia Computer Science*, 207:2950–2957, 2022. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 26th International Conference KES2022.
- [96] Huansheng Ning, Feifei Shi, Tao Zhu, Qingjuan Li, and Liming Chen. A novel ontology consistent with acknowledged standards in smart homes. *Computer Networks*, 148:101–107, 2019.
- [97] Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. TORQUE: A reading comprehension dataset of temporal ordering questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1158–1172, 2020.
- [98] Maxwell Nye, Michael Tessler, Josh Tenenbaum, and Brenden M Lake. Improving coherence and consistency in neural sequence models with dual-system, neuro-symbolic reasoning. *Advances in Neural Information Processing Systems*, 34:25192–25204, 2021.
- [99] Batu Ozturkler, Nikolay Malkin, Zhen Wang, and Nebojsa Jojic. Thinksum: Probabilistic reasoning over sets using large language models. *arXiv preprint arXiv:2210.01293*, 2022.
- [100] Rasmus Berg Palm, Ulrich Paquet, and Ole Winther. Recurrent relational networks. *arXiv preprint arXiv:1711.08028*, 2017.
- [101] Jason Phang, Thibault Févry, and Samuel R Bowman. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*, 2018.
- [102] James Pustejovsky, Parisa Kordjamshidi, Marie-Francine Moens, Aaron Levine, Seth Dworkman, and Zachary Yocum. Semeval-2015 task 8: Spaceval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (semeval 2015)*, pages 884–894. ACL, 2015.

- [103] Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. In *International Conference on Learning Representations*, 2021.
- [104] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.
- [105] Taher Rahgooy, Umar Manzoor, and Parisa Kordjamshidi. Visually guided spatial relation extraction from text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 788–794, 2018.
- [106] Hossein Rajaby Faghihi and Parisa Kordjamshidi. Time-stamped language model: Teaching language models to understand the flow of events. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4560–4570, Online, June 2021. Association for Computational Linguistics.
- [107] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 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, 2016.
- [108] David A Randell, Zhan Cui, and Anthony G Cohn. A spatial logic based on regions and connection. *KR*, 92:165–176, 1992.
- [109] Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*, 2022.
- [110] Jochen Renz and Bernhard Nebel. Qualitative spatial reasoning using constraint calculi. In *Handbook of spatial logics*, pages 161–215. Springer, 2007.
- [111] Homero Roman Roman, Yonatan Bisk, Jesse Thomason, Asli Celikyilmaz, and Jianfeng Gao. RMM: A recursive mental model for dialogue navigation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1732–1745, Online, November 2020. Association for Computational Linguistics.
- [112] Parisa Kordjamshidi Roshanak Mirzaee. Disentangling extraction and reasoning in multi-hop spatial reasoning. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 9.
- [113] Mohammed Saeed, Naser Ahmadi, Preslav Nakov, and Paolo Papotti. RuleBERT: Teaching soft rules to pre-trained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1460–1476, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.
- [114] Sunita Sarawagi et al. Information extraction. *Foundations and Trends® in Databases*, 1(3):261–377, 2008.

- [115] Priyanka Sen and Amir Saffari. What do models learn from question answering datasets? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2429–2438, 2020.
- [116] Dan Shen and Mirella Lapata. Using semantic roles to improve question answering. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 12–21, Prague, Czech Republic, June 2007. Association for Computational Linguistics.
- [117] Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Tianyi Zhou, and Daxin Jiang. Large language models are strong zero-shot retriever. *arXiv preprint arXiv:2304.14233*, 2023.
- [118] Zhengxiang Shi, Qiang Zhang, and Aldo Lipani. Stepgame: A new benchmark for robust multi-hop spatial reasoning in texts. In *Proceedings of the Association for the Advancement of Artificial Intelligence, AAAI ’22*, 2022.
- [119] Hyeong Jin Shin, Jeong Yeon Park, Dae Bum Yuk, and Jae Sung Lee. Bert-based spatial information extraction. In *Proceedings of the Third International Workshop on Spatial Language Understanding*, pages 10–17, 2020.
- [120] Rahil Soroushmojdehi, Sina Javadzadeh, Alessandra Pedrocchi, Marta Gandolla, et al. Transfer learning in hand movement intention detection based on surface electromyography signals. *Frontiers in Neuroscience*, 16:1–18, 2022.
- [121] Keith Stenning and Michiel Van Lambalgen. *Human reasoning and cognitive science*. MIT Press, 2012.
- [122] Kristin Stock, Christopher B Jones, Shaun Russell, Mansi Radke, Prarthana Das, and Niloofar Aflaki. Detecting geospatial location descriptions in natural language text. *International Journal of Geographical Information Science*, 36(3):547–584, 2022.
- [123] Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. A corpus of natural language for visual reasoning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 217–223, Vancouver, Canada, July 2017. Association for Computational Linguistics.
- [124] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, 2019.
- [125] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. *arXiv preprint arXiv:1503.08895*, 2015.
- [126] Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634, Online, August 2021. Association for Computational Linguistics.

