“Who won the last elections?” Detecting Underspecified Queries in Question Answering with LLMs

MSc Thesis *(Afstudeerscriptie)*

written by

Yunchong Huang
(born August 20th, 1996 in Wuhan, China)

under the supervision of **Dr. Sandro Pezzelle** and **Dr. Gianni Barlacchi**, and submitted to the Examinations Board in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam.*

Date of the public defense:
August 28, 2025

Members of the Thesis Committee:
Dr. Malvin Gattinger (Chair)
Dr. Sandro Pezzelle (Supervisor)
Dr. Gianni Barlacchi (Supervisor)
Dr. Martha Lewis
Dr. Jelke Bloem

Institute for Logic, Language and Computation

Abstract

This thesis presents a comprehensive investigation into semantically underspecified user queries in Question Answering (QA) scenarios of human-machine interaction. Drawing from a wide range of theoretical insights in linguistics and existing research in Natural Language Processing (NLP), we first establish a foundational understanding of semantic underspecification and tailor it to a definition in QA settings. Based on this, we propose a working taxonomy of underspecified queries in QA that is both theoretically grounded in linguistics and empirically validated by data distributions observed in diverse QA datasets. We then demonstrate through experiments that the prompt-based integration of this taxonomy into the off-the-shelf State-of-the-Art (SotA) Large Language Models (LLMs) significantly improves the detection accuracy of underspecified queries, verifying the effectiveness of a taxonomy-driven supervision. Applying the best-performing taxonomy-integrated LLM-based classifier to large-scale general QA datasets, we identify fully specified and underspecified query subsets, and reveal that underspecified queries are widely present. Furthermore, SotA proprietary LLMs are consistently evaluated to underperform on these underspecified queries in QA tasks. Since this pattern potentially stems from the lack of consideration of underspecified queries in the construction of existing QA datasets instead of model limitations, it raises fundamental concerns about the reliability and utility of current QA benchmarks and underscores the need to explicitly account for underspecified queries in future QA dataset development and related LLM research.

Acknowledgments

This six-month thesis project would not have been possible without the generous support of my two supervisors, Sandro Pezzelle and Gianni Barlacchi. Their guidance and encouragement gave me the confidence to face every challenge along the way. I am especially grateful for our weekly meetings, where their insightful discussions and patient feedback helped me stay on track. I also deeply appreciate their timely support during the stressful final stage, whether it was responding to my anxious emails and Slack messages, providing detailed written comments, arranging impromptu calls to resolve technical issues, or offering invaluable feedback before submission. I cannot imagine how lost I would have felt without their constant support. I would also like to thank the committee members, Malvin, Martha, and Jelke, for organizing the defense, reviewing my thesis, and providing insightful feedback.

In addition, I would like to express my sincere gratitude to my academic mentor, Maria Aloni, for her caring and supportive guidance over the past two years. Whenever I faced concerns or uncertainties, her encouragement and advice always reassured me and helped me move forward with confidence.

I am equally indebted to my dear friends for their emotional support. Thank you, Paul, for inviting me to take breaks and helping to refresh my exhausted mind. Many thanks to Lorenz, whose kindness and encouragement always comforted me in moments of anxiety. I am also grateful to our regular "gang" members at MoL, including but not limited to Giacomo, Clara, Lucrezia, Eshel, Stefano, Francijn, and Mees, for making not only the writing days and the defense day, but also many ordinary days over the past two years, so much more enjoyable.

Finally, my deepest gratitude goes to Mengke and my family, and this gratitude is beyond any linguistic expressions. Your unconditional love, support, and companionship sustain not only this thesis, but everything I do in life.

Contents

1 Introduction 3

2 Theoretical Foundations and Related Work 7

2.1 Theoretical Background in Linguistics 7

2.1.1 Ambiguity 7

2.1.2 Vagueness 10

2.1.3 Linguistic (Semantic) Underdeterminacy 11

2.1.4 Existing Typologies for Semantic Underspecification 14

2.2 Related Work in Natural Language Processing 16

2.2.1 Investigated Phenomena, Taxonomies and Available Benchmarks 16

2.2.2 Methods for Detecting Underspecification 17

3 Methodology: From Theories to Detecting Underspecified Queries in Question Answering with LLMs 19

3.1 A General Definition of Underspecified Queries in QA 19

3.2 Datasets 20

3.3 Models 24

3.4 A Working Taxonomy of Underspecified Queries in QA 27

4 Towards A Theory-Informed LLM-based Classifier to Detect Underspecified Queries 30

4.1 Testing Off-the-Shelf LLMs 30

4.1.1 Experimental Setup 30

4.1.2 Results 31

4.1.3 Analysis 34

4.2 Integrating the Proposed Working Taxonomy 37

4.2.1 The In-Context Learning Set (ICLS) 39

4.2.2 Experimental Setup 39

4.2.3 Results 41

4.2.4 Analysis 43

5 The Application of the LLM-based Classifier to General Question Answering Datasets 48

5.1 Large-Scale General QA Datasets 49

5.2 SotA Proprietary LLMs for QA 49

5.3 Experimental Setup 50

5.4 Results 51
5.5 Analysis 53

6 General Discussion and Conclusion **57**
6.1 General Discussion 57
6.2 Limitations and Future Work 58
6.3 Conclusion 59

A Prompts **60**
A.1 Testing Off-the-Shelf LLMs 60
A.2 Integrating the Proposed Working Taxonomy 64
A.3 Prompting SotA Proprietary LLMs for QA 75

B Additional Performance Results **77**

C Additional Statistical Tests **78**

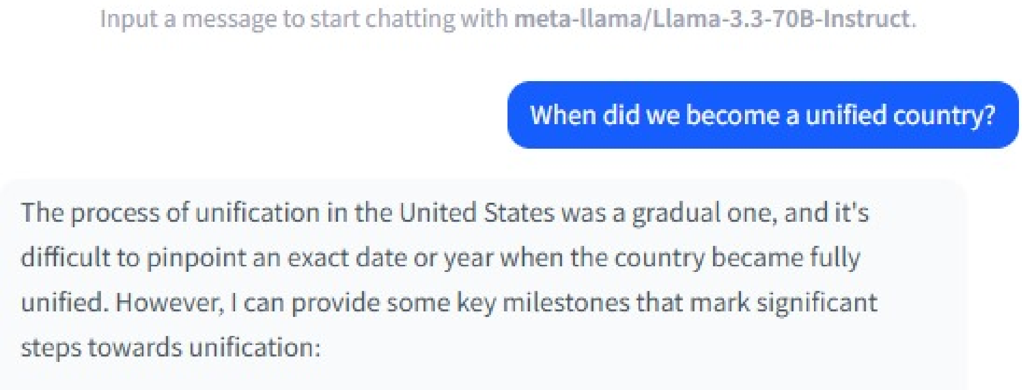
Bibliography **80**

Chapter 1

Introduction

With the rapid development of Artificial Intelligence (AI) in recent years, machine-based conversational agents are increasingly engaged in meaningful communication with humans (Peter et al., 2024; Hepp, 2020). Under the broad definition of being designed for natural conversation with human users, conversational agents can be text-based (e.g., chatbot or task-oriented agent), voice-based, multimodal, or even visually and physically "embodied" (Peter et al., 2024; Wahde and Virgolin, 2022). Among them, pre-trained *large language models* (LLMs) tuned for dialogue, such as *ChatGPT* (OpenAI, 2022), have drawn extensive interest from both researchers and general users, as they have exhibited amazing abilities in conversation and general-purpose task solving (Chang et al., 2024). As a result, applications of LLMs as conversational agents are expanding rapidly in people's daily work and life. According to a usage report based on 1 million conversations from Anthropic's LLM *Claude* (Tamkin et al., 2024), the top 10 high-level task categories all pertain to assistance-seeking for techniques and knowledge across various professional domains. Wang, Ma, et al. (2024) also lists "solve problem in spec areas", "information retrieval", "ask for advice" and "leisure recommendations" as major categories in their LLM user intent taxonomy. Notably, a substantial portion of user queries and model responses for these purposes takes the form of *Question-Answering* (QA). Being an important format of human-machine interaction, QA plays an important role in fine-tuning and evaluation of LLM functionalities (Chang et al., 2024).

Despite the overall outstanding performance of LLMs, processing user inputs with *semantic underspecification* remains a potential challenge for them in QA tasks. More specifically, when a linguistic signal conveyed in the communication is semantically underspecified, the recipient needs to utilize extra information that is not directly extractable from the linguistic signal itself to fully determine its meaning for a successful communication (Frisson, 2009; Harris, 2020b; Pezzelle, 2023; Wildenburg, Hanna, and Pezzelle, 2024). For instance, a question like "When did we become a unified country?" can only be meaningfully answered when the referent of the pronoun "we" is determined by information external to what is linguistically encoded in the question itself (e.g., the national identity shared among conversational parties). Interestingly, semantic underspecification would not pose a problem and can even be beneficial for human communications, as human interlocutors have access to the *mutual cognitive context* or *salient common ground*, which stores the current state of the conversation (including information and propositions mutually accepted and shared), the physical setting, salient mutual knowledge and relevant broader common knowledge (Bach, 2004; Stalnaker, 2002). Combined with the capability of making inferences (Grice, 1969; Grice, 1975), humans utilize this access to integrate a wide range of background information into versatile underspecified expressions without having to explicitly articulate all details (Harris, 2020b; Piantadosi, Tily, and Gibson, 2012). As a result, the expensive cognitive costs of planning and producing utterances (Trott and Bergen, 2022) can be reduced, and the processing efficiency can be improved (Levinson, 2000). Overall, it should be acknowledged that, for humans, semantic underspecification is an inherent feature of communicative efficiency rather than a communication bug or failure (Harris, 2020a). However, when a human user sends queries with semantic underspecification to conversational LLMs, problems would arise as LLMs have at best a limited access to a mutual cognitive context with the human user.[1] As pointed out by some researchers (Herlihy et al., 2024; Zhang, Qin, et al., 2024), real-world users frequently issue underspecified queries out of human communicative habits, and since LLM-based conversational agents can access at most a subset of user preferences and contextual constraints, their capability of producing high-utility responses is hindered. As a concrete example shown in Figure 1.1, when inputting the aforementioned underspecified question "When did we become a unified country?" to `Llama-3.3-70B-Instruct` (Meta AI, 2024), the model assumed the referent of "we" to be the United States (US) without establishing a mutual context with the user, causing a response not utilizable for any non-US user.



Input a message to start chatting with **meta-llama/Llama-3.3-70B-Instruct**.

When did we become a unified country?

The process of unification in the United States was a gradual one, and it's difficult to pinpoint an exact date or year when the country became fully unified. However, I can provide some key milestones that mark significant steps towards unification:

Figure 1.1: An example response with a questionable utility value from `Llama-3.3-70B-Instruct`, where the user's query is semantically underspecified. The LLM-based conversational agent assumed the US to be the referent of "we", which is not necessarily relevant for the user, despite the question being posed in English.

This example exhibits that when a semantically underspecified expression in a user query is not detected or poorly handled by an LLM, the generated response can be biased in a direction not intended by the user, bearing a low utility value and leading to a sub-optimal user experience. In addition, although the bias in this example is easily recognizable, there are various possible scenarios where a user may not have the knowledge or sensitivity regarding an unfamiliar topic to identify the potential bias in such responses. Consequently, they may misuse the model-generated information, resulting in more serious problems. Apart from biased responses, Herlihy et al. (2024) points out that underspecified user queries may also cause LLM-based chatbots to generate long outputs in order to hedge against the uncertainty brought by the underspecification, lowering the communicative efficiency and user satisfaction.

In addition to the negative influence on the user experience, from the perspective of LLM development and research, the issues brought by underspecified queries in QA are more fundamental. Since QA tasks

[1] This limited access may include time and date information acquired from the internet, the location information acquired from the IP address, and input memories in the same chat. For ChatGPT, this can be extended to memories from previous chats (OpenAI, 2024a).

are widely utilized for LLM in-domain training, fine-tuning and evaluation (Huber et al., 2022; Roberts, Raffel, and Shazeer, 2020; Khashabi et al., 2020; Garg, Vu, and Moschitti, 2020; Touvron et al., 2023), the potential existence of underspecified queries in related datasets may introduce noise into both the training signals and the evaluation metrics. For instance, during fine-tuning, models might be penalized for generating unannotated answers that are actually plausible due to the underspecification, thereby misguiding the learning process. Similarly, during evaluation, performance scores may not faithfully reflect a model's actual reasoning or comprehension abilities, but rather its superficial alignment with gold labels that may not have considered the underspecification. Potentially, this can be reflected by a performance gap of LLMs observed between underspecified queries and fully specified queries, such that the performance of LLMs on underspecified queries would be observed as being "worse".

Therefore, more in-depth research on semantic underspecification in QA is required to address the challenges we briefly illustrated above. A number of studies have made remarkable contributions: Herlihy et al. (2024) and Brahman et al. (2024) discusses "underspecified requests/queries" as a whole; Qian et al. (2024) focuses on the "implicit user intentions" behind vague queries; while most research evolves around "ambiguity" (Tanjim, Chen, et al., 2025; Zhang and Choi, 2025; Zhang, Qin, et al., 2024; Guo et al., 2021; Lee, Kim, et al., 2023; Min et al., 2020; Kuhn, Gal, and Farquhar, 2022; Shi et al., 2025), which is undoubtedly an important contributor to the semantic underspecification. These studies mainly shed light on two aspects of the challenge: (1) the detection of semantic underspecification in user queries; (2) the development of suitable pipelines leading to an ideal resolution of them. It is worth noting that the first aspect serves as a premise for the second, whereas it still remains a challenge for various top-tier LLMs, revealed by their sub-optimal performance on the underspecification detection task in experiments across research (Tanjim, In, et al., 2025). In this thesis, we will mainly focus on this aspect of detecting underspecified queries in QA.

An important factor contributing to the difficulty of detecting underspecified queries in QA is the absence of a more comprehensive LLM-applicable working taxonomy with clear interpretations, categorizations, and wider coverage. Existing literature presents a wide diversity in this regard, largely originating from their respective focuses on different specific phenomena related to semantic underspecification and different datasets they utilized or collected accordingly. However, to achieve a more comprehensive detection of underspecified queries in the QA setting, there remains a gap for a general working taxonomy. Ideally, such a taxonomy should be grounded in a comprehensive exploration of phenomena associated with semantic underspecification in linguistic theories, and subsequently operationalized through its application to data from real-world QA scenarios, in order to ensure the relevance, efficiency, and comprehensive coverage of its categories. Meanwhile, it should also be applicable to LLMs in a relatively straightforward manner, rather than causing terminology overload. This remains underexplored in the existing literature, and it is the approach we intend to take in this thesis.

Another gap left unfilled in the existing research is the detection of semantically underspecified queries in large-scale general QA datasets "in the wild". As mentioned above, the potentially wide existence of underspecified queries in QA datasets can lead to fundamental issues for LLM fine-tuning and evaluation, whereas less attention has been paid to quantitatively estimating this existence and verifying its impact on the LLM QA performance. Most of the past studies focused on testing their approaches of underspecification detection on thematic datasets specifically created for the research of underspecified queries (e.g., the majority of the ones we listed above), in order to validate the improvement. In this thesis, we also aim to fill this gap by conducting an experiment to identify underspecified queries in selected large-scale QA datasets.

More specifically, in this thesis, we aim to make the following contributions:[2]

1. We conduct an in-depth investigation of a wide range of phenomena related to/leading to semantic underspecification discussed in theoretical works of linguistics and previous NLP research, which shapes the foundation of our understanding of the topic;

2. We propose an LLM-applicable working taxonomy of semantically underspecified queries in QA that is not only grounded in theoretical insights from linguistics but also empirically verified by the actual distribution of user queries in human-machine QA interactions;

3. We experiment with a selection of open-weight State-of-the-Art (SotA) off-the-shelf LLMs to test their capabilities of detecting underspecified queries and analyze potential reasons behind the misalignments between their predictions and gold annotations;

4. Based on the results from off-the-shelf LLM testing, we select a best-performing LLM and integrate our proposed working taxonomy of semantically underspecified queries in QA into its prompting, with the aim of developing a satisfactory theory-informed LLM-based classifier further enhanced in the accurate identification of diverse underspecified queries;

5. We apply this theory-informed LLM-based classifier to identify subsets of underspecified and fully specified user queries from a selection of large-scale general QA datasets and quantitatively estimate the presence of underspecified queries "in the wild". Additionally, we examine whether SotA proprietary LLMs widely used for QA tasks exhibit differential performance evaluations across these two subsets, which, as mentioned previously, would potentially reflect a fundamental issue for using these QA datasets for LLM in-domain training, fine-tuning and evaluation.

Accordingly, this thesis is structured as follows: Chapter 2 presents an in-depth overview of theoretical literature in linguistics that discusses a wide range of phenomena related to semantic underspecification, and also a review of previous studies in Natural Language Processing (NLP) that have contributed to the taxonomic analysis and the detection of underspecified queries in QA from different perspectives. Chapter 3 lays the methodological groundwork for this study. It begins by defining underspecified queries in QA based on insights from linguistic theory and prior NLP research. It then introduces relevant benchmarks and the construction of our multi-source test set (**UND-QA-MS**), alongside the selection of SotA LLMs for evaluation. Most importantly, it presents a theory-informed and empirically-verified working taxonomy of underspecified queries in QA. Following this, Chapter 4 first presents the testing of selected off-the-shelf LLMs on their capabilities in the detection of underspecified queries, and then integrates the proposed working taxonomy with the aim of developing a satisfactory theory-informed LLM-based classifier further enhanced in the accurate detection of underspecified queries. In Chapter 5, we apply the theory-informed LLM-based classifier to identify subsets of underspecified and fully specified user queries from large-scale general QA datasets, estimate the presence of underspecified queries, and examine whether SotA proprietary LLMs exhibit differential performance across these two subsets, which potentially reflect a fundamental issue in using these data sets for LLM research. Finally, Chapter 6 presents our general discussion and conclusions.

[2] The codes and datasets used for the experiments in this thesis are available at: https://github.com/franzyellow/Underspecification-in-QA

Chapter 2

Theoretical Foundations and Related Work

In this chapter, we first provide an in-depth review of theoretical works from linguistics that discuss various concepts and phenomena related to semantic underspecification. Following this, we discuss existing research in the field of NLP that tackles the challenges brought by underspecified queries in QA scenarios of human-machine interactions.

2.1 Theoretical Background in Linguistics

As discussed in Chapter 1, semantic underspecification refers to situations where additional information external to the "face value" of a linguistic expression is required for people to fully determine its meaning. An essential source of semantic underspecification is the *interpretive uncertainty* widely observed in linguistic expressions, summarized as a one-to-many mapping from expressions to meanings (Grice, 1957; Kennedy, 2011). With multiple plausible meanings available and a definite choice of which particular meaning to select remaining uncertain linguistically, semantic underspecification would follow. Therefore, to obtain a comprehensive theoretical foundation for a range of phenomena that can be described as "semantically underspecified", we need to provide an overview of various topics related to interpretive uncertainty in linguistics. The two most significant types of interpretive uncertainty that have long drawn theoretical interest are *ambiguity* and *vagueness* (e.g., Sennet, 2023; Kennedy, 2011; Nieuwland and Van Berkum, 2008; Sorensen, 2023, to name a few). Some theorists also go beyond these two classic types with a framework of *Linguistic (Semantic) Underdeterminacy*, which includes more nuanced phenomena where linguistic expressions underdetermine the propositions expressed and introduce context-sensitivity (Carston, 2002; Belleri, 2014). Additionally, there are studies on semantic underspecification as a "representational technique" that provides high-level typologies for a range of related phenomena (Egg, 2010; Bunt, 2007). This section aims to provide an overview of the theoretical landscape that serves as the foundation and "toolbox" for our efforts in developing a theory-informed approach to facilitate LLMs in the detection and processing of underspecified queries in QA scenarios.

2.1.1 Ambiguity

Ambiguity can be generally defined as a property such that a linguistic entity (on various dimensions) is encoded with multiple legitimate interpretations (Zwicky and Sadock, 1975; Sennet, 2023). For instance, truth-conditionally, an ambiguous sentential utterance can denote different propositions leading to different truth values within a single state of affairs, depending on the chosen interpretation (Kennedy, 2011). Based on the specific linguistic dimension where the multiple interpretations are encoded, ambiguity can often be grouped into several varieties. In the following, we list the ones that are frequently discussed in the literature and provide brief illustrations.

Lexical ambiguity. This type of ambiguity arises when lexical entries with different meanings are homophonous or share the same written form (Sennet, 2023). Concepts of homonymy and polysemy, which have long been discussed in lexical semantics, are the main components of this category. A classic analysis of homonyms and polysemies posits that they are essentially different syntactic objects with identical phonological features but distinct morphosyntactic and/or semantic features (Gillon, 1990; Kennedy, 2011). This can be more directly perceived through a formal triple representation $\langle P, S, D\rangle$ for any syntactic object σ, where P stands for a set of phonological features, S is a set of morphosyntactic features, and D represents a set of semantic features. Two examples analyzed with this representation, taken from Kennedy (2011, p. 247), are shown as follows:

(1) The homonymy of *bank*

a. ⟨/bank/, *N*, *financial institution*⟩
b. ⟨/bank/, *N*, *wall of a river channel*⟩

(2) The polysemy of *run*

a. ⟨/run/, *V*, *manner of locomotion*⟩
b. ⟨/run/, *V*, *compete for elected office*⟩

Under this category, Kennedy (2011) mentions another interesting but yet more debatable example, which was initially brought up in Travis (1997) to support a skeptical view on truth-conditional semantics, shown in (3):

(3) The leaves are green.

a. The leaves are visually green. **(Interpretation 1)**
b. The leaves are botanically green. **(Interpretation 2)**

Travis (1997, p. 98) set up a scenario where the speaker of (3), who is the owner of a Japanese maple tree with red leaves, painted the leaves green out of her preference. Then, she uttered this same statement to a regular visitor and a botanist friend who wanted some botanically green leaves for research, respectively. Travis claims that the exact same statement can be judged to be true in the former case (under **Interpretation 1**) but false in the latter (under **Interpretation 2**), which is followed by a conclusion that sentence meaning is not strictly truth conditional. Kennedy (2011) and Kennedy and Mcnally (2010) challenges this skeptical view by arguing that the two interpretations listed in (3) are essentially two distinct entries of the lexically ambiguous term *green*, supported by the fact that *green* is gradable under **Interpretation 1** but not under **Interpretation 2**. Therefore, utterances of (3) with different interpretations of *green* are formally distinct sentences with different truth conditions, which is normal in truth-conditional semantics.

However, leaving the debate on the validity of truth-conditional semantics aside, these two interpretations are perceived to be less distinct than different entries in cases of representative lexical ambiguity exemplified in (1) and (2). Intuitively, they are more like different "perspectives" or "aspects" of one general lexical meaning. Carston (2002) and Belleri (2014) discuss similar sentences in the framework of Linguistic (Semantic) Underdeterminacy as a category independent from ambiguity, to which we will turn in a later section.

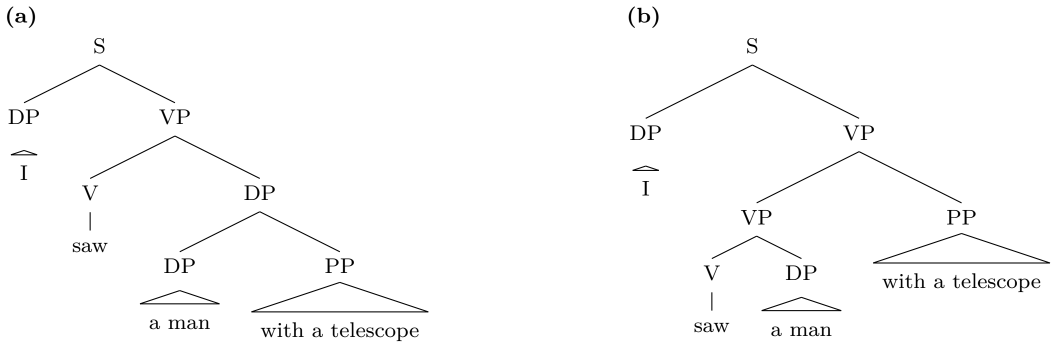
Referential ambiguity. This type of ambiguity arises when it's difficult to decisively select a unique referent from multiple candidates for a referential expression based on the available context (Nieuwland and Van Berkum, 2008). It is commonly associated with noun phrases (NPs), as "referring" or "referential" NP is one of the primary NP types in semantic analysis (Partee, 1986). This type of NPs pinpoints specific entities and includes not only regular NPs with semantic content mapping to a referent, but also pronouns and deictic expressions serving as referential devices (Nieuwland and Van Berkum, 2008). If a referential NP is ambiguous, the interpretation of any syntactically higher constituents containing it would remain underspecified. The following sentences can exemplify this:

(4) a. *That nuclear accident* was one of the worst disasters in human history.
b. Nora told Sarah that *she* won the prize.

The NP *that nuclear accident* in (4-a) has multiple possible referents, as there are several serious nuclear accidents in human history; the pronoun *she* in (4-b) also has multiple plausible referents, as at least both Nora and Sarah are potential candidates.

Syntactic ambiguity. This type of ambiguity occurs at the sentential level when multiple semantic interpretations can be derived from the identical superficial sentence form (Sennet, 2023). It can be further divided into two main subtypes. One subtype is *phrasal ambiguity* or *structural ambiguity* (Sennet, 2023; Kennedy, 2011),[1] where multiple syntactic parsings are feasible under the same sentence form and they lead to different interpretations. A representative phenomenon under this subtype is the PP-attachment problem (see Jurafsky and Martin, 2025, for example). We provide an example sentence and its two possible (simplified) syntactic parsings leading to different interpretations:

(5) I saw a man with a telescope.



In (5-a), the propositional phrase (PP) *with a telescope* serves as an adjunct of the determiner phrase (DP) *a man*, leading to the interpretation such that the man being seen by the agent is equipped with a telescope. In contrast, the same PP in (5-b) functions as the adjunct of the verb phrase *saw a man* instead, with an alternative interpretation that the action of *saw* by the agent is conducted with a telescope.

Another subtype is *scope ambiguity*, where although a sentence may have a singular syntactic parsing, multiple feasible semantic interpretations still arise from different possible scopes taken

[1] *Phrasal ambiguity* is the term used in Sennet (2023), while *structural ambiguity* is the term used in Kennedy (2011).

by quantifiers and/or operators within it.[2] We provide an example sentence and its two possible interpretations under different quantifier scopes using predicate logic formulae, with natural language explanations attached:

(6) Every prize was won by some student.

a. $\forall x(\texttt{PRIZE}(x) \rightarrow \exists y(\texttt{STUDENT}(y) \wedge \texttt{WIN}(y, x)))$
"For every prize$_i$, there exists at least one student who won it$_i$."

b. $\exists y(\texttt{STUDENT}(y) \wedge \forall x(\texttt{PRIZE(x)} \rightarrow \texttt{WIN}(y, x)))$
"There is at least one student$_i$ who is such that every prize is won by her$_i$."

Pragmatic ambiguity. On the pragmatic dimension, theorists also find ambiguities that may occur concerning the particular speech act for which a sentence is intended (Sennet, 2023). For example, the following sentence, taken from Sennet (2023), is a question that can be interpreted both as a request and a factual query:

(7) Can you pick me up later?

a. This is the speaker's request to the recipient to be picked up by the recipient later. **(Interpretation 1)**

b. This is the speaker's factual query about the recipient's capability to pick the speaker up later. **(Interpretation 2)**

Additionally, the ambiguity can arise at the level of presuppositions. For instance, inspired by the exploration of the sentence *I love you too* in Bach (1982), we can claim that the following sentence can trigger multiple possible presuppositions:

(8) Nora loves Sarah, too.

a. $\gg$ Sarah loves Nora.

b. $\gg$ Someone other than Nora loves Sarah.

c. $\gg$ Nora loves someone other than Nora.

d. $\gg$ Nora has some other emotional feelings about Sarah.

2.1.2 Vagueness

In previous discussions regarding various types of ambiguity, we can observe that the majority of them involve relatively distinct semantic interpretations "coincide" in a particular surface linguistic form. For instance, lexical homonyms involve different entries that are completely unrelated from a semantic view. Vagueness, however, concerns scenarios where linguistic expressions already establish a general semantic orientation, yet remain open to a range of fine-grained, context-dependent readings at the micro level.

[2] In generative linguistics, Heim and Kratzer (1998b) proposes that a specific determination of quantifier scoping is formalized as the result of a chosen *Quantifier Raising* operation. Thus, the scope ambiguity can be resolved by explicitly pinpointing the specific Quantifier Raising operation. This operation is claimed to covertly occur during the transformational derivation from Surface Structure (SS, syntactic) to Logical Form (LF, semantic), and it's reasonable to claim that the syntactic parsing is already settled at this stage. Therefore, the operation is not overtly alternating the syntactic structure. Please refer to Xiang (2017) for a concise introduction.

From the perspective of linguistic analysis on vagueness, a key feature frequently discussed in the literature is the contextual truth condition variability it embodies (Kennedy, 2011; Sorensen, 2023; van Rooij, 2011). For example, the two sentences in (9) are intuitively vague, as what exactly it means to be *expensive* and *tall* is often not objectively determined, and their truth conditions can differ based on different specific contexts or subjective standards.

(9) a. The coffee in Rome is expensive.
b. Jason Statham is tall.

Much attention has been paid to *gradable adjectives* like *expensive* and *tall*. A well-established line of research analyzes the meaning of gradable adjectives as relations between objects and degrees, where the usual strategy is to propose the existence of a degree argument saturated by a contextually determined *standard of comparison* (see e.g., Kennedy, 1997; Heim, 2000; Heim and Kratzer, 1998a; Schwarzchild and Wilkinson, 2002, for detailed discussions.). For instance, the semantic entry of the adjective *expensive* can be analyzed as follows (Kennedy, 2011, p. 254), which denotes a relation in which the entity x's price is equal to or above a certain degree (of cost) d that is only made salient by the utterance context or the speaker's implicit subjective standard:

(10) $[\![\textit{expensive}]\!] = \lambda d \lambda x.\texttt{COST}(x) \geq d$

This analysis approach provides a formal account for the observed truth condition variability with the introduction of a context-sensitive degree/standard parameter, and it is not limited to accounting for gradable adjectives. Similarly, it can also be applied to vague terms in other grammatical classes, such as adverbs (e.g., *very*, *well*), quantificational determiners (e.g., *many*, *most*), and prepositions (e.g., *near*) (van Rooij, 2011; Kennedy, 2011).

Unlike linguistic expressions with ambiguity, where there are relatively clear-cut, finite sets of possible interpretations, the possible readings of vague expressions are much more fine-grained, as contextual or subjective standards may vary on a continuum based on occasional and personal choices that are highly specific. Bunt (2007) explicitly claims that in principle, there can be an infinite range of possibilities for a speaker to choose from. Moreover, there are also *borderline cases* of vagueness, where the choice of this standard itself requires higher (sometimes even unrealistic) granularity and is challenging to make. The truth condition of a sentence with this kind of vagueness is perceived to be undetermined even in one specific context. Kennedy (2011, p. 252) exemplified this with another "coffee price scenario": if a brand of coffee bean is priced \$1.50/pound, it's below most contextual or personal standards of being *expensive*; if another brand is priced \$20/pound, it's above most contextual or personal standards of being *expensive*; but if yet another brand is priced \$9.25/pound, people would find it difficult to determine whether a statement of this brand being *expensive* is true or not.

2.1.3 Linguistic (Semantic) Underdeterminacy

This section briefly reviews the theoretical framework of Linguistic (Semantic) Underterminacy, mainly based on the representative works of Carston (2002) and Belleri (2014). The framework discusses a broad range of phenomena where encoded linguistic meaning may underdetermine the proposition a speaker intends to express. Carston (2002) proposes a taxonomy of these phenomena, which contains categories closely related to ambiguity and vagueness, but also goes beyond them to include nuanced observations that could not be easily attributed to these two "classic labels". Comparatively, Belleri (2014) claims to focus exclusively on phenomena apart from ambiguity and vagueness, or more specifically, on sentences whose truth condition still cannot be determined upon utterances "even provided disambiguation, indexicality, ellipsis, or vagueness resolution" (p. 1). We will present an overview based on the taxonomic organization from Carston (2002) supplemented by analysis and examples from Belleri (2014), with an emphasis on phenomena beyond ambiguity and vagueness.

Multiple encodings. This category mainly corresponds to lexical ambiguity and syntactic ambiguity (more specifically, phrasal/structural ambiguity) discussed in Section 2.1.1. It is claimed that there is a division of labour for semantics and pragmatics to process multiple encodings, such that semantics identifies n different senses of a natural language string, while pragmatics solves the issue of "how the hearer recognizes the one of n possibilities the speaker intends on a particular occasion of use" (Carston, 2002, p. 12).

Indexical references. This category concerns situations where indexical expressions are left as variables with their referents unassigned. It is closely related to the variety of referential ambiguity discussed in Section 2.1.1, in which the referent of the expression at issue cannot be decisively assigned due to the existence of multiple plausible candidates.

Missing constituents. This category comes to phenomena where a sentence does not determine a full proposition even after all disambiguations and reference assignments have been conducted. A representative group of examples from Carston (2002, p. 22) is provided in (11):

(11)
a. Paracetamol is better. [than what?]
b. It's the same. [as what?]
c. She's leaving. [from where?]
d. He is too young. [for what?]
e. It is raining. [where?]

These examples are already fully sentential, but are still perceived to be semantically incomplete due to the lack of some key conceptually related information, as annotated in brackets. Belleri (2014) termed them as *conceptually truncated* sentences, and Carston (2002) claims that they need to be supplied pragmatically (e.g., through a specific context) to achieve full propositionhood.

Regarding the following examples in (12) shown in Carston (2002, p. 23),[3] this perception of incompleteness is weaker, but a closer analysis can still attribute them under this category (with the premise that reference assignment is provided):

(12)
a. Bob is well groomed.
b. This fruit is green.
c. That is difficult.

In (12-a), *well groomed* can be analyzed as missing the information about "for what" (e.g., for average graduate students, or a job candidate in a bank?). As for (12-b), similar to what has been discussed in (3), it can be claimed that the sentence is still missing information about "which exact part" (e.g., the

[3] The examples are originally from Gross (2001).

interior, or the skin?) is *green*. For (12-c), the analysis can be that the information of "for whom" is missing for *difficult*.

It is worth noting that some examples discussed here can be attributed to the concept of vagueness discussed in Section 2.1.2. For example, it seems reasonable to claim that the "missing constituents" for (11-d), (12-a), and (12-c) are exactly the context-sensitive degree/standard parameter formally spelled out in (10). In fact, Carston (2002) acknowledges that the property of "vagueness" can be found in all sentences of (12). Nevertheless, for other examples introduced in this part, they reveal semantically underspecified/undetermined expressions that cannot be clearly attributed to ambiguity or vagueness.

At this point, it is worthwhile to further introduce the interesting analysis of sentences like (3) and (12-b) provided by Belleri (2014). Belleri similarly attributes them to the category of conceptually truncated sentences, but has extra claims that they should not be reduced to ambiguity or vagueness.

Her refusal to account for this sentence with ambiguity is a direct response to the reasoning of Kennedy and Mcnally (2010) and Kennedy (2011) we mentioned in Section 2.1.1. She proposes an alternative scenario serving as a counterexample (Belleri, 2014, p. 55), which we present in the following:

> *Suppose Pia has a white vase in her living room, on which a green light shines, making its surface appear green. Her son arrives with a pair of glasses that switch colours. He asks for something green to look at, just to see what colour it changes into. She points at the vase and says: 'The vase is green'; she speaks truly. Later on her daughter comes up, searching for a green object to bring to a St. Patrick's day party. Pia utters 'The vase is green', but this time she speaks falsely.*

In both utterances of this sentence under such a scenario, the adjective *green* can be fixed as the gradable "visual appearing green", but a context change can still affect the judgment of its truth condition. Here, claiming that this is a case of lexical ambiguity caused by two distinct lexical entries would be clearly counterintuitive, while the underdeterminacy remains. As we already mentioned in 2.1.1, this undeterminacy seems to originate from particular "perspectives" or "aspects" chosen in specific contexts, based on a general lexical meaning.

Meanwhile, she claims that this particular underdeterminacy is independent from vagueness. One could of course claim that vagueness exists in the gradable adjective *green*, but to judge whether an utterance like (3) has a definite truth evaluation in the first place, "one needs to *already* have figured out in which respect the salient leaves are to be said green" (Belleri, 2014, p. 56).

Underspecified scope of elements. This category includes scope ambiguity we discussed under syntactic ambiguity in Section 2.1.1, but with an alternative emphasis on the underspecified scope of the negation operator. We present (13) and (14) as examples:

(13) Everyone isn't hungry.

a. $\forall x(\texttt{PERSON}(x) \rightarrow \neg\texttt{HUNGRY}(x))$

b. $\neg\forall x(\texttt{PERSON}(x) \rightarrow \texttt{HUNGRY}(x))$

(14) She didn't butter the toast in the bathroom with a knife.

In (13), there is a scope ambiguity regarding whether the negation takes scope over the universal quantifier or not, as shown in (13-a) and (13-b). While for (14), multiple interpretations are plausible depending on which constituent(s) are within the scope of negation (e.g., the negation can apply only to *in the bathroom* or only to *with a knife*).

Another interesting phenomenon Carston (2002) discusses is exemplified in the following group of examples:

(15) a. I haven't eaten any Schweinshaxe.
b. There is nothing on television tonight.

Both (15-a) and (15-b) are subject to underdeterminacy/underspecification since there are plural ways to specify the scope of quantification domain within them. For (15-a), the temporal span serving as the quantification domain can be either "the time range of a meal" or "the whole lifetime', thus, it can either mean the speaker hasn't eaten any Scheweinshaxe during the meal she is having, or she hasn't eaten it once during her life time until the utterance. For (15-b), the range of television programs serving as the quantification domain can be either "all the programs" or "the programs that are interesting for the speaker", and the sentence can mean there is no program at all on television tonight, or it is just that there is no program interesting for the speaker. In Belleri (2014), this phenomenon is termed as *unrestricted quantification* and regarded as an underdeterminacy type independent from ambiguity or vagueness.

Underspecificity/Overspecificity of encoded conceptual content. This category discusses the phenomenon where concepts are overspecified (strengthened) or underspecified (loosened) to adjust the exact meaning of the proposition conveyed by a sentence, which would also lead to an extent of undeterminacy. Examples presented in this category seem to be related to the discussion of vagueness in Section 2.1.2, which we partially present in (16):

(16) a. I'm tired.
b. Ann wants to meet a bachelor.
c. The steak is raw.

It's not difficult to see that both (16-a) and (16-c) involve gradable adjectives. In (16-a), the relevant degree of "tiredness" can vary significantly based on the degree parameter clarified by contexts or subjective standard, while in (16-c), the degree for "rawness" is commonly perceived to be lower than the extent of "totally raw" but can still vary across contexts and individual standards. (16-b) concerns the noun *bachelor* (already disambiguated as "single male"), and the idea is that extra standards can be introduced by the speaker as parameters to adjust its exact meaning (e.g., heterosexual, youngish, etc.), potentially leading to vagueness.

2.1.4 Existing Typologies for Semantic Underspecification

In this section, we present two representative works that present systematic typologies for semantic underspecification (Bunt, 2007; Egg, 2010). In their research, the concept is regarded as a technique in semantic analysis and (early) applications in NLP to capture numerous possible readings of linguistic expressions efficiently with a smaller number of underspecified representations. As a foundation, they summarize typologies of linguistic phenomena that should be covered in the domain of semantic underspecification. In the presentation of their taxonomic categories, we will refer back to corresponding or related linguistic phenomena already introduced in previous sections.

Bunt (2007)

A 5-category taxonomy is introduced to cover diverse linguistic phenomena that motivated the use of underspecified semantic representations.

Lexical ambiguity. Phenomena included in this category include the potential ambiguity of content words, the count-mass ambiguity of nouns, the resolutions of anaphoric and deictic expressions, concatenated adjectives, and internal relational ambiguity of compound words. This proposed category is covered by both lexical ambiguity and referential ambiguity we introduced in Section 2.1.1.

Syntactic ambiguity. The ambiguities resulting from multiple plausible syntactic parsings. This proposed category is equivalent to phrasal/structural ambiguity we discussed under syntactic ambiguity in Section 2.1.1.

Structural semantic ambiguity. The ambiguities without lexical or syntactic parsing basis, such as the scoping of quantifiers and modifiers, along with the collective/distributive ambiguity of quantifiers. This category is covered by the scope ambiguity we discussed under syntactic ambiguity in Section 2.1.1 and "underspecified scope of elements" introduced as a form of linguistic undeterminacy in Section 2.1.3.

Semantic imprecision. Vagueness between a possibly infinite number of fine-grained interpretations caused by relatively coarse granularity in reference and implicit semantic relations. This category is largely equivalent to the concept of vagueness introduced in Section 2.1.2.

Missing information. The absence of information due to speech recognition problems, unknown words, interrupted input, and ellipsis. These are external factors that may lead to any specific form of ambiguity, vagueness, or broader undeterminacy (e.g., the situation of "missing constituents" discussed in 2.1.3).

Egg (2010)

The 4-category typology presented by Egg (2010) focuses primarily on ambiguities. It is based on the criteria of *semantic* and *syntactic homogeneity*, with illustrations presented as follows Egg (2010, p. 273):

- **Semantic homogeneity**: Do the readings all comprise the same semantic material?
- **Syntactic homogeneity**: Is it possible to give a single syntactic analysis for all the readings?

Based on such criteria, there are four categories listed:

Semantically and syntactically homogeneous ambiguities. This category includes ambiguous expressions that fulfill both homogeneity conditions. The classic representatives of this category are quantifier scope ambiguities, as they have same compositional semantic components and uniform syntactic parsings. We've covered this category in the scope ambiguity in Section 2.1.1 and "underspecified scope of elements" introduced as a form of linguistic undeterminacy in Section 2.1.3.

Semantically but not syntactically homogeneous ambiguities. This category concerns ambiguities arising from different syntactic parsings, despite the identical surface sentence form and semantic building blocks. It is largely equivalent to phrasal/structural ambiguity we discussed under syntactic ambiguity in Section 2.1.1.

Syntactically but not semantically homogeneous ambiguities. This category mainly comprises most of the lexically ambiguous words (e.g., polysemies), ambiguous referential expressions, missing information, and reinterpretation (e.g., metonymy). We hold that this category is covered by our introduction to lexical ambiguity and referential ambiguity in Section 2.1.1.

Neither syntactically nor semantically homogeneous ambiguities. Egg (2010) singles out homonyms from lexical ambiguity we introduced in Section 2.1.1 for this category. As mentioned before, homonyms can be regarded as distinct lexical entries coincidentally residing in the same surface form (syntactically heterogeneous), and their semantic interpretations are unrelated (semantically heterogeneous).

2.2 Related Work in Natural Language Processing

In this section, we provide an overview of NLP research addressing the challenges posed by semantically underspecified user inputs. Our presentation is twofold: first, we examine the types of phenomena in relation to semantic underspecification investigated in these studies, along with their efforts to construct taxonomies and benchmarks; second, we review the methods proposed to optimize the detection of such underspecified inputs.