- [127] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, 2019.
- [128] Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*, 2019.
- [129] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [130] Takuma Udagawa, Takato Yamazaki, and Akiko Aizawa. A linguistic analysis of visually grounded dialogues based on spatial expressions. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 750–765, Online, November 2020. Association for Computational Linguistics.
- [131] Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Large language models still can’t plan (a benchmark for llms on planning and reasoning about change). *arXiv preprint arXiv:2206.10498*, 2022.
- [132] Emile Van Krieken, Erman Acar, and Frank Van Harmelen. Semi-supervised learning using differentiable reasoning. *arXiv preprint arXiv:1908.04700*, 2019.
- [133] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [134] Sagar Gubbi Venkatesh, Anirban Biswas, Raviteja Upadrashta, Vikram Srinivasan, Partha Talukdar, and Bharadwaj Amrutur. Spatial reasoning from natural language instructions for robot manipulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11196–11202. IEEE, 2021.
- [135] Somin Wadhwa, Silvio Amir, and Byron C Wallace. Revisiting relation extraction in the era of large language models. *arXiv preprint arXiv:2305.05003*, 2023.
- [136] Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57:45, 2006.
- [137] Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. Gpt-re: In-context learning for relation extraction using large language models. *arXiv preprint arXiv:2305.02105*, 2023.
- [138] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, 2018.

- [139] Feng Wang, Peifeng Li, and Qiaoming Zhu. A hybrid model of classification and generation for spatial relation extraction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1915–1924, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.
- [140] Mengru Wang, Jianming Zheng, Fei Cai, Taihua Shao, and Honghui Chen. Drk: Discriminative rule-based knowledge for relieving prediction confusions in few-shot relation extraction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2129–2140, 2022.
- [141] Zhile Wang, Jianhua Yang, and Yu Guo. Unknown fault feature extraction of rolling bearings under variable speed conditions based on statistical complexity measures. *Mechanical Systems and Signal Processing*, 172:108964, 2022.
- [142] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.
- [143] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.
- [144] Diedrich Wolter. Sparq-a spatial reasoning toolbox. In *AAAI Spring Symposium: Benchmarking of Qualitative Spatial and Temporal Reasoning Systems*, page 53, 2009.
- [145] Lang-Tao Wu, Jia-Rui Lin, Shuo Leng, Jiu-Lin Li, and Zhen-Zhong Hu. Rule-based information extraction for mechanical-electrical-plumbing-specific semantic web. *Automation in Construction*, 135:104108, 2022.
- [146] Xin Xu, Yuqi Zhu, Xiaohan Wang, and Ningyu Zhang. How to unleash the power of large language models for few-shot relation extraction? *arXiv preprint arXiv:2305.01555*, 2023.
- [147] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764, 2019.
- [148] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. pages 2369–2380, October–November 2018.
- [149] Zhun Yang, Adam Ishay, and Joohyung Lee. Coupling large language models with logic programming for robust and general reasoning from text. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5186–5219, Toronto, Canada, July 2023. Association for Computational Linguistics.
- [150] Liang Yao, Chengsheng Mao, and Yuan Luo. KG-BERT: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*, 2019.

- [151] Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. Docred: A large-scale document-level relation extraction dataset. *arXiv preprint arXiv:1906.06127*, 2019.
- [152] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1753–1762, 2015.
- [153] Chong Zhang, Jiagao Lyu, and Ke Xu. A storytree-based model for inter-document causal relation extraction from news articles. *Knowledge and Information Systems*, 65(2):827–853, 2023.
- [154] Ningyu Zhang, Xiang Chen, Xin Xie, Shumin Deng, Chuanqi Tan, Mosha Chen, Fei Huang, Luo Si, and Huajun Chen. Document-level relation extraction as semantic segmentation. *arXiv preprint arXiv:2106.03618*, 2021.
- [155] Yue Zhang, Quan Guo, and Parisa Kordjamshidi. Towards navigation by reasoning over spatial configurations. In *Proceedings of Second International Combined Workshop on Spatial Language Understanding and Grounded Communication for Robotics*, pages 42–52, Online, August 2021. Association for Computational Linguistics.
- [156] Yue Zhang and Parisa Kordjamshidi. Explicit object relation alignment for vision and language navigation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 322–331, Dublin, Ireland, May 2022. Association for Computational Linguistics.
- [157] Chen Zheng and Parisa Kordjamshidi. Relational gating for " what if " reasoning. *arXiv preprint arXiv:2105.13449*, 2021.
- [158] Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. Temporal common sense acquisition with minimal supervision. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7579–7589, 2020.
- [159] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.
- [160] Rui Zhu, Krzysztof Janowicz, Ling Cai, and Gengchen Mai. Reasoning over higher-order qualitative spatial relations via spatially explicit neural networks. *International Journal of Geographical Information Science*, pages 1–32, 2022.
- [161] Xiangxin Zhu, Carl Vondrick, Charless C Fowlkes, and Deva Ramanan. Do we need more training data? *International Journal of Computer Vision*, 119(1):76–92, 2016.
- [162] Jordan Zlatev. Holistic spatial semantics of thai. In *Cognitive linguistics and non-Indo-European languages*, pages 305–336. De Gruyter Mouton, 2008.