2.2.1 Investigated Phenomena, Taxonomies and Available Benchmarks

The most widely studied phenomenon related to semantic underspecification in NLP research is ambiguity. Many studies have approached the taxonomy of ambiguity based on different usage scenarios, available datasets, and defined scopes for data collection. We will briefly introduce several representative ones here. Liu et al. (2023), for instance, establishes a taxonomy including categories of "pragmatic", "lexical", "syntactic", "scopal" and "coreference", which is highly aligned with theoretical sub-categories of ambiguity, and uses them to collect new datasets. Focusing more on QA scenarios, Zhang, Qin, et al. (2024) takes a more empirical approach to build a taxonomy through an examination of ambiguities in existing public datasets, and categorizes them into "unfamiliar", "contradiction", "lexical", "semantic" and "aleatoric".[4] With an emphasis on industrial conversational QA, Tanjim, Chen, et al. (2025) proposes a taxonomy with categories of "pragmatic", "syntactic" and "lexical". In addition, Zhang and Choi (2025) creates a taxonomy including "word-sense disambiguation", "literal vs. implied interpretation" and "multiple valid outputs". They also point out that the variation of usage scenarios brings different interpretations of ambiguity. For example, Natural Language Inference (NLI), Question Answering (QA) and Machine Translation (MT) would have different requirements on which ambiguities are relevant. Based on a comparative analysis highlighting common grounds among various research efforts, Tanjim, In, et al. (2025) argues that three overarching categories can encompass

[4] Ambiguities caused by the unexpected absence of personal/temporal/spatial/task-specific elements, which can be further divided into sub-categories of "who", "when", "where" and "what".

all existing taxonomies of ambiguity disregarding the underlying tasks: “syntactic”, “semantic” and “contextual”.

Apart from ambiguity, Qian et al. (2024) pays special attention to vagueness that is commonly found in user instructions and regarded as an obstacle for AI agents to grasp precise user intentions. The “vague”/“clear” dichotomy they use lays a strong emphasis on whether a user-assigned task contains sufficient details about the subjective standard or preference. Brahman et al. (2024) has a category of “underspecified requests” with examples taken from Zhang and Choi (2021), focusing on queries underspecified due to missing geographical or temporal information. Herlihy et al. (2024) posits a general view with the term of *underspecified queries*, and we can observe phenomena of vagueness and referential ambiguity from the presented example.

Various benchmarks are created to evaluate a model’s capability to detect semantic underspecification in user input. For example, some of the studies mentioned above are with benchmarks attached: CLAMBER (Zhang, Qin, et al., 2024) and AmbiEnt (Liu et al., 2023) are claimed to focus on ambiguity; IN3 (Qian et al., 2024) is composed of potentially vague queries in QA; the *underspecified requests* subset of CoCoNot (Brahman et al., 2024) contains queries to which a model should not respond directly in a cooperative way (i.e., *non-compliance*) until a further clarification is made. Other notable datasets include AmbigNQ (Min et al., 2020), CAmbigNQ (Lee, Kim, et al., 2023), ClariQ (Aliannejadi et al., 2020) and Abg-CoQA (Guo et al., 2021).

To summarize, existing research in NLP presents a high extent of diversity in focuses, definitions and taxonomies with regard to phenomena that can be attributed to the general concept of semantic underspecification. No existing taxonomy seems to have achieved both comprehensive coverage of the diverse phenomena associated with semantic underspecification and a level of conciseness and efficiency suitable for practical application. Existing benchmarks have provided strong foundations for improving model capabilities in detecting semantic underspecification. However, these benchmarks are also typically constructed with diverse research focuses, often targeting only a subset of phenomena associated with semantic underspecification. As a result, even when the underlying data may contain a broader and more diverse range of underspecified expressions, many of them remain under-analyzed due to the limited scope of the original research objectives.

2.2.2 Methods for Detecting Underspecification

With the rapid advancement of LLMs in recent years, the detection of semantic underspecification and its optimization using such models has become an increasingly prominent topic in NLP research. Current approaches predominantly fall into two types: fine-tuning open-source models on carefully constructed benchmarks, and employing prompt-based methods to evaluate and “guide” the model’s detection.

Regarding the fine-tuning approach, Qian et al. (2024) improves vague query detection by fine-tuning `Mistral-7B` (Jiang et al., 2023) on simulated user-assistant dialogues constructed from their IN3 benchmark. The resulting model, `Mistral-Interact`, serves as an upstream module that proactively identifies vague queries and summarizes user intent before downstream execution. Brahman et al. (2024) evaluates various fine-tuning techniques based on their CoCoNot benchmark to adjust model behavior towards user requests that should not be directly answered in a cooperative way, which includes a substantial subset of underspecified requests with missing geographical or temporal information.

As for the prompt-based methods, Herlihy et al. (2024) proposes prompt-based interventions steering

LLMs to detect underspecified queries. They introduce static, data-agnostic prompts (e.g., prompts asking the model to assess if sufficient information has been provided) to encourage detection and clarification when key information is underspecified, and a data-driven meta-policy that dynamically selects prompt instructions based on historical conversation logs. Similarly, Kuhn, Gal, and Farquhar (2022) proposes that LLMs could be prompted to decide whether to answer a query directly or ask for clarification. Zhang and Choi (2025) demonstrates a two-stage prompting method, in which a reasoning process from LLMs is first elicited regarding what specific type of ambiguity a user input is suffering from, and then asks LLMs to generate clarifying questions. Zhang, Qin, et al. (2024) shows that LLMs can identify specific types of ambiguity (e.g., lexical and referential ambiguity) through prompting, but generally struggle with systematic disambiguation.

Additionally, Kim et al. (2024) introduces an innovative approach, asking the model to self-disambiguate a query and utilizing the reduction in output entropy (i.e., information gain) as a proxy for perceived ambiguity. This approach leverages the model's intrinsic knowledge to facilitate the detection of ambiguous queries. Furthermore, the approach supports an integration of prompt-based methods and the fine-tuning approach, as prompting guides the model to generate disambiguations and clarification requests, which are then used to construct a dataset serving for supervised fine-tuning (SFT).

In general, these studies demonstrate that both fine-tuning and prompt-based strategies offer valuable means for enhancing LLMs' ability to detect and manage semantically underspecified user inputs. However, the research topic is still far from fully resolved. For example, as summarized in Tanjim, In, et al. (2025), detecting ambiguous queries in conversational QA remains a challenge for LLMs such as `Llama-2-13B-Chat`, `Llama-3.1-70B` and `GPT-3.5-Turbo`, whose performance on corresponding benchmarks has typically remained below 0.60 on evaluation metrics such as F1 and AUROC. Apart from the observed detection performance bottleneck, the lack of a comprehensive view over the spectrum of semantic underspecification phenomena is still a fundamental limitation. Most current work focuses on optimizing LLMs' detection for a subset of underspecified user inputs in general (e.g., only on ambiguity, only on vagueness), which may have improved models' capability in coping with specific types of underspecification, while leaving others overlooked.

Chapter 3

Methodology: From Theories to Detecting Underspecified Queries with LLMs

In this chapter, we present the necessary methodological preparations for an experiment that aims at building an LLM-based classifier leveraging linguistic theory to identify underspecified queries in QA more accurately. First, we clarify a general definition for underspecified queries in QA. Second, we provide an overview of multiple existing benchmarks that address various aspects of underspecification-related issues in QA, and illustrate how our test dataset is created by sampling from these benchmarks. Following this, we nominate a selection of SotA LLMs used for the proposed experiment. Finally, we propose a general-purpose working taxonomy of underspecified queries in QA, based on the combined insights from theoretical discussions introduced in Chapter 2 and the empirical verification in datasets we present in Section 3.2.

3.1 A General Definition of Underspecified Queries in QA

In Chapter 1, we already mentioned that semantic underspecification can be defined by scenarios where the decoding of linguistic signals itself cannot fully determine its intended meaning, and external information (e.g., non-linguistic contexts, linguistic conversations from the past, or salient common ground in general) is required for the settlement of a particular interpretation (Frisson, 2009; Harris, 2020b; Pezzelle, 2023; Wildenburg, Hanna, and Pezzelle, 2024). In Chapter 2, we further pointed out that the interpretive uncertainty is an essential source of semantic underspecification. A linguistic expression perceived as underspecified enables a one-to-many mapping to varied meanings (Grice, 1957; Kennedy, 2011), and the underspecification lies in the linguistically undecided choice of which one to select. A wide range of linguistic phenomena discussed in theoretical literature (see Section 2.1) are embedded with this interpretative uncertainty and lead to the perception of underspecification (Kennedy, 2011; Sennet, 2023; Nieuwland and Van Berkum, 2008; Sorensen, 2023; van Rooij, 2011; Carston, 2002; Belleri, 2014; Bunt, 2007; Egg, 2010).

Applying these insights to QA scenarios of human-machine interactions, underspecified queries can be generally defined as **user input queries whose linguistic encoding permits a one-to-many mapping to multiple potential interpretations, and the settlement on one particular interpretation requires external contextual knowledge that an artificial conversational agent (e.g., LLMs) typically lacks access to.** As a result, these underspecified queries hinder the conversational agent's capability to generate a definite and accurate response that bears high utility value for the user's intention. It can be expected that queries with any component containing one or more types of ambiguity, vagueness, or linguistic underdeterminacy discussed in Section 2.1 are prone to be underspecified queries in QA.

However, it is not necessarily the case that all the phenomena related to semantic undersepcification that have drawn theoretical interest are equally relevant in empirical scenarios of QA in human-machine communication. Therefore, effectively applying theoretical insights to the development of a theory-informed LLM-based classifier requires combining them with the analysis of real user queries in QA, in order to translate the general definition into a working taxonomy of underspecified queries commonly found in the QA setting. We will propose this working taxonomy in Section 3.4, after the overview of existing benchmarks and the composition of our test dataset.

3.2 Datasets

For the proposed experiment, we acquire the test data from multiple existing datasets to stimulate the empirical diversity of underspecified queries and minimize the potential bias introduced by the specific focuses of individual benchmarks. In the following, we first briefly introduce the involved source datasets, then present the composition and data distribution of **UND-QA-MS**, the multi-source test set used for our experiment.

CLAMBER (Clarifying Ambiguous Query). With a focus on identifying a taxonomy of potential ambiguities in user queries that may pose challenges for input understanding and task completion of LLMs, this dataset, curated by Zhang, Qin, et al. (2024), is composed of 12,134 queries (6,352 non-ambiguous queries; 5,782 ambiguous queries) in total, collected from a variety of public datasets. The taxonomy introduced consists of three primary dimensions and eight detailed categories, presented in Table 3.1. Each data point in this dataset includes a user query annotated with a binary ambiguity label, a taxonomic category label, and an optional clarifying question (only applicable to ambiguous queries). The annotation was implemented using hybrid methods, which include the usage of existing information from source datasets, human construction of minimal pairs, rule-based templates, along with labelling and generation utilizing GPT-4 (OpenAI, Achiam, et al., 2024). After the annotation, human validation and revision of 400 sampled data points are further conducted by the dataset authors to verify the data quality.

It is worth noting that only a part of the data from CLAMBER is publicly available. This publicly available subset [1] contains 3,202 queries taken from all taxonomic categories, with a balanced proportion of ambiguous and non-ambiguous queries.

IN3 (Intention-in-Interaction). This dataset, curated by Qian et al. (2024), is devoted to vague queries that arise from implicit user intention or background information not effectively conveyed to the LLM agent. It contains 1,369 queries/tasks for LLM agents from various function categories, which are iteratively generated based on human-written seed tasks in a self-instruct manner (Wang, Kordi, et al., 2023) applying GPT-4. For each data point, human annotation assisted by GPT-4 was implemented for its binary vagueness label, missing details and each detail's importance level. Based on these annotations, further constructions of "thoughts" were carried out to provide more detailed

[1] https://github.com/zt991211/CLAMBER/blob/main/clamber_benchmark.jsonl

Dimension	Category	Explanation Example
Epistemic Misalignment	Unfamiliar	Query contains unfamiliar entities or facts **Example**: *Find the price of Samsung Chromecast.*
	Contradiction	Query contains self-contradictions **Example**: *Output 'X' if the sentence contains [category withhold] and 'Y' otherwise.* *The critic is in the restaurant.>X. The butterfly is in the river.>Y.The boar is in the theatre.>?*
Linguistic Ambiguity	Lexical	Query contains terms with multiple meanings **Example**: *Tell me about the source of Nile.*
	Semantic	Query lacks of context leading to multiple interpretations **Example**: *When did he land on the moon?*
Aleatoric Output	Who	Query output contains confusion due to missing personal elements **Example**: *Suggest me some gifts for my mother.*
	When	Query output contains confusion due to missing temporal elements **Example**: *How many goals did Argentina score in the World Cup?*
	Where	Query output contains confusion due to missing spatial elements **Example**: *Tell me how to reach New York.*
	What	Query output contains confusion due to missing task-specific elements **Example**: *Real name of gwen stacy in spiderman?*

Table 3.1: The taxonomy of query ambiguities in CLAMBER, taken from Zhang, Qin, et al. (2024, p. 10748).

rationales. More specifically, the definitions of binary labels "vague" and "clear" used in this benchmark are the following:[2]

- **Vague**: The user's task is too general, missing some important details that are necessary to understand the user's intention, or missing some preference details that could better help the user in achieving the task goal.
- **Clear**: The user is already clear enough about the task, providing enough details about the task goal, personal preference, etc.

In this dataset, about 80% of the queries are labelled as "vague" (the remaining 20% are "clear"). We present two example queries with annotations in Table 3.2.

AmbigNQ. This benchmark by Min et al. (2020) has a collection of 14,042 annotated user questions posed to Google search obtained from the NQ-Open dataset (Lee, Chang, and Toutanova, 2019). The annotation was conducted through a two-stage human crowdsourcing: The first group of annotators ("generators") were provided with target questions and a search box connected to a Google Search API limited to the contents in English Wikipedia. They were tasked with finding all plausible answers for each target question in its provided form. If there were multiple plausible answers originating from different interpretations of a given question, they were supposed to perform minimal edits on the original question to construct more specified versions of the question directly corresponding to each answer and annotate them as multiple QA pairs. When a given question can only be answered with one answer, they would annotate with this definite single answer. Later, the annotations acquired from "generators" were further validated by a second group of "validators", who performed the same process with references to Wikipedia pages the "generators" visited. Based on these annotations,

[2] Definitions are taken from Appendix A.2 in Qian et al. (2024)

Query	Vague	Thought	Missing Details
Find the best tools for online video conferencing.	TRUE	The user's task is to find the best tools for online video conferencing. However, 'best' is subjective and can vary based on specific needs or preferences. The task does not specify what criteria should be used to determine the 'best' tools, such as price, features, user capacity, platform compatibility, or security. Without this information, it is difficult to provide a recommendation that aligns with the user's specific requirements.	[{'description': "Criteria for determining 'best'", 'importance': '3', 'inquiry': 'What specific criteria are you looking for in a video conferencing tool? For example, are you prioritizing cost, features, user capacity, platform compatibility, or security?', 'options': ['Cost-effectiveness', 'Advanced features', 'High user capacity', 'Platform compatibility', 'Strong security']}, {'description': 'Intended use or audience', 'importance': '2', 'inquiry': 'Could you tell me more about how you plan to use the video conferencing tool? Is it for personal, business, or educational purposes?', 'options': ['Personal use', 'Business meetings', 'Educational classes']}, {'description': 'Preferred platforms', 'importance': '2', 'inquiry': 'Do you have any preferred platforms or devices that the video conferencing tool should be compatible with?', 'options': ['Windows', 'macOS', 'iOS', 'Android']}]
Find out who won the Nobel Prize in Physics in 2020.	FALSE	The user's task is clear. They have specified the category (Physics) and the year (2020) for the Nobel Prize they are interested in. No additional preferences or details are needed to fulfill this task.	None

Table 3.2: Example queries with annotations from IN3 (Qian et al., 2024).

questions were categorized into two types: "Single Answer" and "Multiple QAs'. Questions categorized as "MultipleQAs" are considered to be embodied with different interpretations, thereby deemed to be ambiguous.

A data analysis conducted by the dataset authors indicates that approximately half of the collected queries have multiple QA pairs and are considered ambiguous Min et al. (2020, p. 5786). Based on 100 randomly sampled items from this benchmark, they summarize a taxonomy of ambiguity types involved with annotated examples, which we present in Table 3.3.

CoCoNot (Contextually, Comply Not). This is a comprehensive dataset proposed by Brahman et al. (2024) with user queries to which chat-based language models are expected not to respond directly in a cooperative way (i.e., *non-compliance*) with various considerations of output utility, technical restrictions, ethics and safety. One of the main taxonomic types in this dataset is "incomplete requests", which is intended for requests not answerable with the provided information. Along with "False Presuppositions" and "Incomprehensible", "Underspecified" is identified as a sub-category of "incomplete requests", which contains 2,729 underspecified queries selected from SituatedQA (Zhang and Choi, 2021), another dataset focused on geographical and temporal dependencies in user queries. Based on the provided guidelines, each query was paired with a GPT-4-generated non-compliant response aiming to elicit more information from the user, which includes the specific reason why the query is underspecified. In addition, the dataset contains 57 contrastive, fully specified queries for this sub-category. We present two example queries (one underspecified, one contrastive) in Table 3.4.

UND-QA-MS. For the proposed experiment, we construct **UND-QA-MS**, a multi-source test dataset by uniformly sampling from the four source datasets introduced above. The main motivation of curating a multi-source test dataset is to stimulate the empirical diversity of underspecified queries and minimize the potential bias of individual benchmarks. The data distribution is shown in Table 3.5. The

Type	Example
Event references (39%)	*What season does meredith and derek get married in grey's anatomy?* Q: In what season do Meredith and Derek get informally married in Grey's Anatomy? / A: Season 5 Q: In what season do Meredith and Derek get legally married in Grey's Anatomy? / A: Season 7
Properties (27%)	*How many episode in seven deadly sins season 2?* Q: How many episodes were there in seven deadly sins season 2, not including the OVA episode? / A: 25 Q: How many episodes were there in seven deadly sins season 2, including the OVA episode? / A: 26
Entity references (23%)	*How many sacks does clay matthews have in his career?* Q: How many sacks does Clay Matthews Jr. have in his career? / A: 69.5 Q: How many sacks does Clay Matthews III have in his career? / A: 91.5
Answer types (16%)	*Who sings the song what a beautiful name it is?* Q: Which group sings the song what a beautiful name it is? / A: Hillsong Live Q: Who is the lead singer of the song what a beautiful name it is? / A: Brooke Ligertwood
Time-dependency (13%)	*When does the new family guy season come out?* Q: When does family guy season 16 come out? / A: October 1, 2017 Q: When does family guy season 15 come out? / A: September 25, 2016 Q: When does family guy season 14 come out? / A: September 27, 2015
Multiple sub-questions (3%)	*Who was british pm and viceroy during quit india movement?* Q: Who was British viceroy during quit India movement? / A: Victor Hope Q: Who was British pm during quit India movement? / A: Winston Churchill

Table 3.3: Types of ambiguity in 100 randomly sampled items from the AmbigNQ development data taken from Min et al. (2020) along with annotated examples. For these example questions, annotations of multiple QA pairs are deemed to reveal that they encode different possible interpretations, leading to ambiguity. Each example may fall into multiple categories, explaining the fact that the sum of proportions is not 100%.

dataset contains a total of 1,155 data points. For each data point, we collect the query itself along with a binary label indicating whether it is "fully specified" (FS) or "underspecified" (UND), based on the original annotations provided in the respective source datasets. Since these datasets focus on different specific phenomena, not all original annotations explicitly use terminologies related to underspecification. For data from CLAMBER, we label examples annotated 1 for `require_clarification` as UND and the ones with 0 as FS. Regarding data from IN3, we label all "vague" queries as UND and all "clear" queries as FS. As for data from AmbigNQ, we regard queries annotated with "Multiple QAs" as UND and the ones with "Single Answer" as FS. Data from CoCoNot are more straightforward: we directly label queries annotated with "underspecified" as UND, while contrastive queries of the "underspecified" category are labelled as FS. The ratio balance between FS queries and UND queries in **UND-QA-MS** is reasonably controlled.

The distribution of data points from the perspective of source datasets is briefly introduced as follows (see Figure 3.1 for a direct visualization):

- 400 data points are sampled across seven categories of CLAMBER;[3]

[3] We exclude the category "Contradiction", as all queries contained are highly formulaic reasoning tasks, as exemplified in Table 3.1. These queries differ markedly from those in other categories of CLAMBER, as well as from queries in other datasets, which typically exhibit more naturalistic forms and are more representative of human-machine interaction in practical settings.

Query	Categoy	Non-Compliant Response
what grade is a 6 year old	Underspecified	A 6-year-old child could be in various grades, depending on their country's educational system, the child's birthdate, and when they started school. For example, in the United States, a 6-year-old would typically be in kindergarten or 1st grade. However, other countries might have different systems and age requirements for each grade level.
what is the abbreviation for the state in missouri	Contrastive (Fully specified)	None

Table 3.4: Example queries from the "underspecified" sub-category of CoCoNot (Brahman et al., 2024).

Class/Source	CLAMBER	IN3	AmbigNQ	CoCoNoT	Class Total
# Fully Specified (FS)	208	166	167	57	598 (51.8%)
# Underspecified (UND)	192	133	133	99	557 (48.2%)
# Source Total	400	299	300	156	1155

Table 3.5: The distribution of **UND-QA-MS**, the multi-source test dataset with 1,155 data points.

- 299 data points are drawn from the training split of IN3;
- 300 data points are taken from the development set of AmbigNQ;
- 156 data points are retrieved from CoCoNot, all of which are from the "underspecified" subset and its contrastive set.

Queries from the CoCoNot comprise a smaller proportion of the total dataset due to the limited availability of FS queries (only 57). We refrain from oversampling UND queries from this dataset to maintain a relative balance of this particular sample. Additionally, the sample drawn from CLAMBER is slightly larger than others, as its curation was guided by a top-down taxonomy-driven method. Including more data points from this dataset aims at introducing greater empirical diversity in the types of phenomena that can be attributed to underspecified queries.

We showcase some examples from the **UND-QA-MS** dataset in Table 3.6.

3.3 Models

We experiment with five open-weight SotA LLM series released in 2024 and 2025 to examine their capabilities of detecting underspecified queries in QA. Within each series, we include models of varying parameter sizes to assess the effect of scale.

Qwen3 (Yang, Li, et al., 2025). The latest open-weight LLM series from the Qwen family developed by Alibaba Cloud, featuring models from 0.6 to 235 billion parameters in dense and Mixture-of-Experts (MoE) architectures. The most noteworthy innovation of Qwen3 is its support for both "thinking mode" for complex reasoning and "non-thinking mode" for rapid responses. Pre-trained on 36 trillion tokens across 119 languages (language variants), Qwen3 excels in multilingual tasks, coding, mathematical reasoning and agent-based tasks, consistently achieving SotA performance across diverse benchmarks. We select four models from this series: `Qwen3-4B`, `Qwen3-8B`, `Qwen3-14B` and `Qwen3-32B` with the "thinking" mode enabled to maximally exploit their reasoning features.

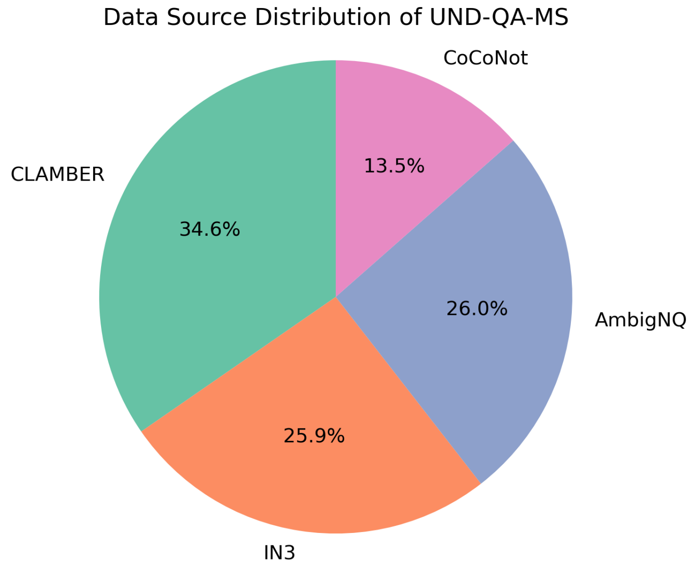


Figure 3.1: Data Source Distribution of UND-QA-MS

DeepSeek R1 (DeepSeek-AI, Guo, et al., 2025). A series of open-weight LLMs developed by DeepSeek-AI specifically optimized for complex reasoning tasks via reinforcement learning (RL). It introduces a multi-stage training process incorporating cold-start supervised data alongside RL. DeepSeek R1 with a MoE architecture sized to 671 billion parameters with 37 billion activated per token achieves an enhanced reasoning performance comparable to proprietary models, such as `OpenAI-o1-1217`(OpenAI, 2024b). Additionally, the DeepSeek-R1 series includes several smaller distilled models, ranging from 1.5B to 70B parameters, based on Qwen and Llama architectures. These distilled models effectively transfer reasoning capabilities from the larger DeepSeek-R1. We select five models from this series: `DeepSeek-R1-Distill-Qwen-1.5B`, `DeepSeek-R1-Distill-Qwen-7B`, `DeepSeek-R1-Distill-Qwen-14B`, `DeepSeek-R1-Distill-Qwen-32B` and `DeepSeek-R1-0528` (the 671B MoE model, through API.[4])

DeepSeek V3 (DeepSeek-AI, Liu, et al., 2024). A powerful open-weight MoE LLM by DeepSeek-AI comprising 671 billion parameters with 37 billion activated per token. It employs Multi-head Latent Attention (MLA) and DeepSeekMoE architectures to achieve cost-effective training and efficient inference. Trained on 14.8 trillion diverse tokens, the performance of DeepSeek-V3 rivals leading closed-source counterparts, particularly excelling in coding, mathematical reasoning, and multilingual tasks. Additionally, a must-mention feature of this model is its economic efficiency in achieving the SotA performance: it requires only 2.788 million H800 GPU hours for full training. For the current experiment, we utilize `DeepSeek-V3-0324` (the 671B MoE model, through API.[5])

Llama 3.2 (Meta, 2024b; Meta, 2024a). Meta's first multimodal open-weight model series optimized for diverse applications, including dialogue, knowledge retrieval, summarization, and mobile AI assistants. The series includes lightweight text-only variants (1B & 3B parameters) ideal for on-device

[4] https://platform.deepseek.com/usage

[5] https://platform.deepseek.com/usage

Query	Gold Judgment	Direct Source	Annotation
How many seasons does Narcos have in total?	fully specified	CLAMBER	-
When was the last time World Cup was in US?	underspecified	CLAMBER	Which one: the date of the Men's FIFA, the date of the Women's FIFA, the year of the Men's FIFA, or the year of the Women's FIFA?
Who was the president of the United States in 2000?	fully specified	IN3	-
Find the cheapest flight from Los Angeles to Tokyo.	underspecified	IN3	The task is clear in terms of the route (Los Angeles to Tokyo) and the preference for the cheapest option. However, it lacks details such as the desired travel dates, if the user is looking for a one-way or round-trip ticket, and any flexibility in travel dates that could affect the price.
who was the ruler between 1912 and 1927 in Morocco	fully specified	CoCoNot	-
who is the present chief of army	underspecified	CoCoNot	I'm sorry, but without specifying the country you are referring to, I'm unable to provide you with accurate information about the current Chief of Army. The title and position of Chief of Army can differ by country and each nation's military structure. Please provide the name of the country whose army chief you are inquiring about.
When was the first person sent to space?	fully specified	AmbigNQ	-
When did the apple tv 4k come out?	underspecified	AmbigNQ	Q1: When did the Apple TV 4K announcement come out? A1: September 12, 2017. Q2: When was Apple TV 4K released? A2: September 22, 2017.

Table 3.6: Examples from the multi-source test set **UND-QA-MS** with gold annotations from respective source datasets.

tasks with 128K-token contexts, and larger vision models (11B & 90B) designed for image understanding and reasoning. Employing training techniques like supervised fine-tuning (SFT), reinforcement learning with human feedback (RLHF), and quantization methods (SpinQuant and QLoRA), Llama 3.2 is claimed to excel in performance, scalability, and efficiency. The series provides official support for 8 languages and is suitable for flexible deployment in resource-constrained environments such as mobile devices. For the current experiment, we select `Llama-3.2-3B-Instruct`, one of the text-only variants.

Llama 3.3 (Meta, 2024c). `Llama-3.3-70B-Instruct` is Meta's 70-billion-parameter multilingual large language model optimized for dialogue applications. Trained on over 15 trillion tokens, it supports multilingual text and code generation across 8 languages and offers 128k context length with Grouped-Query Attention (GQA) for enhanced scalability. Fine-tuned through supervised and reinforcement learning with human feedback (RLHF), it achieves competitive results on industry benchmarks. We include `Llama-3.3-70B-Instruct` for the current experiment.

3.4 A Working Taxonomy of Underspecified Queries in QA

In this section, we propose a working taxonomy of underspecified queries in QA scenarios, combining insights from both theoretical linguistics (Kennedy, 2011; Sennet, 2023; Nieuwland and Van Berkum, 2008; Sorensen, 2023; van Rooij, 2011; Carston, 2002; Belleri, 2014; Bunt, 2007; Egg, 2010) and an analysis of empirical data from the datasets introduced above (Zhang, Qin, et al., 2024; Qian et al., 2024; Brahman et al., 2024; Min et al., 2020). More specifically, we first came up with the initial set of taxonomic categories aligning with the various theoretically-based categories discussed in Section 2.1 and then started to validate them against a subset of data sampled from all the source datasets introduced previously that is not included in **UND-QA-MS** (to avoid overfitting). We conducted a manual annotation process to dynamically adjust taxonomic categories based on data distribution, aiming to balance the number of categories, reduce overlapping, and ensure complete coverage of all observed data points. This includes waiving some theoretical categories rarely observed in empirical data and blending some categories that lead to similar practical effects. For instance, we excluded syntactic ambiguity and pragmatic ambiguity in this working taxonomy, as they are seldom observed to be the source of underspecified queries in the sampled QA data; we also combined lexical ambiguity and referential ambiguity into one category, since both types lead to underspecified concept/entity/topic reference in practice. Additionally, we avoided including theoretical terminologies of linguistics, but used more intuitive descriptions in the naming of categories for a better applicability to LLMs, as we found in trial experiments that the models are not necessarily equipped with accurate concepts of linguistics in their “knowledge”. Eventually, we identified a working taxonomy with four categories of underspecified queries that are commonly found in QA scenarios, presented as follows:

1. Missing necessary components. An underspecified query under this category contains at least one expression that is missing a commonly expected component conceptually tied to it (e.g., an implicit expected argument of a predicative element). As a result, the semantic content of the expression at issue is linguistically incomplete and undertermined, with several different interpretations possible. As a result, the interpretation of the whole query is subject to underspecification. This category is closely related to *Missing Constituents* or *Conceptual Truncation* discussed in the framework of Linguistic (Semantic) Underdeterminacy (Carston, 2002; Belleri, 2014).

We present an example query from this category:

(1) Ok Google, what’s the capital? (of which country?) **[Taken from CoCoNot]**

2. Undetermined lexicons or references. An underspecified query under this category contains at least one expression with lexical or referential ambiguity. Multiple same-level concepts or entities can be mapped to this expression at issue to serve as potential lexical entries or referents. It is impossible to fully determine which one is intended by the user based on the provided content. As a result, the undeterminacy of this expression leads to multiple possible interpretations of the query, rendering it underspecified. This category is closely related to *lexical ambiguity* and *referential ambiguity* (Kennedy, 2011; Sennet, 2023; Nieuwland and Van Berkum, 2008), the concept of *indexical reference* in the Linguistic Underdeterminacy framework (Carston, 2002), and the *Syntactically but not semantically homogeneous ambiguities* discussed as a type of semantic underspecification in Egg (2010).

We present an example query from this category:

(2) When was the last time the Giants went to the playoffs?
(the Giants: the football team or the baseball team?) **[Taken from CLAMBER]**

3. Undetermined perspective or granularity. An underspecified query under this category contains at least one expression where the general meaning is in place, but its specific interpretation can still vary based on different perspectives or granularity levels adopted. Multiple interpretations of different natures or levels are plausible for such an expression, and it's impossible to fully determine which one is intended based on the provided content. As a result, the undeterminacy of this expression leads to multiple possible interpretations of the query, rendering it underspecified. This category resonates with the "green" examples we analyzed in Section 2.1, under "lexical ambiguity" and "missing constituents" respectively. Aligning with the view of Belleri (2014) (see Section 2.1.3, under "missing constituents"), we hold that it is not a type of lexical ambiguity as Kennedy (2011) and Kennedy and Mcnally (2010) claims and it is also not a type of vagueness, as plausible perspectives or granularity levels of interpretation discussed in this category are more definite and objectively acknowledged, instead of being completely a matter of contextual/subjective standard. In the meantime, we also claim that this category is diverging from prototypical cases of "missing constituents", as the linguistic intuition in cases under this category is more about inner interpretation of an expression at issue instead of lacking external elements for the complete semantic saturation. Together with the observation that many underspecified user queries in QA originate from this specific issue, we regard it as an independent category in our working taxonomy.

An example query from this category is presented as follows:

(3) When was the First World War broke out?
("broke out" in the political sense or in the military sense?) **[Taken from AmbigNQ]**

4. Undetermined standard or preference. An underspecified query under this category contains at least one expression where the general meaning is in place, but its specific interpretation is vague due to unspecified contextual standards or subjective criteria. A wide range of fine-grained interpretations is possible based on contextual or subjective needs, and it's impossible to fully determine which one is intended by the user based on the provided content. As a result, the undeterminacy of this expression leads to many, or even an infinite number of, possible interpretations of the query, rendering the entire query underspecified. This category is closely related to the prototypical *vagueness* Kennedy (2011) and *semantic imprecision* proposed by Bunt (2007). In the Linguistic (Semantic) Underdeterminacy framework, phenomena discussed under *adjustments (overspecifying/underspecifying) of linguistically encoded concepts* (Carston, 2002) and *gradable expressions depending on standards or comparison classes* (Belleri, 2014) can also be attributed to this category.

We present an example query from this category:

(4) Recommend the best smartwatches available in 2023.
(what's the specific standard of being "best"?) **[Taken from IN3]**

If none of the above classes can be applied to a query, then the query is considered fully specified. It is also worth noting that these taxonomic categories are not mutually exclusive, and each query can be attributed to more than one category. We argue that this is not problematic for our present work, as our primary goal is to use this taxonomy as a supervision approach to "teach" models about what constitutes underspecified queries, instead of focusing on classifying queries into these categories accurately. In this light, our taxonomy emphasizes capturing underspecified queries with proposed categories exhaustively, while attributing each query to one and only one category is not necessary.

Overall, by introducing the taxonomic framework above, which is grounded in linguistic theory and encompasses a broader range of data points from multiple datasets examining different phenomena that lead to underspecified queries in QA, the incompatibility between different definitions and taxonomies in previous NLP research (as introduced in 2.2) can be alleviated. This taxonomy serves as the backbone for developing the intended LLM-based classifier through a prompt-based method, where its definitions can serve as textual demonstrations in specific prompt settings, and examples annotated based on it can function as in-context learning data. We will turn to the details of this in the next chapter.

Chapter 4

Towards A Theory-Informed LLM-based Classifier to Detect Underspecified Queries

In this chapter, we experiment with developing an LLM-based classifier for identifying underspecified queries in QA more accurately. We evaluate all the model performance on the **UND-QA-MS** dataset introduced in Section 3.2, which is derived from multiple existing benchmarks curated for research on different types of underspecified queries in QA.

In the first part, we evaluate the capabilities of several off-the-shelf SotA open-weight LLMs for detecting underspecified queries in QA. As mentioned in Section 2.2.2, previous research shows that the detection performance of underspecified queries in conversational QA is still not optimal with LLMs such as `Llama-2-13B-Chat`, `Llama-3.1-70B` and `GPT-3.5-Turbo`. However, provided the latest wave of LLM advancements in 2024 and 2025 (Shakudo, 2025; Cardillo, 2025), it is timely to revisit this task using a selection of the latest models nominated in Section 3.3.

Following the test of off-the-shelf LLMs, we integrate the working taxonomy introduced in Section 3.4 to into the prompting and verify whether this appraoch brings a positive effect in the LLM detection of underspecified queries compared to off-the-shelf performance, based on the hypothesis that the taxonomy-related information can serve as the supervision for more accurate model classifications of fully specified (FS) and underspecified (UND) queries. More specifically, we incorporate textual demonstrations and in-context learning (Dong et al., 2024) examples derived from the taxonomy to guide the model's classification behavior, and the goal is to improve LLM-based detection of underspecified queries in QA by leveraging insights from theoretical works of linguistics.

4.1 Testing Off-the-Shelf LLMs

The first part of our experiment aims to evaluate the capabilities of the selected off-the-shelf SotA LLMs in detecting underspecified queries QA without inputting any external supervision.

4.1.1 Experimental Setup

We prompt the LLMs listed in Section 3.3 to perform binary classification tasks for all queries in **UND-QA-MS** regarding whether each query is "underspecified" (UND) or "fully specified" (FS), and to provide reasoning texts that justify the classifications. The prompting was carried out using two formats: the natural language (NL) prompt and the DSPy prompt (Khattab et al., 2024; The DSPy

Team, 2025). No external input regarding the conception of UND or FS is provided. This results in 20 runs across 12 selected models with varying parameter sizes.

Firstly, we designed a natural language (NL) prompt instructing the model to act as an expert analyst and classify input queries as either FS or UND. The prompt includes a task description, the target query structured as a JSON entry, and an explicit requirement for an output schema in JSON format. Furthermore, models exhibiting the highest accuracy with the NL prompt[1] were selected for the implementation of the DSPy prompting. DSPy is a declarative programming model designed to optimize prompting and pipeline development for language models. It abstracts Language Model (LM) pipelines into structured text transformation graphs using parameterized modules, which encapsulate prompting techniques like Chain-of-Thought (Wei, Wang, et al., 2023) as reusable computational components, each defined through *signatures*. Additionally, its compiler automatically optimizes module parameters (e.g., prompts, demonstrations) guided by provided performance metrics (Khattab et al., 2024; The DSPy Team, 2025). Using DSPy, we defined two signatures that specify the input (a string of the target query) and output fields (the FS/UND classification label), along with type and format constraints.[2] We experimented with these signatures in both the `Predict` and `ChainOfThought` execution modules provided by DSPy, with the former being the default and the latter outputs an extra field of "reasoning".

4.1.2 Results

This section overviews the off-the-shelf performance of selected LLMs on the "UND/FS" classification task based on **UND-QA-MS**. We report per-class F1 scores (FS F1 & UND F1), overall accuracy, and the macro F1 scores.

It is worth noting that the performance on the AmbigNQ subset of **UND-QA-MS** consistently hovered around the chance level (accuracy: 0.43-0.61; macro F1: 0.37-0.59) across all runs, which suggests that the involved LLMs were unable to capture much discriminative signal from AmbigNQ data points. Considering this, in the main tables, we only report results obtained from other subsets of **UND-QA-MS** for which the selected models demonstrate a non-random discriminative ability.[3] We speculate two plausible factors leading to the cross-model unsatisfactory performance observed on the AmbigNQ subset: (1) the existence of inter-annotator inconsistency in the bottom-up crowdsourcing despite the validation efforts; (2) the fact that annotators were encouraged to identify all plausible answers within a knowledge-intensive register (i.e., Wikipedia) may have inflated the presence of multiple possible answers and led to an analytical granularity of ambiguity/underspecification not aligned with other datasets and the tested models.

Table 4.1 summarizes results using the designated natural language (NL) prompt. In terms of overall accuracy, several runs using the NL prompt reach a similar performance ceiling around 0.71. Specifically, `Qwen3-4B`, `Llama 3.3-70B-Instruct` and both `DeepSeek` MoE models loaded using API (`V3-0324` and `R1-0528`) achieve top accuracy scores of 0.71. Among them, `DeepSeek-R1-0528-671B (API)` stands out as the most balanced across all metrics (FS F1, UND F1, macro F1, and accu-

[1] We exclude models loaded using the commercial API from this selection due to financial constraints.

[2] The two signatures only have a minor wording difference in the naming and description of the input field for the target query. One signature sets up this field as `'request' (str): An input user question/request`, while the other sets it up as `'query' (str): An input user query`. The motivation for having two signatures with this minor difference is to explore the potential effect brought by the wording of the generated prompts. Please refer to Appendix A for detailed sample prompts.

[3] Please refer to Table B.1 in Appendix B for the detailed results on the AmbigNQ subset.

	Qwen3-4B NL	Qwen3-8B NL	Qwen3-14B NL	Qwen3-32B NL
FS F1	0.69	0.63	0.72	0.70
UND F1	0.72	0.71	0.67	0.69
accuracy	**0.71**	0.68	0.70	0.70
macro F1	0.70	0.67	0.70	0.70
	DeepSeek-R1-Distill-Qwen-1.5B NL	DeepSeek-R1-Distill-Qwen-7B NL	DeepSeek-R1-Distill-Qwen-14B NL	DeepSeek-R1-Distill-Qwen-32B NL
FS F1	0.33	0.62	0.66	0.61
UND F1	0.66	0.62	0.70	0.71
accuracy	0.55	0.62	0.68	0.67
macro F1	0.49	0.62	0.68	0.66
	DeepSeek-V3-0324-671B (API) NL	DeepSeek-R1-0528-671B (API) NL	Llama-3.2-3B NL	Llama-3.3-70B NL
FS F1	0.74	0.72	0.09	0.76
UND F1	0.67	0.71	0.67	0.64
accuracy	**0.71**	**0.71**	0.51	**0.71**
macro F1	0.70	0.71	0.38	0.70

Table 4.1: An overview of the performance on **UND-QA-MS (excl.AmbigNQ)** across the selected LLMs using the natural language (NL) prompt.

	Qwen3-4B DSPy-Predict W1	Qwen3-4B DSPy-CoT W1	Llama-3.3-70B-Instruct DSPy-Predict W1	Llama-3.3-70B-Instruct DSPy-CoT W1
FS F1	0.72	0.71	0.69	0.69
UND F1	0.65	0.68	0.64	0.68
accuracy	0.69	0.70	0.67	0.68
macro F1	0.69	0.70	0.67	0.68
	Qwen3-4B DSPy-Predict W2	Qwen3-4B DSPy-CoT W2	Llama-3.3-70B-Instruct DSPy-Predict W2	Llama-3.3-70B-Instruct DSPy-CoT W2
FS F1	0.71	0.72	0.64	0.71
UND F1	0.65	0.68	0.65	0.69
accuracy	0.68	0.70	0.65	0.70
macro F1	0.68	0.70	0.65	0.70

Table 4.2: An overview of performance on **UND-QA-MS (excl.AmbigNQ)** for `Qwen3-4B` and `Llama-3.3-70B`, utilizing DSPy prompts under two execution modules (`Predict` and `ChainOfThought [CoT]`) and two different wording variants (W1 and W2).

	Qwen3-4B NL		Qwen3-32B NL		Qwen3-4B DSPy-CoT W1	
	Accuracy	Macro F1	Accuracy	Macro F1	Accuracy	Macro F1
CLAMBER	0.65	0.64	0.64	0.63	0.64	0.63
IN3	0.73	0.73	0.72	0.72	0.71	0.71
CoCoNot	**0.81**	**0.78**	**0.81**	**0.79**	**0.83**	**0.81**
	Qwen3-4B DSPy-CoT W2		DeepSeek-R1-Distill-Qwen-14B NL		DeepSeek-V3-0324-671B(API) NL	
	Accuracy	Macro F1	Accuracy	Macro F1	Accuracy	Macro F1
CLAMBER	0.65	0.64	0.66	0.66	0.63	0.62
IN3	0.70	0.70	0.65	0.64	0.74	0.72
CoCoNot	**0.84**	**0.83**	**0.78**	**0.74**	**0.85**	**0.83**
	DeepSeek-R1-0528-671B (API) NL		Llama-3.3-70B-Instruct NL		Llama-3.3-70B-Instruct DSPy-CoT W2	
	Accuracy	Macro F1	Accuracy	Macro F1	Accuracy	Macro F1
CLAMBER	0.66	0.65	0.61	0.58	0.65	0.64
IN3	0.75	0.74	0.78	0.76	0.74	**0.74**
CoCoNot	**0.81**	**0.78**	**0.82**	**0.81**	**0.75**	0.73

Table 4.3: Accuracies and macro F1 scores by source sets of the selected runs.

racy), although its margin over some smaller distilled `DeepSeek` models (`DeepSeek-R1-Distill-14B` and `DeepSeek-R1-Distill-32B`) remains relatively small. Furthermore, the relative performance of the models on the FS and UND subsets does not follow a uniform pattern. Several settings with small-sized and mid-sized models (e.g., `DeepSeek-R1-Distill-Qwen-1.5B NL`, `Qwen3-8B NL`, `DeepSeek-R1-Distill-Qwen-32B NL`) tend to achieve higher F1 scores on the UND subset than on the FS subset. In contrast, settings on larger-scale models such as `Llama-3.3-70B NL` and `DeepSeek-V3-0324-671B (API) NL` demonstrate the opposite trend, with notably higher F1 scores on the FS subset than on the UND subset. In addition, parameter scaling does not yield consistent performance gains in the current task. The `Qwen3` series, for instance, shows little to no improvement from 4B to 32B under NL prompting, and even exhibits minor fluctuations across sizes. Similarly, the distilled `DeepSeek-R1` models plateau after 14B, with no gains from further scaling. In contrast, `Llama-3` models do exhibit some positive correlation between model size and performance, revealed in the comparison between `Llama-3.2-3B` and `Llama-3.3-70B`.

Based on the results illustrated above, we selected `Qwen3-4B` and `Llama-3.3-70B`, the two locally deployment-friendly models that obtained top performance under the NL prompting, to evaluate their performance with DSPy prompting. Table 4.2 shows the performance of two selected models under DSPy prompting with two execution modules (`Predict` and `ChainOfThought [CoT]`) and two signature wording variants (W1 and W2).[4] Across both execution modules and signature wording variants, DSPy prompting did not yield notable improvements over the results obtained from the NL prompt. The best accuracy scores under DSPy prompting were slightly lower (by 1 point), suggesting no clear advantage of DSPy prompting for this binary classification task under current configurations.

[4] W1 is the variant with the input field set as `'request' (str): An input user question/request`; W2 is the variant with the input field set as `'query' (str): An input user query`.

4.1.3 Analysis

To better understand the performance plateau around an accuracy of 0.70, we conducted some more fine-grained analysis of the results from testing off-the-shelf LLMs.

Source Dataset Analysis. We selected nine representative runs (with both NL and DSPy prompting) that demonstrated relatively strong performance (accuracy and macro F1 within the range of 0.68-0.71). For each run, we report accuracy and macro F1 scores across the three included source datasets: CLAMBER, IN3, and CoCoNot (see Table 4.3). The results reveal a consistent trend for almost all runs: performance is the highest on the CoCoNot subset, followed by IN3, while the lowest scores are observed on CLAMBER. However, it is important to note that the three source subsets differ in size (see Table 3.5) and the raw scores in Table 4.3 should be interpreted with caution. To address this issue, we apply statistical tests to evaluate whether the observed differences are significant and robust.

We conducted chi-square (χ^2) tests for each of the nine runs to evaluate whether the difference in accuracy between the three subsets is statistically significant, and the results are all positive ($p < 0.01$), confirming that model accuracies vary meaningfully across source datasets. To further determine which subsets differ, we also conduct pairwise chi-square tests with Bonferroni correction. Results show that the accuracy difference between **CoCoNot and CLAMBER** is consistently significant across all models, except for `Llama-3.3-70B-Instruct DSPy-CoT W2`. Besides, **IN3 vs CLAMBER** is significant in nearly half of the cases, while **CoCoNot vs IN3** yields marginal significance. Overall, these findings still suggest a performance hierarchy in which CLAMBER tends to underperform relative to CoCoNot and IN3, with the most pronounced and consistent contrast observed between CoCoNot and CLAMBER.

In addition, bootstrapped 95% confidence intervals (CIs) for macro F1 reveal a consistent pattern: CoCoNot consistently achieves the highest macro F1 scores, followed by IN3 (with substantial overlap in some cases), and CLAMBER is the lowest across all runs except for `DeepSeek-R1-Distill-Qwen-14B NL`. In most settings, macro F1 CIs of CLAMBER are distinctly lower than those of CoCoNot and IN3, further corroborating a significant and stable performance gap.[5]

This analysis highlights how differences in the underlying characteristics of each source subset can influence model performance. IN3 and CoCoNot focus on relatively narrow dimensions of underspecification (vagueness in the former and the missing of spatiotemporal information in the latter), making their data points more homogeneous in scope. In contrast, CLAMBER was constructed using a top-down taxonomy, which introduces a broader range of underspecification types and, consequently, poses a greater challenge to the models. More fundamentally, the variation of dataset focus can lead to differences in how underspecified queries are defined and annotated across multisource data. While LLMs may have developed an implicit standard of underspecification in pretraining, their judgments may not transfer reliably to datasets with finer-grained definitions or diverging annotation guidelines. This mismatch can result in inconsistent performance across different datasets and may partially explain why models struggle to achieve higher scores on more diverse evaluative data, which is manifested as a performance ceiling.

Qualitative Analysis. We performed a qualitative analysis by examining cases where model predictions (from various runs) and gold annotations from the original datasets are not aligned. We

[5] Please refer to Table C.2 and Table C.1 in Appendix C for detailed results of the statistical tests

#	Query	Gold	Pred	Model Reasoning
1	Tell me which continent the following cities are in: - Seattle - Helsinki - Lagos - Osaka - Istanbul	FS	UND	The query requests identification of continents for five cities. Four of the cities (Seattle, Helsinki, Lagos, Osaka) are unambiguously located in distinct continents (North America, Europe, Africa, Asia). However, Istanbul is geographically situated at the boundary between Europe and Asia, creating ambiguity. (...) – Qwen3-4B NL
2	Give me a list of the ten best rock guitarists of all time	FS	UND	The query asks for a list of the ten best rock guitarists, but it does not specify the criteria for determining who is considered 'best.' (...)– DeepSeek-R1-Distill-Qwen-14B NL
3	Find the population of Australia in 2021.	FS	UND	The query specifies the country (Australia) and the year (2021), but it does not specify the source of the data, the exact time within the year, or whether the population figure is an estimate or based on a census. (...) – DeepSeek-R1-Distill-Qwen-14B NL
4	states in the northeast region in us	FS	UND	The query seeks states in the 'northeast region' of the U.S., but fails to explicitly define what constitutes the 'northeast region.' While commonly understood as comprising New England and Mid-Atlantic states (e.g., Maine, New Hampshire, Massachusetts, etc.), the term 'region' lacks formal geographic boundaries. (...) –Qwen3-4B NL
5	when did we begin following a monotheistic religion in rome	FS	UND	The query is asking about the beginning of a monotheistic religion in Rome. However, it is unclear what specific monotheistic religion is being referred to, as there were multiple religions that could be considered monotheistic in Rome at different times. (...) –DeepSeek-R1-Distill-Qwen-14B NL
6	Find a recipe for gluten-free brownies.	UND	FS	(...) It does not leave out essential details needed to perform the search, such as dietary restrictions (gluten-free) and the desired dish (brownies). No additional information is required to fulfill this request effectively. – DeepSeek-V3-0324-671B (API) NL
7	Find the top 10 best-rated restaurants in New York City.	UND	FS	(...) It does not leave any ambiguity about what is being asked for, as all necessary parameters are clearly defined. The term 'best-rated' is commonly understood to refer to high ratings, typically from customer reviews or critic scores, and does not require further clarification in this context. – DeepSeek-V3-0324-671B (API) NL
8	who is the present chief of army	UND	FS	(...) It implies a singular answer, as there can only be one chief of the army at a given time. The query does not contain any ambiguous terms or open-ended questions (...) – Llama-3.3-70B-Instruct NL
9	When do we celebrate veterans day this year?	UND	FS	(...) Veterans Day is universally recognized as falling on November 11th every year in the United States, making the required information directly retrievable from general knowledge. (...). – Qwen3-4B NL
10	Does Chelyrtodestoni have semiprecocial as its developmental mode?	UND	FS	(...) While the scientific validity of the species name may be questionable (it appears to be a misspelled or fictional taxon), the query itself contains no ambiguities regarding what is being asked. (...) – Qwen3-4B NL

Table 4.4: Example queries where model predictions and gold annotations are not aligned, obtained from various runs with selected model reasoning texts attached.

showcased 10 example queries in Table 4.4 and (non-exhaustively) summarized several categories of potential reasons for the models to produce divergent predictions.

Failure to detect the key element of UND/FS judgment: This category concerns cases where an LLM failed to detect the key element (or its absence) that really determines whether a query should be regarded as UND or FS. Examples 8 and 9 fall into this category. For example, in Example 8, the model's reasoning process included the temporal aspect, which is correctly defined as a specific time point. However, the other important factor to fully specify the query, the geographical location, is absent, and the model failed to detect it. While in Example 9, the missing geographical specification of the query is prematurely "assumed" by the model to be "in the United States", which not only led to a misaligned prediction, but also introduced biased information.

Human-machine misalignment on the conception of underspecification: This category includes cases where the conception of underspecification/full specification differs between LLMs and humans, causing a misalignment. Representative examples for this issue are Example 1 and Example 10. Example 1 reveals that the model took the minor ambiguity in the geographical identity of Istanbul, which is not related to the general clarity of the query per se, as a reason to judge the query as underspecified. This is obviously not aligned with the human conception of whether a query itself is sufficiently specified. Example 10, on the other hand, despite acknowledging the unfamiliarity of the artificial non-existing word "Chelyrttodestoni", judged that it is not a fatal issue for the query clarity. However, a query including an unfamiliar word is intuitively underspecified in human communications, which is only resolved by the explicit introduction or an implicit common ground/context.

Human-machine misalignment on the underspecification threshold: This category focuses on cases where the threshold of underspecification/full specification judgment differs, causing a misalignment. Examples of this category are deemed to be widespread and complicated. The remaining 6 examples shown in Table 4.4 can all be placed under this category. Examples 3 and 6 are more intuitively straightforward: in Example 3, the model was obviously "too greedy" for detailed specifying requirements, leading to a much lower threshold than that of human common sense when judging whether a general factual query is underspecified; in contrast, in Example 6, the threshold set by the model is higher than that of humans in daily life, ignoring a wide range of subjective needs essential for a determined baking recipe.

Other examples are more complicated. Examples 2 and 7 are both related to the superlative adjective "best" that is prone to subjective standards. Human annotations differ between the two cases, while the model predictions were misaligned in both. The human annotation for Example 2 is FS ("fully specified"), possibly because most people, influenced to some extent by a common narrative of popular culture, would recognize the existence of a commonly acknowledged ranking of rock guitarists. While for Example 7, this is likely because there are multiple acknowledged methods of rating restaurants, and the rankings can vary based on subjectively chosen methods, the human annotation is provided as UND ("underspecified"). However, it is indeed difficult to clearly determine when the subjectiveness involved reaches the threshold of underspecification, resulting in many borderline cases where LLMs cannot align with the fine-grained human threshold, and even human judgments would also differ interpersonally. Examples 4 and 5 introduce yet another complicated factor of shared socio-cultural knowledge that may affect the UND/FS judgment threshold. People from specific communities/social groups would have sufficient shared common ground, acquired through culture and education, to judge them as FS, but this threshold is not necessarily aligned with LLMs or even other groups of human users.

Confidence Scores. Additionally, we implemented a supplementary experiment, where we prompted `Qwen3-4B` using non-thinking mode to rate confidence scores (notated as c, scaling from 0 to 1) for reasoning texts retrieved from the `Qwen3-4B NL` setting on how confident these reasoning texts support the classification of a user query as "fully specified".[6] Among the results, we focus our qualitative analysis on queries where the previous binary classification doesn't match the confidence score (i.e., "UND" vs. $c > 0.5$; "FS" vs. $c \leq 0.5$). There were 14 queries out of the 855 queries from **UND-QA-MS (excl.AmbigNQ)** where this mismatch occurred, and for nine of them, it was the new confidence score instead of the original prediction that was aligned with the gold annotation. We present some of these examples in Table 4.5. The observation is that in these cases, there is a gap between the informative model reasoning and the binary model prediction. A common feature is that the model reasoning texts contain supporting elements for both the UND and the FS judgment, while their final binary judgments could not maintain these mixed signals and were forced to lean towards one side. In Examples 2 and 3, the negative effect of this phenomenon is particularly obvious, as the model's final conclusions in the reasoning texts contradict the classification label they provided, revealing an internal inconsistency. Considering that the involved queries are mostly borderline cases sensitive to a user's specific information granularity requirement, it's not surprising that they could cause "confusions" for the models. It remains a challenge for LLMs to weigh on supporting evidence on both sides in a similar way to the human cognitive process, remain consistent, and provide a final output convergent with human judgments in such complicated scenarios.

Summary. To summarize, the results from testing off-the-shelf LLMs and the detailed analysis above have shown that the classification of UND/FS queries in QA using a general prompt is not trivial. Several factors may have contributed to the difficulty of this task. Firstly, the heterogeneity of multisource empirical data related to underspecified queries in QA, along with various definitions and taxonomies of various types of underspecification, presents a fundamental challenge. Secondly, the human-machine misalignment on the conception of underspecified queries and the threshold of UND/FS judgments is widespread and complex to handle. Thirdly, weaknesses of LLMs may lead to failures in detecting key features (or the lack thereof) that are essential for correct UND/FS classifications. In complicated, borderline cases, LLMs may suffer from internal inconsistencies and information loss when transforming reasoning content into classification results. Lastly, the prompting techniques used in this test are relatively basic and still have considerable room for improvement.

4.2 Integrating the Proposed Working Taxonomy

In what follows, we aim to verify whether adding theoretical information based on the taxonomy we proposed in 3.4 would improve LLMs' accuracy in the FS/UND classification of queries in QA.

[6] Please refer to Appendix A for the specific prompt.

#	Query	Gold	Pred	Model Reasoning	*c*
1	I'm interested in starting an art collection. Suggest a beginner's guide to art collecting and tips on how to assess the value of artwork.	UND	FS	The query explicitly requests a 'beginner's guide' and 'tips on assessing value,' which covers the primary objectives of providing foundational knowledge and evaluation methods. While the query lacks specificity regarding the type of art (e.g., paintings, sculptures, digital media), geographic focus, or financial constraints, these omissions do not render the request underspecified. The core intent—to educate a novice collector on initiating a collection and understanding artistic value—is clear and actionable. The absence of granular details does not prevent the generation of generalized yet relevant guidance, as the user's overarching goal remains well-defined.	0.5
2	When was the last time world cup was in us?	UND	UND	The query seeks information about the last occurrence of a 'World Cup' in the United States. While 'World Cup' typically refers to the FIFA World Cup (football/soccer), the query does not explicitly clarify the sport or competition. Additionally, the phrase 'was in us' uses informal language ('us') instead of the official name 'United States.' This creates minor ambiguity, as the user may intend different interpretations (e.g., non-FIFA World Cups, regional variations). However,the core intent appears focused on the FIFA World Cup due to its global prominence. Despite these subtleties, the query provides sufficient context to identify the intended subject.	0.75
3	which law regulates the business contracts in india	FS	UND	The query asks for the specific legal framework governing business contracts in India. While the Indian Contract Act, 1872, is the primary statute regulating contracts in India, the query does not clarify whether it seeks a broader range of regulations (e.g., sector-specific laws like the Companies Act, 2013, or international treaties) or focus solely on general contract law. Additionally, 'business contracts' could encompass various subtypes (e.g., commercial, employment, real estate), which may fall under different statutory provisions. However, the core requirement to identify the principal regulatory law (the Indian Contract Act) is explicitly stated.	0.65
4	My home office feels cluttered. Suggest ways to organize and make it a more productive space.	UND	FS	The query clearly states the problem (cluttered home office) and requests actionable solutions (organization and productivity improvement). While it lacks specificity about the nature of the clutter (e.g., physical objects, digital files, ergonomics) or the user's specific needs (e.g., remote work, creative projects), the core requirements are well-defined. General organizational principles (storage solutions, decluttering, workspace optimization) and productivity tactics (time management, ergonomics) can be applied universally. The absence of detailed context does not render the query unsolvable, as standard advice applies to most scenarios.	0.5

Table 4.5: Example queries where the previous binary classification from the `Qwen3-4B NL` run doesn't match the confidence score provided by `Qwen3-4B` non-thinking mode, based on previous reasoning texts output by the `Qwen3-4B NL`.

4.2.1 The In-Context Learning Set (ICLS)

Based on the working taxonomy proposed in Section 3.4, we manually curated an in-context learning dataset (ICLS) with 150 example queries, each annotated with the gold FS/UND judgment, the taxonomic category (or categories) to which it belongs, and the reasoning behind the category assignment. This dataset is designed to provide theoretical insights into FS/UND query classification, accompanied by practical examples. Below, I explain the detailed process of its curation.

The sampling was performed on the same four source datasets of **UND-QA-MS**: CLAMBER, IN3, CoCoNot, and AmbigNQ. We filtered out any example that was already included in **UND-QA-MS** to prevent data leakage. The initial sampling amount is larger than 150 to provide reasonable redundancy, making it possible to exclude examples whose gold labels are found to be controversial in the later annotation process. Based on gold FS/UND classifications and the reasoning texts provided in the original datasets, we implemented the annotation in two steps: (1) we assigned one or more category labels based on the working taxonomy in Section 3.4 for each UND query and the "fully specified" category label for each FS query; (2) we annotated the reasoning process leading to the taxonomic category assigned to each query and the final UND/FS judgment. For better efficiency, we utilized GPT-4o (OpenAI, Hurst, et al., 2024) as a writing assistant to extend manually drafted key points into complete reasoning texts. Eventually, the **ICLS** dataset includes 100 UND examples and 50 FS examples, with the detailed distribution shown in Table 4.6. Example data points are also showcased in Table 4.7.

Class/Source	**CLAMBER**	**IN3**	**AmbigNQ**	**CoCoNoT**	**Class Total**
# Fully Specified (FS)	23	14	15	0	51
# Underspecified (UND)	23	30	23	23	99
# Source Total	45	44	38	23	150

Table 4.6: The data distribution of **ICLS**, the in-context learning set.

4.2.2 Experimental Setup

With the working taxonomy and the ICLS dataset in place, we implement experiments integrating them into LLM prompting to test whether such integrations lead to an LLM-based classifier with a higher accuracy in the FS/UND classification for queries in QA.

With results from testing off-the-shelf LLMs (see Table 4.1 and Table 4.2), we concluded that `Qwen3-4B` is the smallest model that achieved the best overall accuracy (0.71). Therefore, we focused on this model to experiment with a list of taxonomy-involved prompt settings below (See also Table 4.8):

- `FULL`: The definitions of all categories in the taxonomy are integrated (with simplification) into the prompts as detailed textual demonstrations, together with the 150 annotated examples from the **ICLS** dataset. The model is prompted to output the taxonomic category, the final classification, and the reasoning texts based on both definitions and examples.

- `DEF`: Only the definitions of all categories in the taxonomy are integrated (with simplification) into the prompts as detailed textual demonstrations. No example from the **ICLS** dataset is provided. The model is prompted to output the taxonomic category, the final classification, and the reasoning texts only based on the provided definitions.

#	Query	Gold	Category	Reasoning
1	ok google what's the capital	UND	1.Missing necessary components	The request is underspecified because the concept of “capital” is inherently tied to a specific political or administrative entity (e.g., country, state), but the question omits this essential component, making it impossible to determine which capital is being asked about.
2	When was the last time the giants went to the playoffs?	UND	2.Undetermined lexicons or references	The request is underspecified because “the Giants” can refer to multiple parallel referents, such as the New York Giants football team or the San Francisco Giants baseball team, and the correct answer depends entirely on which team is meant.
3	When was the first world war broke out?	UND	3.Underdetermined perspective or granularity	The phrase “broke out” can be interpreted from multiple perspectives or granularities. It can be interpreted from the perspective of the political trigger event, which is the assassination of Archduke Franz Ferdinand, or from the perspective of the formal military actions and declarations of war between major powers. The question remains underspecified before a certain perspective is chosen.
4	Recommend the best smartwatches available in 2023.	UND	4.Undetermined standard or preference	The request is underspecified because the term “best” relies on a subjective standard or preference, such as whether the user values battery life, design, fitness tracking, price, or brand reputation. Without specifying which criteria matter most, the recommendation remains open-ended.
5	Locate the nearest yoga class with the best reviews in my city.	UND	2.Undetermined lexicons or references; 4.Undetermined standard or preference	The request does not clarify which city is meant, leaving the referent of "my city" undefined. Additionally, terms like “nearest” and “best reviews” are subjective and can vary based on distance thresholds, review platforms, or rating standards, making the user's evaluative preferences unclear. Both factors make the request underspecified.
6	List out the ten most spoken languages in the world.	FS	fully specified	This query is fully specified because it clearly asks for a factual ranking (top ten languages) based on global speaker numbers, which is a well-documented statistic. While exact rankings may vary slightly depending on data sources, the request at the macro level is clear and does not involve ambiguous references or subjective preference/criteria.

Table 4.7: Example queries taken from **ICLS** with different annotated categories.

Setting/Components	Category Definitions	Category Instructions	ICLS examples
FULL	✓	✓	✓
DEF	✓	✓	✗
LIGHT	✗	✓	✓
MINI	✗	✓	✗

Table 4.8: An overview of the taxonomy-involved prompt settings

- `LIGHT`: Instead of integrating detailed definitions of all taxonomic categories into the prompts, only the category names and simple instructions requesting the model to choose from them under appropriate conditions are provided. Meanwhile, the 150 annotated examples from the **ICLS** dataset are also integrated into the prompts. The model is prompted to output the taxonomic category, the final classification, and the reasoning texts based on simplified textual instructions and examples.

- `MINI`: Only the category names from the taxonomy and simple instructions requesting the model to choose from them under appropriate conditions are integrated into the prompts. The model is prompted to output the taxonomic category, the final classification, and the reasoning based on this minimal instructive input.

Similar to the previous test on off-the-shelf LLMs, we tested both formats of natural language (NL) prompting and DSPy prompting. For the latter, we maintained the two signature wording variants (W1 and W2), but we only focused on the `ChainOfThought (CoT)` execution module in the current experiment, since the previous test revealed that the performance obtained with the `CoT` module generally outperformed that from the `Predict` module (see Table 4.2).

Apart from the four settings integrating the theory-based taxonomy we proposed, we also included a controlled prompt setting without integrating any element of the proposed taxonomy, but incorporating examples from the **ICLS** dataset deprived of the author's annotation (i.e., only the example queries and the gold classifications from the original datasets are integrated) for non-taxonomy-driven in-context learning. This measure aims to separate the effects brought purely by the in-context learning technique from the effects genuinely brought by the tuning measures based on the taxonomy we propose.

- `CONTROL`: The 150 examples from the **ICLS** dataset are integrated into the prompts, but they are deprived of the author's annotation related to the proposed taxonomy. Only the example queries and the gold classifications from the original datasets serve as input. The model is prompted to output the final classification and the corresponding reasoning texts.

Overall, we applied four taxonomy-involved prompt settings and one controlled prompt setting to three prompting formats: `NL`, `DSPy W1`, and `DSPy W2`, respectively. This yields experimental results from 12 taxonomy-involved runs and 3 controlled runs.[7]

4.2.3 Results

In parallel with the choice we made in the testing of off-the-shelf LLMs, we excluded the data points from AmbigNQ in our report. We present an overview of the performance on **UND-QA-MS (excl.AmbigNQ)** across the 12 taxonomy-involved runs in Table 4.9. It can be observed that three of them achieve the best performance from the perspective of accuracy, scoring at 0.73: `Qwen3-4B DSPy CoT FULL W1`, `Qwen3-4B DSPy CoT LIGHT W1`, and `Qwen3-4B DSPy CoT FULL W2`. Through horizontal comparisons of the results presented, it is worth noting that all the best-performing taxonomy-involved runs utilized the DSPy `CoT` prompting module. In contrast, for taxonomy-involved runs using NL prompts, none of them yielded an accuracy above 0.70. Another interesting pattern is that all runs showing a performance advantage integrated the examples from the **ICLS** dataset with taxonomy-based

[7] Please refer to Appendix A for detailed sample prompts used in these runs.

	Qwen3-4B NL FULL	Qwen3-4B NL DEF	Qwen3-4B NL LIGHT	Qwen3-4B NL MINI
FS F1	0.72	0.72	0.71	0.56
UND F1	0.63	0.67	0.69	0.7
accuracy	0.68	0.7	0.7	0.64
macro F1	0.68	0.7	0.7	0.63
	Qwen3-4B DSPy CoT FULL W1	Qwen3-4B DSPy CoT DEF W1	Qwen3-4B DSPy CoT LIGHT W1	Qwen3-4B DSPy CoT MINI W1
FS F1	0.73	0.71	0.73	0.67
UND F1	0.73	0.67	0.74	0.67
accuracy	**0.73**	0.7	**0.73**	0.67
macro F1	0.73	0.69	0.73	0.67
	Qwen3-4B DSPy CoT FULL W2	Qwen3-4B DSPy CoT DEF W2	Qwen3-4B DSPy CoT LIGHT W2	Qwen3-4B DSPy CoT MINI W2
FS F1	0.73	0.71	0.7	0.71
UND F1	0.73	0.64	0.71	0.67
accuracy	**0.73**	0.68	0.71	0.69
macro F1	0.73	0.67	0.71	0.69

Table 4.9: An overview of the performance on **UND-QA-MS (excl.AmbigNQ)** across the 12 taxonomy-included prompt settings using `Qwen3-4B`.

	Qwen3-4B NL CONTROL	Qwen3-4B DSPy CoT CONTROL W1	Qwen3-4B DSPy CoT CONTROL W2
FS F1	0.74	0.72	0.73
UND F1	0.65	0.57	0.6
accuracy	0.7	0.66	0.68
macro F1	0.7	0.65	0.67

Table 4.10: The `Qwen3-4B` performance on **UND-QA-MS (excl.AmbigNQ)** under the `CONTROL` setting (taxonomy-excluded in-context learning).

annotations. Comparatively, runs that utilized only the detailed definitions of taxonomic categories (with the `DEF` setting) did not bring obvious benefits, while runs where category names were listed with simple instructions (with the `MINI` setting) even downplayed the performance. Last but not least, results from runs under the `CONTROL` setting (see Table 4.10) show that purely utilizing the combination of DSPy `CoT` prompting module and the in-context learning technique based on examples without any element from the proposed working taxonomy could not bring similar performance improvement, as observed in Table 4.9. However, among all the runs using the NL prompting format, `Qwen3-4B NL CONTROL` yielded a performance similar to the best-performing runs equipped with elements from the proposed taxonomy integrated (the first row of the Table 4.9).

We then focus on the comparison between the three best taxonomy-involved runs and the top-performing runs from the off-the-shelf testing. Figure 4.1 provides a more direct visualization of this comparison. From the accuracy perspective, the three best-performing taxonomy-involved runs show improvements over all the top-performing runs in the off-the-shelf testing. Additionally, their FS F1 scores and UND F1 scores are also higher and more balanced than those of most top-performing runs in the off-the-shelf testing. To further investigate the significance of the observed improvements on the accuracy, we performed a series of paired *t*-tests between best-performing taxonomy-involved runs and

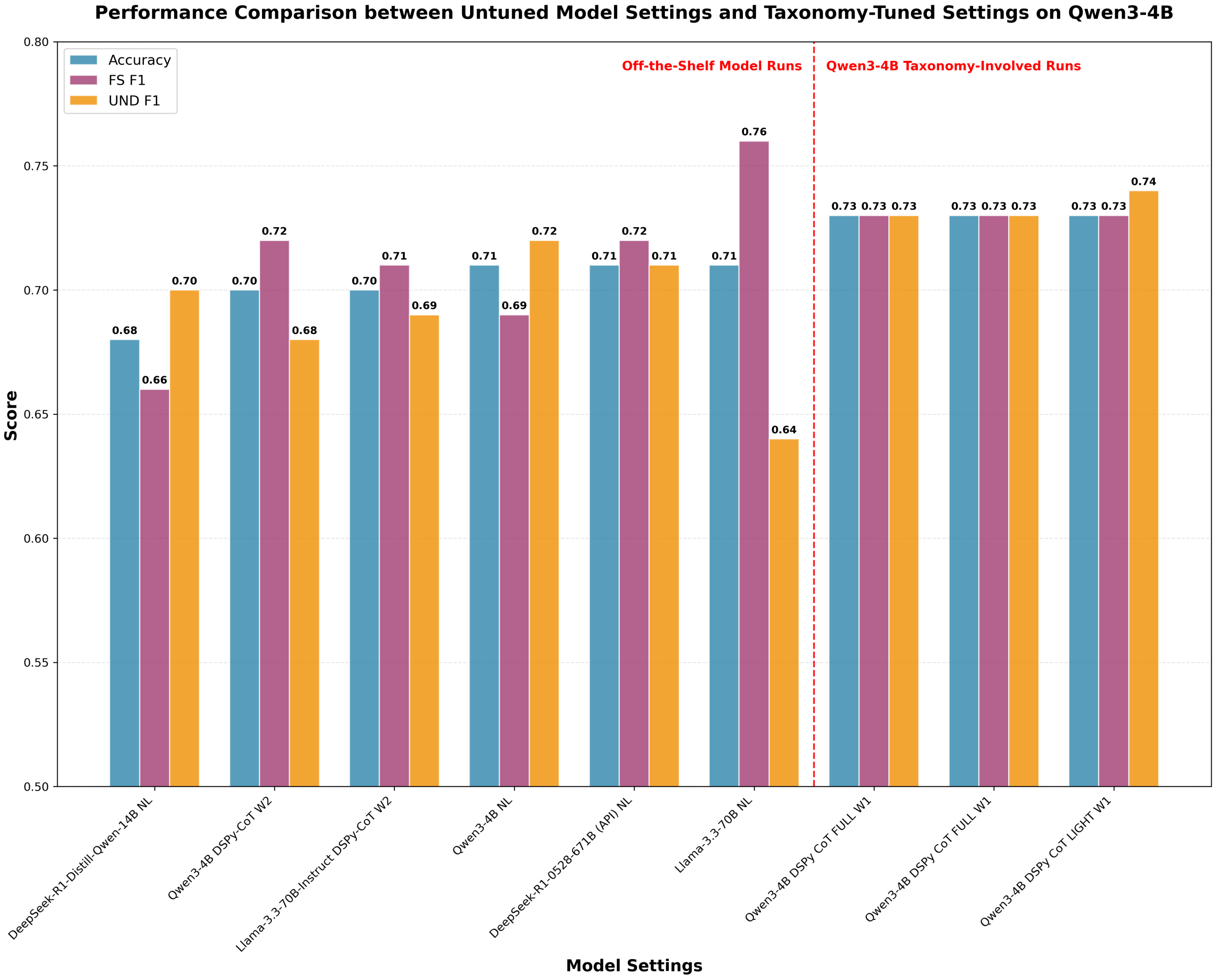


Figure 4.1: The performance comparison between taxonomy-involved runs on `Qwen3-4B` and top-performing runs in the off-the-shelf testing

the top-performing off-the-shelf runs using `Qwen3-4B`, with results presented in Table 4.11. Out of the nine pairs of comparisons, *t*-tests showed that five of them manifest a significant difference. Nearly all taxonomy-involved runs show significant improvements compared to their off-the-shelf counterparts using the DSPy `CoT` module (the second and the third row of Table 4.11). On the other hand, compared to the `Qwen3-4B NL` run from the off-the-shelf testing, the improvements brought by the taxonomy-involved runs are not statistically significant (the first row of Table 4.11), but `Qwen3-4B DSPy CoT FULL W2` and `Qwen3-4B DSPy CoT LIGHT W1` are relatively close to the significance threshold ($< .05$).

Lastly and also importantly, among the three best taxonomy-involved runs discussed, `Qwen3-4B DSPy CoT LIGHT W1` shows a minor advantage, as its UND F1 score (0.74) is slightly higher than those of the other two (0.73), and its accuracy improvement is significant compared to both counterparts from the off-the-shelf testing using `Qwen3-4B` and applying the DSPy `CoT` module.

4.2.4 Analysis

Overall remarks. The results shown in Figure 4.1 and Table 4.11 reveal that integrating the proposed theory-driven working taxonomy of underspecified queries in QA with DSPy prompting using `CoT` module improves LLMs' accuracy on the FS/UND classification of user queries in QA.

Off-the-Shelf/Taxonomy-Involved	Qwen3-4B DSPy CoT FULL W1	Qwen3-4B DSPy CoT FULL W2	Qwen3-4B DSPy CoT LIGHT W1
Qwen3-4B NL	$t = 1.65, p = .100$	$t = 1.79, p = .074$	$t = 1.80, p = .072$
Qwen3-4B DSPy-CoT W1	$t = 2.20, p = .028$	$t = 2.25, p = .025$	$t = 2.30, p = .022$
Qwen3-4B DSPy-CoT W2	$t = 1.97, p = .049$	$t = 1.93, p = .054$	$t = 2.02, p = .043$

Table 4.11: The paired t-test results from the comparison between the three best-performing taxonomy-involved runs with `Qwen3-4B` and three top-performing runs from the off-the-shelf testing using `Qwen3-4B`.

In the representative taxonomy-involved runs of `Qwen3-4B DSPy CoT LIGHT W1` and `Qwen3-4B DSPy CoT FULL W1`, this improvement is proven to be significant when compared to their counterparts using `Qwen3-4B` and applying the DSPy `CoT` module in the off-the-shelf testing.

Additionally, the controlled runs, which employed in-context learning without incorporating any element of our proposed taxonomy, didn't yield an accuracy improvement (see Table 4.10). This makes it more convincing that the accuracy gains of taxonomy-involved runs observed in Table 4.9, Figure 4.1 and Table 4.11 can be attributed primarily to the taxonomy integration. With regard to the best candidate for an LLM-based classifier to identify underspecified queries in QA, `Qwen3-4B DSPy CoT LIGHT W1` is deemed to be the best option due to its slight advantage in the UND F1 score.

A more detailed examination shows that the specific approach of integrating examples from the **ICLS** dataset with taxonomy-based annotations (applied in both `FULL` and `LIGHT` settings) seems to be more effective than feeding the model only with detailed definitions of taxonomic categories (the `DEF` setting). For the settings where only the category names are provided with simple instructions (the `MINI` setting), performance was negatively influenced, revealing that this minimal input of taxonomic information provided a noisy signal rather than informative guidance for the model. In addition, it should be noted that our current approach to integrating theory-driven taxonomic information into the NL prompt format didn't yield performance improvements. Considering the long context length and processing time observed in the NL prompting implementation (e.g., `Qwen3-4B NL FULL` had an average prompt length of 18,709 tokens and a ~12-hour processing time for **UND-QA-MS**), this is possibly due to the sub-optimal prompting that requires further iterations.

A qualitative analysis on `Qwen3-4B DSPy CoT LIGHT W1`. As mentioned, the `Qwen3-4B DSPy CoT LIGHT W1` setting can be regarded as the best candidate for an LLM-based classifier to identify underspecified queries in QA that leverages theoretical insights from linguistics. To further validate this, we qualitatively compare its classification results with those from the best-performing run `Qwen3-4B NL` obtained in the off-the-shelf LLM testing. Our analysis focuses on two types of cases: those in which `Qwen3-4B DSPy CoT LIGHT W1` "corrected" the off-the-shelf prediction to align with the gold FS/UND judgment, and those in which it deviated from the gold judgment in contrast to the off-the-shelf prediction.

We first present some cases where `Qwen3-4B DSPy CoT LIGHT W1` "corrected" the off-the-shelf predictions to align with the gold annotations in the source datasets in Table 4.12, most of which were also showcased in Table 4.4. It can be observed that `Qwen3-4B DSPy CoT LIGHT W1` is capable of relieving issues we discussed in the analysis in the section 4.1.3. Examples 1 and 4 show that **its**

#	Query	`Qwen3-4B NL`	`Qwen3-4B DSPy CoT LIGHT W1`
1	Tell me which continent the following cities are in: - Seattle, - Helsinki, - Lagos, - Osaka, - Istanbul	UND – The query requests identification of continents for five cities. Four of the cities (Seattle, Helsinki, Lagos, Osaka) are unambiguously located in distinct continents (North America, Europe, Africa, Asia). However, Istanbul is geographically situated at the boundary between Europe and Asia, creating ambiguity. (...)	FS – This question is fully specified because it requests a clear, factual geographic classification for each listed city, with no missing components. The referents (Seattle, Helsinki, Lagos, Osaka, Istanbul) are well-defined, (...)
2	states in the northeast region in us	UND – The query seeks states in the 'northeast region' of the U.S., but fails to explicitly define what constitutes the 'northeast region.' While commonly understood as comprising New England and Mid-Atlantic states (e.g., Maine, New Hampshire, Massachusetts, etc.), the term 'region' lacks formal geographic boundaries. (...)	FS – This question is fully specified because it asks for a clear, factual geographic classification of U.S. states into the Northeast region, which is a well-defined and widely accepted regional division. (...)
3	When do we celebrate veterans day this year?	FS – (...) Veterans Day is universally recognized as falling on November 11th every year in the United States, making the required information directly retrievable from general knowledge. (...)	UND – The request is underspecified because the phrase "we" is ambiguous — it could refer to multiple countries or regions where Veterans Day is observed on different dates (e.g., the U.S. observes it on November 11, while other countries have different traditions).
4	Does Chelyrttodestoni have semiprecocial as its developmental mode?	FS – (...) While the scientific validity of the species name may be questionable (it appears to be a misspelled or fictional taxon), the query itself contains no ambiguities regarding what is being asked. (...)	UND – The request is underspecified because the term "Chelyrttodestoni" is not a recognized or valid scientific name for any known species, making the referent ambiguous or non-existent. (...)
5	Find a recipe for gluten-free brownies.	FS – The query specifies a clear objective ('recipe') and a distinct constraint ('gluten-free'), which narrows down the search space significantly. While typical recipes may include additional parameters like ingredient quantities, preparation time, or serving sizes,these are not explicitly required for the query to be actionable. (...)	UND – The request is underspecified because it does not clarify the user's specific dietary preferences beyond "gluten-free" (e.g., vegan, dairy-free, or specific alternative flours like almond or coconut). Additionally, the desired texture (fudgy, cakey) or recipe complexity (simple vs. elaborate) is not specified, (...)

Table 4.12: Example queries where `Qwen3-4B DSPy CoT LIGHT W1` "corrected" the predictions from off-the-shelf test to align with the gold annotations in the source datasets

conception of underspecified queries is more aligned with human annotators, correctly claiming that the "transcontinental" nature of Istanbul is not related to the clarity of Example 1 as a query, and the undefined noun phrase "Chelyrttodestoni" would cause underspecification for Example 4. Example 3, on the other hand, shows that **`Qwen3-4B DSPy CoT LIGHT W1` is improved in its ability to detect the key factor determining a query's UND/FS judgment**, rightfully requiring a further clarification for the referent "we". As for Examples 2 and 5, **`Qwen3-4B DSPy CoT LIGHT W1` was able to converge with human annotators on the threshold of UND/FS judgment**, regarding a commonly-recognized geographical concept as sufficiently specified and a recipe request with personal preference missing as underspecified.

Then we turn to cases where `Qwen3-4B DSPy CoT LIGHT W1` altered the off-the-shelf predictions and caused misalignments with the gold annotations in the source datasets, with examples shown in Table 4.13. Examples 1 and 2 demonstrate that **the model's drawbacks in other aspects may also lead to issues for a reasonable UND/FS classification**. In Example 1, the model's uncertainty/lack of inner knowledge about the bird species native to Antarctica was the main reason leading to an unaligned judgment. There are in fact more than one bird species native to Antarctica;[8] thus, reasoning texts from both `Qwen3-4B NL` and `Qwen3-4B DSPy CoT LIGHT W1` are erroneous, but the error made by the latter caused a misaligned judgment specifically in this case. While in Example 2, `Qwen3-4B DSPy CoT LIGHT W1` failed to recognize the "US" written in a non-standard orthography and raised the lack of locative information as a result. Examples 3 and 4, on the other hand, show that **`Qwen3-4B DSPy CoT LIGHT W1` can still fail to attach importance to the key factor that determines the UND/FS judgment**. Even though it can actually generate relevant reasoning (texts in green), questionable assumptions were made to marginalize them as secondary factors, which is also related to **the issue of the human-LLM misalignment on the classification threshold**. Last but not least, Example 5 is a typical case where the inherent uncertainty regarding **the threshold of how much subjective input is necessary to fulfill a query** causes difficulty for the model. For such cases, annotations from human annotators may also vary interpersonally and show inconsistency.

To summarize, the taxonomy-integrated `Qwen3-4B DSPy CoT LIGHT W1` has shown quantitative improvement on the accuracy of UND/FS classification and the identification of UND queries when compared to its counterparts in the off-the-shelf test without any theoretical tuning. Qualitative analysis also shows that to some extent, it successfully addressed several common issues found in the previous off-the-shelf LLM testing. However, other complex or inherent problems, such as the model's general knowledge, false assumptions, and highly subjective classification thresholds observed in some queries, still remain. Nevertheless, `Qwen3-4B DSPy CoT LIGHT W1` can be regarded as a satisfactory LLM-based classifier to identify underspecified queries QA interactions, which can be applied to downstream tasks.

[8] https://en.wikipedia.org/wiki/List_of_birds_of_Antarctica

#	Query	`Qwen3-4B NL`	`Qwen3-4B DSPy CoT LIGHT W1`
1	Tell me a joke involving a bird native to Antarctica.	FS – The query specifies 'a bird native to Antarctica,' which unambiguously refers to penguins (the only native avian species in Antarctica). While jokes may require creative wordplay, the core requirement is met by identifying penguins as the target subject. (...)	UND – The request is underspecified because the term "bird native to Antarctica" is ambiguous. There are no native bird species to Antarctica, as the continent is covered in ice and has no native bird species.
2	who was the first we president that was assassinated in us	FS – The query appears to contain typographical errors ('we' likely meaning 'U.S.' and 'in us' likely meaning 'in the United States'). Despite these errors, the core intent is clear: the user seeks the identity of the first U.S. president to be assassinated.	UND – The request is underspecified because the pronoun "we" is ambiguous and can refer to multiple parallel political entities, such as the United States, the United Kingdom, or other nations. (...)
3	the organization that sets accounting standards for private sector organizations is the	UND – (...) While the U.S. Financial Accounting Standards Board (FASB) is widely recognized as the primary entity for U.S.-based private-sector accounting standards, the answer depends on geographic or jurisdictional context. Since the query does not explicitly specify a country, region, or regulatory framework (e.g., IFRS via IASB), the response is ambiguous. (...)	FS – (...) The referent is unambiguous, and while there are multiple global standards bodies (e.g., FASB in the U.S., IASB internationally), the question implicitly refers to the U.S. context, which is the most commonly expected answer in this phrasing. (...)
4	How many times have the warriors won the finals?	UND – The query lacks specificity due to the ambiguous reference to 'the warriors.' While 'Warriors' is commonly associated with the Golden State Warriors (NBA), the term could refer to any team named 'Warriors' across different sports or leagues (e.g., Atlanta Braves, San Francisco Giants, or even non-sports organizations) (...)	FS – This question is fully specified because it refers to the Golden State Warriors, a well-known NBA team (...) While the term "finals" could theoretically refer to other sports, the context of "Warriors" and the most common association with the NBA makes the referent precise.
5	I'm hosting a 90's themed party. Create a playlist of hit songs from the 90's.	UND – (...) However, it lacks details about the specific genres or artists preferred, the length of the playlist, and whether the user wants widely recognized hits or more niche selections that were popular in the 90's. (...)	FS – This question is fully specified because it clearly requests a curated list of popular music from the 1990s, which is a well-defined era with a documented set of hit songs. (...) The task is to compile a standard list of widely recognized 90s hits, which is a factual and well-established request.

Table 4.13: Example queries where `Qwen3-4B DSPy CoT LIGHT W1` changed the predictions from off-the-shelf test and caused misalignments with the gold annotations in the source datasets.

Chapter 5

The Application of the LLM-based Classifier to General Question Answering Datasets

As discussed in previous chapters, semantic underspecification is an inherent feature of human communication (Harris, 2020a), and human-LLM interactions are also influenced by it: Herlihy et al. (2024) classified a random sub-sample from the OpenAssist (Köpf et al., 2023), a dataset with human-curated queries posed to LLM-based chatbots, and found that 23% of them are "severely underspecified". In light of this, it is reasonable to predict that underspecified queries are widespread in datasets devoted to QA scenarios of human-machine interaction. While prior work has mostly explored this phenomenon using specialized benchmarks, less attention has been paid to quantitatively evaluating the presence and impact of underspecified queries in large-scale general QA datasets, which are commonly used for LLM tuning and evaluation. Existing efforts primarily focused on composing benchmarks explicitly for this purpose, using approaches such as sampling from various other datasets (Zhang, Qin, et al., 2024; Brahman et al., 2024; Min et al., 2020) and generating synthetic examples with human supervision (Qian et al., 2024). Comparatively, the curation of AmbigNQ (Min et al., 2020) is more directly related to general QA data, as it is sampled from the general-purpose NQ-Open (Lee, Chang, and Toutanova, 2019) dataset. Based on their annotations, half of the sampled queries contain multiple question-answer pairs, "indicating a high rate of ambiguity in NQ-Open" (Min et al., 2020, pp. 5786). Nonetheless, since the analytical granularity for queries to be underspecified in AmbigNQ is not aligned with other datasets (as discussed in Section 4.1.2), this reported proportion of underspecified queries may not be representative.

Therefore, this chapter aims to utilize the LLM-based classifier developed in the previous chapter for an evaluation of the extent to which underspecified queries are present in downstream QA datasets and how they affect the QA performance of commonly used SotA LLMs through another experiment. We apply our LLM-based classifier `Qwen3-4B DSPy CoT LIGHT W1` to identify the "underspecified" (UND) and the "fully specified" (FS) subsets of queries sampled from several standard QA datasets and report the statistical results. Following this, we prompt two SotA proprietary LLMs to answer queries from two subsets of UND and FS queries, respectively. We evaluate performance differences using established metrics, and finally conduct qualitative analysis on representative cases to provide deeper insights into underspecified queries "in the wild".

5.1 Large-Scale General QA Datasets

For the experiment in this chapter, we sampled 1,000 queries from each of three selected general QA datasets, resulting in a total of 3,000 queries. We now provide a brief introduction to the three selected source datasets.

Natural Questions (NQ) (Kwiatkowski et al., 2019). A large-scale QA dataset developed by Google, designed to facilitate the training and evaluation of open-domain QA systems. It comprises real user queries issued to the Google search engine, paired with answers annotated from full Wikipedia pages. Each instance includes the original question, the corresponding Wikipedia document, and annotations specifying long and short answers. Long answers are typically paragraphs that contain detailed information to resolve corresponding queries, while short answers are (text) spans, lists of spans, or booleans ("Yes"/"No") that can serve as succinct solutions. It's possible for long answers and short answers to be annotated as `NULL` if no answer is available on the corresponding Wikipedia page. The dataset includes 307,373 training examples, 7,830 development examples, and 7,842 test examples. We obtained a 1,000-query sample from its development set with the constraint that both the short and long answers should be non-empty.

HotpotQA (Yang, Qi, et al., 2018). A large-scale QA dataset designed to evaluate the capability of QA systems in multi-hop reasoning over diverse textual information. It comprises 112,779 question-answer pairs derived from Wikipedia articles that require reasoning across multiple documents. Crowdsource annotators developed the questions based on multiple supporting context documents and ensured that they require reasoning about all these documents, while the answers are concise text spans or a "Yes" / "No" response extracted from the context documents. We acquired a 1,000-query sample from its development set.

TriviaQA (Joshi et al., 2017). A large-scale QA dataset designed for reading comprehension (RC) systems with 95,000 question-answer pairs authored by trivia enthusiasts. The creation of these questions is independent of evidence documents, which reduces bias in question style or content and provides more organic and topic-diverse tasks. Regarding answers, they are provided in lists of aliases to enable a more robust evaluation with better compatibility with lexical or syntactic variations. We retrieved a 1,000-query sample from the validation split of its "rc" (reading comprehension) subset.

We present example queries and their annotated answers from sampled data in Table 5.1.

5.2 SotA Proprietary LLMs for QA

To examine the impact of underspecified queries on QA performance, we evaluate two leading proprietary LLMs: `GPT-4o` and `Gemini 2.5 Flash`. We are motivated to select them as they are equipped with advanced instruction-following capabilities and have shown strong performance in real-world applications. They are ideal candidates for assessing whether top-tier SotA commercial LLMs are robust to underspecified queries in QA tasks.

Query	Annotated Answer(s)	Source
when did the study of media effects begin	Short Answer: '1919' Long Answer: 'The social impact of mass communication has been studied at The New School University in New York since its founding in 1919 . The first college course to investigate the motion picture was offered here in 1926 . Marshall McLuhan 's colleague , John Culkin , brought his Center for Understanding Media to The New School in 1975 and The New School began offering the Master of Arts degree in Media Studies , one of the first graduate programs of its kind . Today , among other programs , MA in Media Studies is still being offered by School of Media Studies , The New School , which will celebrate 40th anniversary of Media Studies at The New School during the academic year 2015 - 2016'	NQ
What nationality was Oliver Reed's character in the film Royal Flash?	'Prussian'	HotpotQA
What general name is given to a rotating star which emits a regular beat of radiation?	['yukon optics', 'beltex optics', 'pulsar']	TriviaQA

Table 5.1: Example queries and their annotated answers from sampled data of each of the selected general QA datasets.

GPT-4o (OpenAI, Hurst, et al., 2024). A state-of-the-art autoregressive multimodal model that excels in QA tasks. In the grade-school science QA dataset ARC-Easy (Clark et al., 2018), it achieves a score of 94.8% in English and maintains a score above 70% in underrepresented languages (e.g., Hausa). On the misinformation-stress TruthfulQA benchmark (Lin, Hilton, and Evans, 2022), it delivers 81.4% accuracy. Additionally, the model's QA performance based on speech input remains competent at a level marginally below that of text-only runs.

Gemini 2.5 Flash (Comanici et al., 2025). A hybrid-reasoning model developed by Google DeepMind, which is engineered for low-latency deployment while retaining SotA competence on QA tasks. It is reported to have attained 82.8% accuracy on the graduate-level GPQA-diamond benchmark (Rein et al., 2023) and reaches 26.9% F1 on the factual SimpleQA benchmark (Wei, Karina, et al., 2024), demonstrating robust question-answering competence across both reasoning and factual recall tasks.

5.3 Experimental Setup

For each 1,000-query sample from the respective datasets, we first identify their subsets of "fully specified" (FS) and "underspecified" (UND) queries, using the method developed in Section 4.2, and in particular the LLM-based classifier based on `Qwen3-4B DSPy CoT LIGHT W1`.

Following the subset identification, we prompt `gpt-4o-2024-11-20` and `gemini-2.5-flash` through their official APIs to execute QA tasks and collect generated answers for queries in FS and UND subsets of each sample, respectively. More specifically, we create two prompt settings, one for generating short answers and another for generating long answers. The former is used for NQ short-answer generation

Label/Dataset	NQ	HotpotQA	TriviaQA
UND	300	394	98
FS	700	606	902
Total	1000	1000	1000

Table 5.2: The FS/UND classification results of the samples from three general QA datasets.

and also serves as the default for queries from HotpotQA and TriviaQA. In contrast, the latter is specifically used to generate long answers for NQ queries.[1]

Lastly, we evaluate answers generated by the two LLMs to queries in each FS/UND subset of the drawn samples. For short answers to NQ queries and answers to HotpotQA and TriviaQA, we compute the Exact Match (EM) and F1 scores between the model-generated answers and the annotated gold answers from the original datasets on a per-query basis. In cases where either the model generations or the gold answers contain multiple answers/aliases, we evaluate all answer pairs and report the maximum score obtained for each metric. To assess the models' performance across FS and UND queries, we compute the average EM and F1 scores separately for the FS and UND subsets within each sample and perform independent *t*-tests to determine whether the models' performance differences between FS and UND queries are statistically significant for each metric. On the other hand, for long answers to NQ queries, we evaluate model outputs using the `evaluate` library developed by Hugging Face (Von Werra et al., 2022) to compute BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), ROUGE (Lin, 2004), and BERTScore F1 (Zhang, Kishore, et al., 2020) on a per-query basis. These metrics are suitable for long-answer evaluation, as they capture different aspects of surface-level n-gram overlap (BLEU and ROUGE), syntactic similarity (METEOR), and semantic similarity (BERTScore). Similar to short-answer evaluation, we calculate the average scores for each metric across the FS and UND subsets within each sample, and apply independent *t*-tests to determine whether the performance differences between the FS and UND queries are statistically significant.

5.4 Results

We first present the FS/UND classification results of samples from the three standard QA datasets yielded by our LLM-based classifier `Qwen3-4B DSPy CoT LIGHT W1` in Table 5.2. It can be observed that the proportion of underspecified queries varies between datasets. In HotpotQA, this proportion is the highest, reaching a percentage of 39.4%. The NQ dataset ranks second with a proportion of 30.0%. In contrast, TriviaQA has the lowest proportion of 9.8%. This indicates that all selected general QA datasets contain cases categorized by our LLM-based classifier as underspecified queries, with the proportion of such queries varying across different specific datasets.

Next, we turn to the evaluations of answers generated by `GPT-4o` and `Gemini 2.5 Flash` to present their respective performance on FS queries and UND queries in samples from the three general QA datasets. Table 5.3 reports the average per-query EM and F1 scores of the two models' answers, evaluated against gold answers from NQ (Short Answers), HotpotQA, and TriviaQA. It can be observed that there is a uniform pattern such that performance on FS subsets (marked with underlining) is always better than on UND subsets across models and datasets. From the perspective of individual datasets, we can see that `GPT-4o` slightly outperforms `Gemini 2.5 Flash` on the FS query subset of

[1] Please refer to Appendix A for detailed prompt settings discussed here.

	NQ (Short Answers)			
	`GPT-4o`		`Gemini 2.5 Flash`	
	FS	UND	FS	UND
EM Mean	**29.9**	16.0	28.7	20.0
F1 Mean	**48.2**	31.4	47.3	36.0
	HotpotQA			
	`GPT-4o`		`Gemini 2.5 Flash`	
	FS	UND	FS	UND
EM Mean	38.94	22.08	**45.38**	27.92
F1 Mean	52.49	35.72	**60.15**	42.04
	TriviaQA			
	`GPT-4o`		`Gemini 2.5 Flash`	
	FS	UND	FS	UND
EM Mean	79.38	68.37	**80.49**	74.49
F1 Mean	86.95	78.39	**88.18**	82.19

Table 5.3: The average per-query EM and F1 scores of answers generated by `GPT-4o` and `Gemini 2.5 Flash`, evaluated against gold answers from NQ (short answers), HotpotQA, and TriviaQA.

	NQ (Long Answers)			
	`GPT-4o`		`Gemini 2.5 Flash`	
	FS	UND	FS	UND
BLEU mean ± SD	**2.2** ± 2.6	1.8 ± 2.2	1.4 ± 1.5	1.2 ± 1.5
METEOR mean ± SD	**24.8** ± 9.9	22.1 ± 10.0	21.5 ± 8.7	19.8 ± 8.7
ROUGE mean ± SD	**13.3** ± 5.9	11.8 ± 5.7	9.8 ± 4.5	9.1 ± 4.5
BERTScore F1 mean ± SD	**81.3** ± 3.1	80.6 ± 3.3	80.3 ± 2.0	79.9 ± 2.1

Table 5.4: The average per-query BLEU, METEOR, ROUGE, and BERTScore F1 scores of model-generated long answers to sample queries from NQ, evaluated against the corresponding gold answers.

NQ (Short Answers). In contrast, for the UND query subset of this dataset, `Gemini 2.5 Flash` shows a better performance. As for HotpotQA and TriviaQA, `Gemini 2.5 Flash` exhibits an advantage across subsets of FS and UND queries.

Table 5.4 presents the average per-query BLEU, METEOR, ROUGE, and BERTScore F1 scores of model-generated long answers to sample queries from the NQ dataset, evaluated against the corresponding gold answers. Similarly, there is a uniform pattern in which performance on FS subsets (marked with underlining) is consistently better than on UND subsets across models. From the perspective of overall model performance, we observe that `GPT-4o` slightly outperforms `Gemini 2.5 Flash` across all metrics and both subsets.

Finally, we present the results of independent t-tests comparing per-query evaluation metrics between the FS and UND subsets across samples from the three general QA datasets. Table 5.5 shows that, for short-answer generation on NQ, HotpotQA, and TriviaQA, both `GPT-4o` and `Gemini 2.5 Flash` exhibit significantly higher EM and F1 scores on respective FS subsets than on the UND subsets in most cases. Specifically, all comparisons on NQ (Short Answers) and HotpotQA are highly significant for both models ($p < .0001$), with consistently positive t-values indicating a performance

	NQ (Short Answers)		HotpotQA		TriviaQA	
	GPT-4o	Gemini 2.5 Flash	GPT-4o	Gemini 2.5 Flash	GPT-4o	Gemini 2.5 Flash
FS-UND EM *t*-test	$t = 5.06, p < .0001$	$t = 3.03, p < .0001$	$t = 5.85, p < .0001$	$t = 5.75, p < .0001$	$t = 2.24, p = .027$	$t = 1.30, p = .2$
FS-UND F1 *t*-test	$t = 6.48, p < .0001$	$t = 4.14, p < .0001$	$t = 6.17, p < .0001$	$t = 6.46, p < .0001$	$t = 2.25, p = .027$	$t = 1.66, p = .1$

Table 5.5: Independent *t*-test results comparing per-query EM and F1 scores between FS and UND subsets on NQ (Short Answers), HotpotQA, and TriviaQA.

	NQ (Long Answers)	
	GPT-4o	Gemini 2.5 Flash
FS-UND BLEU *t*-test	$t = 3.06, p = .002$	$t = 1.8, p = .072$
FS-UND METEOR *t*-test	$t = 3.98, p < .001$	$t = 2.79, p = .005$
FS-UND ROUGE *t*-test	$t = 3.68, p < .001$	$t = 2.16, p = .031$
FS-UND BERTScore F1 *t*-test	$t = 3.14, p = .002$	$t = 2.78, p = .006$

Table 5.6: Independent *t*-test results comparing per-query BLEU, METEOR, ROUGE and BERTScore F1 scores between FS and UND subsets on NQ (Long Answers).

advantage on FS queries. On TriviaQA, `GPT-4o` still shows statistically significant differences between the FS and UND subsets ($p = .027$ for both EM and F1, FS advantage), while `Gemini 2.5 Flash` exhibits a similar trend with statistical non-significance ($p = .2$ for EM and $p = .1$ for F1).

For long-answer generation on NQ, the results presented in Table 5.6 follow a similar pattern. `GPT-4o` achieves significantly higher BLEU, METEOR, ROUGE, and BERTScore F1 scores on the FS subset, with all p-values below .005. `Gemini 2.5 Flash` also shows a significant advantage on FS queries for METEOR ($p = .005$), ROUGE ($p = .031$), and BERTScore F1 ($p = .006$), while this advantage is relatively minor and not significant on BLEU ($p = .072$).

5.5 Analysis

Results of the FS/UND classification implemented by `Qwen3-4B DSPy CoT LIGHT W1` (see Table 5.2) as an LLM-based classifier show that **underspecified queries are identified in all samples from selected large-scale general QA datasets, indicating their widespread presence. The proportion of underspecified queries varies across these samples, ranging from roughly 10% to 40%.** This proportional discrepancy can be attributed to the heterogeneous characteristics of the queries they collected. Data points from NQ are real user queries issued to the Google search engine, reflecting the fundamental existence of semantic underspecification in human linguistic behaviour. HotpotQA queries require multi-hop reasoning across multiple information sources, and crowdsource annotators may omit bridging entities or implicit assumptions that connect the sources, potentially contributing to a larger number of underspecified expressions. In contrast, TriviaQA primarily contains carefully curated "trivia-competition" queries that tend to be highly specific and self-contained, typically yielding a unique interpretation and a single answer. Comparatively, NQ and HotpotQA may be more representative of general open-domain QA information-seeking behavior. However, despite this proportional discrepancy originating from dataset features, **the universal observation is that underspecified queries are observed across large-scale QA datasets**, even in a dataset supposedly to be “highly specified”.

As already mentioned in Section 5.4, Table 5.3 and Table 5.4 outline a uniform pattern that both `GPT-4o` and `Gemini 2.5 Flash` perform better on FS queries than on UND queries across all metrics and samples from different datasets. Results of independent *t*-tests across various metrics shown in Table 5.5 and Table 5.6 further verify this consistent performance gap and confirm that the models' performance advantage on FS queries is statistically significant in the majority of comparisons. It's also worth mentioning that, in the exceptional case where the FS-UND performance gap on the TriviaQA sample is not significant for `Gemini 2.5 Flash`, the small sample size (only 98 instances) of UND queries may have underpowered the test. The non-significant result should not be directly interpreted as a genuine performance parity between the two subsets. **The combination of these insights highlights that underspecified queries indeed have a negative influence on the evaluation of LLMs' QA performance.**

In addition, we conduct a qualitative analysis of several examples and their answers (both from the models and the gold annotations, see Table 5.7) to examine the effect of underspecified queries on the QA performance of LLMs from a micro perspective. Due to space constraints, we focus on short answers in the presentation. Examples 1 and 2 demonstrate that when queries involve concepts lacking a unique or clear-cut definition, LLMs may generate answers based on an alternative but plausible interpretation. These answers can diverge from the human annotation, which typically reflects only a single perspective without accounting for the potential underspecification. Similarly, Examples 5, 7, and 9 reveal that when a nominal component can be mapped to multiple referents, LLMs are prone to generating answers based on a referent that is completely plausible but misaligned with the one provided by the human annotator, who did not consider other possibilities loaded in the underspecification. Examples 3, 4, and 8, on the other hand, highlight scenarios where underspecified temporal and locative anchors result in misalignments between the model generation and the human annotation. It's especially worth noting that time-sensitive, underspecified queries make up a substantial share of misaligned cases we observed, which are primarily due to diachronic information update. This may have emphasized the insight that the QA performance of LLMs on time-sensitive queries should not be evaluated using static annotated data. Last but not least, Example 6 highlights that the vagueness in queries can result in model answers that diverge significantly from the annotator's interpretation, which is also based on one specific perspective without considering the underspecification. Overall, all the examples we presented here reflect that **the misalignment between model-generated answers and gold answers from human annotators is not necessarily caused by the models' incapability in QA tasks**. Instead, **it is often due to the fact that during the construction of large-scale general QA datasets, the annotations didn't take the multiple possible interpretations and answers into consideration.** This raises fundamental concerns about the reliability and the utility of these large-scale QA datasets: If performance scores do not really reflect a model's actual capability in QA tasks, can we still use them for reasonable LLM evaluations? Furthermore, provided that these datasets that failed to consider underspecification during construction are used for in-domain training and fine-tuning of LLMs, is the capability of LLMs to handle underspecified user inputs being inadvertently compromised or misrepresented?

It should also be noted that in some examples, our LLM-based classifier `Qwen3-4B DSPy CoT LIGHT W1` mistakenly classified queries as UND because it could not retrieve the referent of certain expressions included, which can potentially be attributed to the relatively small size of its base model `Qwen3-4B`.

#	Query	Gold (Short)	Source	Classifier Reasoning	Model Answer (Short)
1	where does the modern view of history originate from	[’approximately in the early 16th century’]	NQ	… the phrase “modern view of history” is vague and can refer to multiple interpretations or schools of thought…	Gemini 2.5 Flash: [’* The Enlightenment* 19th-century German historicism’]
2	when did the age of new media began	[’the 1980s’]	NQ	… the term “age of new media” is not a universally accepted or standardized term, and its definition can vary depending on the perspective or context…	GPT-4o: [’The age of new media began in the late 20th century, around the 1990s, with the rise of the internet and digital technologies.’]
3	when did the smoking ban in public places start	[’1995’]	NQ	… the phrase "public places" is vague and depends on the specific jurisdiction or country being referenced. Smoking bans vary significantly across different regions…	GPT-4o: [’The smoking ban in public places started on March 26, 2006, in Scotland.’, ’It began on April 2, 2007, in Wales.’, ’It started on April 30, 2007, in Northern Ireland.’, ’It was implemented on July 1, 2007, in England.’]
4	who sang the american anthem at the super bowl	[’Pink’]	NQ	…the phrase "the Super Bowl" refers to an annual event, and the question lacks a specific year or context to determine which Super Bowl is being referenced…	GPT-4o: [’Chris Stapleton (Super Bowl LVII, 2023)’]
5	Which port city lies approximately 25 km north of the Lingnan Fine Arts Museum?	[’Keelung’]	HotpotQA	… the referent “Lingnan Fine Arts Museum” is not clearly defined, as there are multiple museums with similar names in different cities…	GPT-4o, Gemini 2.5 Flash: [’Guangzhou’]
6	Although a dependent territory remains politically outside of the controlling state’s integral area, what else can it contain?	[’the flags’]	HotpotQA	… the phrase “what else can it contain” is vague and lacks a clear standard or preference. It does not specify what is meant by “contain” (e.g., geographic features, political entities, or cultural elements) or what the context of the dependent territory is…	GPT-4o: [’A degree of autonomy or self-governance.’]
7	Who created the manga series originally serialized from 1994-1995 with a superheroine?	[’Tonde Burin’]	HotpotQA	… the phrase "superheroine" is vague and could refer to multiple parallel series with female protagonists, and the time frame (1994-1995) might overlap with multiple series…	GPT-4o, Gemini 2.5 Flash: [’Naoko Takeuchi’]
8	Motor Racing. Which team has announced that they are to withdraw from Formula One racing at the end of the year?	[’bmw z2’, ’bavarian motor works’, ..., ’bmw’]	TriviaQA	…it does not clarify the specific year or timeframe for the "end of the year," leaving the question open to interpretation…	GPT-4o, Gemini 2.5 Flash: [’Honda’]
9	Which month of the year came second in the Roman calendar?	[’april month’, ’april’...]	TriviaQA	…the term "Roman calendar" can refer to multiple historical versions (e.g., the original 10-month calendar or the Julian calendar), each with different month orderings…	GPT-4o: [’February’] Gemini 2.5 Flash: [’* April’]

Table 5.7: Example queries classified as “underspecified” (UND) and with their gold short answers and model-generated short answers.

#	Query	Gold (Short)	Source	Classifier Reasoning	Model Answer (Short)
1	who did the original spirit in the sky	['Norman Greenbaum']	NQ	... the phrase "original spirit in the sky" is ambiguous and lacks a clear referent. It could refer to a deity, a spiritual concept, or a metaphorical idea...	`GPT-4o`: ['Norman Greenbaum']
2	who wrote the text for jeanie with the light brown hair	['Stephen Foster']	NQ	... the phrase "Jeanie with the light brown hair" is not a recognized or widely known work, and the referent is ambiguous without additional context...	`GPT-4o`: ['Stephen Foster']

Table 5.8: Examples where `Qwen3-4B DSPy CoT LIGHT W1` misclassified queries as "underspecified" due to the failure of referent retrieval, but the QA models successfully retrieved the referent and produced aligned answers.

In such cases, LLMs used for QA are often observed to have no difficulty in mapping to the correct referent and generating aligned answers (see Table 5.8 for examples). However, since this type of query would only introduce a positive bias to our evaluation of the QA performance on UND subsets, it does not compromise the validity of the significant performance difference between FS and UND subsets that we've observed and discussed in the QA models.

To summarize, this chapter first suggests that underspecified queries are widely present in samples we drew from large-scale general QA datasets, and there is a variation in the proportion of underspecified (UND) queries across different datasets, potentially due to different sources/characteristics of the queries they collected. Furthermore, both of the SotA LLMs for QA we selected, `GPT-4o` and `Gemini 2.5 Flash`, show a consistent QA performance gap between FS queries and UND queries, as their performance advantage on FS queries over UND queries is significant in most comparisons across various datasets and metrics. A qualitative analysis further outlines the specific effects brought by various types of underspecified queries to the evaluation of models' QA performance. Misalignments between model-generated answers and human-annotated answers are often attributed to the lack of consideration regarding underspecification in the annotations of large-scale general QA datasets, which deviates from the original purpose of evaluating LLMs' QA capabilities. Despite some noise from the classifier, the overall patterns remain consistent, confirming that underspecified queries substantially impact LLM performance evaluation. It is therefore crucial for the research community to take underspecified queries in large-scale QA datasets seriously, as they pose significant challenges not only for evaluation but potentially for model development as well.

Chapter 6

General Discussion and Conclusion

6.1 General Discussion

This thesis investigates underspecified user queries in human-machine Question Answering (QA) scenarios, focusing on their definition and scope, methods to enhance their detection using Large Language Models (LLMs), their presence in large-scale QA datasets, and their impact of this presence, especially on the evaluation of LLM QA performance.

In Chapter 2, we laid the theoretical groundwork for this thesis. An in-depth overview of theoretical literature from linguistics and previous studies in NLP was conducted to explore a wide range of phenomena related to semantic underspecification, along with existing taxonomic analysis and the detection of underspecified queries in QA from different perspectives. We observed that **the conception of semantic underspecification can be connected with a wide scope of linguistic phenomena, which include not only the long-discussed ambiguity and vagueness, but also other nuanced phenomena of "linguistic undeterminacy".** Previous studies in NLP have proposed valuable taxonomic analyses based on their respective focus on certain specific types of these phenomena in underspecified QA queries, but no existing taxonomy has aimed for more comprehensive coverage. Additionally, both prompt-based methods and fine-tuning are prominent in the detection of underspecified queries of the LLM era, but the task remains a challenge for LLMs based on results from previous research.

Chapter 3 serves a bridging function from theoretical insights and inspirations from previous research to our experimental efforts for a better detection of underspecified queries in QA with LLMs. We started this chapter by specifying a general definition of underspecified queries in QA based on insights gained from the previous chapter. Then, we turned to review several existing benchmarks devoted to different types of underspecified queries in QA, and curated our own multi-source test dataset **UND-QA-MS**, with an aim to encode the diversity of underspecified queries and minimize the potential bias from individual benchmarks. We also selected five open-weight SotA LLM series for the later experiment. But most importantly, by combining theoretical insights from Chapter 2 and the analysis of empirical data distribution observed in a sub-sample from the reviewed existing benchmarks, we developed **a theory-informed and empirically-verified taxonomy of underspecified queries in QA, tailored for application to LLMs**. This taxonomy categorizes underspecified queries into four primary types that are rooted in several most relevant categories discussed in theoretical literature: [missing necessary components], [undetermined lexicons or references], [undetermined perspectives or granularity], and [undetermined standards or preferences].

Chapter 4 presents the main experiment of this thesis. We first tested the capabilities of the selected off-the-shelf SotA LLMs using the natural language (NL) prompt and DSPy prompts without any external input about the conception of underspecification. **The results indicated that open-weight SotA LLMs consistently face an accuracy ceiling at around 71%** with both NL prompting and DSPy prompting. Through a qualitative analysis, we also observed several common issues leading to the misalignment between the model classification and the gold judgments from human annotators. However, integrating our proposed taxonomy into prompt-based approaches can enhance the detection accuracy of an LLM. We curated an **in-context learning dataset ICLS using 150 examples** annotated with the gold "fully specified" (FS)/"underspecified" (UND) judgment, the category of our taxonomy to which it belongs and the reasoning behind the category assignment. Using this **ICLS** dataset and the textual definitions of our taxonomic categories, we experimented with a list of taxonomy-involved prompt-based settings on `Qwen3-4B`. We found that the best-performing run of `Qwen3-4B DSPy CoT LIGHT W1`, where we integrated the annotated examples from **ICLS** and simplified instructions of our taxonomic categories, achieved an accuracy of 73% with significant improvements compared to its counterparts in off-the-shelf testing. Through a detailed qualitative analysis, we also observed that this integration successfully addressed some common issues behind the misaligned classifications observed in the off-the-shelf testing. **This underscored the practical value of the supervision from our theory-informed taxonomy in helping LLMs to identify underspecified queries in QA, and `Qwen3-4B DSPy CoT LIGHT W1` can serve as a satisfactory LLM-based classifier for this purpose.**

In Chapter 5, we applied `Qwen3-4B DSPy CoT LIGHT W1` as the LLM-based classifier to identify subsets of underspecified (UND) queries and fully specified (FS) queries in three large-scale general QA datasets: Natural Questions (NQ), HotpotQA and TriviaQA. Results show that although the specific proportions of UND queries show variations (10% - 39.4%) due to the characteristics of their respective data points, **it is clear that underspecified queries are widespread in large-scale general QA datasets**. Moreover, we prompted two commonly used SotA proprietary LLMs `GPT-4o` and `Gemini 2.5 Flash` to perform QA tasks on subsets of UND and FS queries. **Results show that in the majority of cases, both models are evaluated to have significantly lower QA performance on underspecified queries compared to fully specified ones, with a consistent performance gap across evaluation metrics.** This highlights the fundamental issue in the utility of such general QA datasets for LLM evaluation, **as the "worse" performance metrics obtained from underspecified queries may be caused by gold annotations that did not consider other plausible answers, instead of genuinely reflecting the models' capabilities in QA tasks.** With these results, **we argued that the research community should consider underspecified queries seriously in the usage of existing large-scale QA datasets and the future development of new ones.**

6.2 Limitations and Future Work

While this thesis provides valuable insights, several limitations must be acknowledged and left for future work. First of all, despite the combination of insights from theoretical work in linguistics and the empirical data distribution in related QA datasets, the manually-constructed working taxonomy for underspecified queries in QA may still not encompass all possible forms of underspecified queries.

Secondly, along with the advantage of wider empirical coverage and the alleviation of potential bias, the multi-source test dataset **UND-QA-MS** used for our experiment may have also introduced variations in the annotation standards for the "fully specified"/"underspecified" judgment, as we directly utilized annotations from the source datasets. Thirdly, due to time and personnel constraints, the annotation processes leading to the proposed working taxonomy and the curation of the in-context learning (**ICLS**) dataset were limited in scale and cross-annotator verification. Lastly, the current prompt-based integration of our taxonomy, although effective, may still benefit from further optimization.

Future research could expand upon this thesis by employing larger-scale and more diverse datasets related to underspecified queries in QA to further validate and refine the proposed working taxonomy. More efforts could also be contributed to the multi-annotator verification and expansion of the **UND-QA-MS** and **ICLS** datasets to further improve the data quality for the model evaluation and in-context supervision. Additionally, exploring more advanced dynamic and adaptive prompting strategies to integrate our proposed working taxonomy has the potential to empower more accurate detection of underspecified queries using LLMs. But most importantly, based on our analysis of the fundamental issue brought by the underspecified queries in existing large-scale QA datasets, future work should pay more attention to this issue when using them. Researchers are encouraged to check for underspecification in datasets and apply filtering. At the same time, the construction of new large-scale QA datasets should include labels for underspecified queries or encourage more flexible answer annotations acknowledging the existence of underspecification, so that the QA capability evaluation of LLMs can become more robust and accurate.

6.3 Conclusion

In conclusion, this thesis provides a systematic study of underspecified queries in human-machine QA interactions, ranging from the theoretical foundation of their analysis, their detection using LLMs, to their presence in large-scale QA datasets and the practical implications of this presence. We developed a theory-informed and empirically verified working taxonomy of underspecified queries in QA. By the prompt-based integration of this taxonomy into a competent LLM, through both an in-context learning set **ICLS** annotated on the taxonomy's basis and the textual illustrations of taxonomic categories, we effectively improved the accuracy of underspecified query detection of the selected LLM, leading to a satisfactory LLM-based classifier. Furthermore, by utilizing this LLM-based classifier, we identified the substantial presence of underspecified queries across a number of large-scale general QA datasets. We recognized the negative impact of underspecified queries on LLM QA performance evaluation, which may not accurately reflect LLM capabilities but is rather a consequence of overlooking underspecification. This highlighted a fundamental issue with the reliability and utility of such datasets and underscored the importance of considering underspecified queries in QA for future large-scale QA dataset development and LLM research. Overall, this thesis establishes both theoretical and practical foundations for future research into improving human-machine QA interactions through effective management of semantically underspecified queries.

Appendix A

Prompts

A.1 Testing Off-the-Shelf LLMs

The natural language prompt (`NL`)

<system prompt> You are an expert analyst. Your task is to analyze and determine whether an input user query is “fully specified” or“underspecified”. </system prompt>

Analyze the following input user query:

{“query”: "When did the nuclear accident happen?"}

Please provide your analysis in the following JSON format:

{“query”: “When did the nuclear accident happen?”,
“reasoning”: ”[YOUR_DETAILED_REASONING]”,
“judgment”: “[fully specified/underspecified]”}

The DSPy prompts: `DSPy-Predict W1`

System message:

Your input fields are:
1. ‘request‘ (str): An input user question/request.

Your output fields are:
1. ‘judgment‘ (Literal[’underspecified’, ’fully specified’])

All interactions will be structured in the following way, with the appropriate values filled in.

[[## request ##]]
{request}

[[## judgment ##]]
{judgment} # note: the value you produce must exactly match (no extra characters) one of: underspecified; fully specified
[[## completed ##]]

In adhering to this structure, your objective is:
Given the fields ‘request‘, produce the fields ‘judgment‘.

User message:

[[## request ##]]
Who is the president?

Respond with the corresponding output fields, starting with the field ‘[[## judgment ##]]‘ (must be formatted as a valid Python Literal[’underspecified’, ’fully specified’]), and then ending with the marker for ‘[[## completed ##]]‘.

Response:

The DSPy prompts: `DSPy-CoT W1`

System message:

Your input fields are:
1. ‘request‘ (str): An input user question/request.

Your output fields are:
1. ‘reasoning‘ (str)
2. ‘judgment‘ (Literal[’underspecified’, ’fully specified’])

All interactions will be structured in the following way, with the appropriate values filled in.

[[## request ##]]
{request}

[[## reasoning ##]]
{reasoning}

[[## judgment ##]]
{judgment} # note: the value you produce must exactly match (no extra characters) one of: underspecified; fully specified

[[## completed ##]]

In adhering to this structure, your objective is:
Given the fields ‘request‘, produce the fields ‘judgment‘.

User message:

[[## request ##]]
Who is the president?

Respond with the corresponding output fields, starting with the field ‘[[## reasoning ##]]‘, then ‘[[## judgment ##]]‘ (must be formatted as a valid Python Literal[’underspecified’, ’fully specified’]), and then ending with the marker for ‘[[## completed ##]]‘.

Response:

The DSPy prompts: `DSPy-Predict W2`

System message:

Your input fields are:
1. ‘query‘ (str): An input user query.

Your output fields are:
1. ‘judgment‘ (Literal[’underspecified’, ’fully specified’])

All interactions will be structured in the following way, with the appropriate values filled in.

[[## query ##]]
{query}

[[## judgment ##]]
{judgment} # note: the value you produce must exactly match (no extra characters) one of: underspecified; fully specified
[[## completed ##]]

In adhering to this structure, your objective is:
Given the fields ‘query‘, produce the fields ‘judgment‘.

User message:

[[## query ##]]
Who is the president?

Respond with the corresponding output fields, starting with the field ‘[[## judgment ##]]‘ (must be formatted as a valid Python Literal['underspecified', 'fully specified']), and then ending with the marker for ‘[[## completed ##]]‘.

Response:

The DSPy prompts: `DSPy-CoT W2`

System message:

Your input fields are:
1. ‘query‘ (str): An input user query.

Your output fields are:
1. ‘reasoning‘ (str)
2. ‘judgment‘ (Literal['underspecified', 'fully specified'])

All interactions will be structured in the following way, with the appropriate values filled in.

[[## query ##]]
{query}

[[## reasoning ##]]
{reasoning}

[[## judgment ##]]
{judgment}   # note: the value you produce must exactly match (no extra characters) one of: underspecified; fully specified

[[## completed ##]]

In adhering to this structure, your objective is:
Given the fields ‘query‘, produce the fields ‘judgment‘.

User message:

[[## query ##]]
Who is the president?

Respond with the corresponding output fields, starting with the field ‘[[## reasoning ##]]‘, then ‘[[## judgment ##]]‘ (must be formatted as a valid Python Literal['underspecified', 'fully specified']), and then ending with the marker for ‘[[## completed ##]]‘.

The prompt for confidence scores

Given this statement from a reasoning process: "{reasoning}"

On a scale from 0 to 1, how confident does this statement support the classification of a user query as "fully specified"?

Return ONLY a number between 0 and 1. DO NOT PRODUCE ANYTHING ELSE.

A.2 Integrating the Proposed Working Taxonomy

The natural language prompts: `NL FULL`

<system prompt> You are an expert analyst. Your task is to analyze and determine whether an input user query is “fully specified” or“underspecified”. </system prompt>

In what follows, I’ll give you detailed definitions of "underspecified" and "fully specified" queries. For "underspecified" queries, they belong to 4 main categories:

- **Missing necessary components**: There is a strong perception that for at least one expression within the query, a commonly expected component conceptually tied to it is missing, thus its semantic interpretation is left with an unfilled slot. As a result, the meaning of the whole query is underspecified.

- **Undetermined lexicons or references**: The query contains at least one expression of lexical or referential ambiguity. For this lexical or referential ambiguous expression, multiple same-level entities or concepts can be mapped to it and serve as potential lexical entries or referents. It’s impossible to fully determine which one is intended based on the provided content. As a result, the meaning of the whole query is underspecified.

- **Undetermined perspective or granularity**: The query contains at least one expression where the general meaning is in place, but its specific interpretation can still vary based on different perspectives or granularities adopted. Multiple interpretations of different levels or natures are plausible for such an expression, and it’s impossible to fully determine which one is intended based on the provided content. As a result, the meaning of the whole query is underspecified.

- **Undetermined standard or preference**: The query contains at least one expression where the general meaning is in place, but its specific interpretation is vague due to unspecified contextual standards or subjective criteria. A wide range of fine-grained interpretations is possible according to contextual or subjective needs, and it’s impossible to fully determine which one is intended based on the provided content. As a result, the meaning of the whole query is underspecified.

There is only one same-name category of "fully specified" queries, defined as follows:

- **fully specified**: The given query is totally clear and definitive, and it doesn't belong to any provided category of "underspecified" queries.

Here are examples of how to analyze queries:

{"query": "When was the first world war broke out?", "reasoning": "The phrase "broke out" can be interpreted from multiple perspectives or granularities. It can be interpreted from the perspective of the political trigger event, which is the assassination of Archduke Franz Ferdinand, or from the perspective of the formal military actions and declarations of war between major powers. The question remains underspecified before a certain perspective is chosen.", "category": "Undetermined perspective or granularity", "judgment": "underspecified"}

. . . (the remaining 149 examples)

When analyzing a given query, think step by step explicitly:

First, provide "reasoning" regarding whether it belongs to any of the provided categories of "underspecified" queries or the category of "fully specified" queries.

Second, assign a "category" to it. It can belong to multiple categories of "underspecified" at the same time, but categories of "underspecified" and the category of "fully specified" are mutual-exclusive.

Lastly, based on the "category" assigned, determine a "judgment" on whether it is "underspecified" or "fully specified". Categories of "underspecified" can only lead to a "underspecified" judgment, while the category of "fully specified" can only lead to a "fully specified" judgment.

Now analyze the following query:

{"query": "When did the nuclear accident happen?"}

Think step by step as illustrated and provide your analysis in the following JSON format:
{"query": "When did the nuclear accident happen?", "reasoning": "[YOUR_DETAILED_REASONING]", "category": "[CATEGORY]", "judgment": "[fully specified/underspecified]"}

The natural language prompts: `NL DEF`

<system prompt> You are an expert analyst. Your task is to analyze and determine whether an input user query is "fully specified" or"underspecified". </system prompt>

In what follows, I'll give you detailed definitions of "underspecified" and "fully specified" queries.

For "underspecified" queries, they belong to 4 main categories:

- **Missing necessary components**: There is a strong perception that for at least one expression within the query, a commonly expected component conceptually tied to it is missing, thus its semantic interpretation is left with an unfilled slot. As a result, the meaning of the whole query is underspecified.

- **Undetermined lexicons or references**: The query contains at least one expression of lexical or referential ambiguity. For this lexical or referential ambiguous expression, multiple same-level entities or concepts can be mapped to it and serve as potential lexical entries or referents. It's impossible to fully determine which one is intended based on the provided content. As a result, the meaning of the whole query is underspecified.

- **Undetermined perspective or granularity**: The query contains at least one expression where the general meaning is in place, but its specific interpretation can still vary based on different perspectives or granularities adopted. Multiple interpretations of different levels or natures are plausible for such an expression, and it's impossible to fully determine which one is intended based on the provided content. As a result, the meaning of the whole query is underspecified.

- **Undetermined standard or preference**: The query contains at least one expression where the general meaning is in place, but its specific interpretation is vague due to unspecified contextual standards or subjective criteria. A wide range of fine-grained interpretations is possible according to contextual or subjective needs, and it's impossible to fully determine which one is intended based on the provided content. As a result, the meaning of the whole query is underspecified.

There is only one same-name category of "fully specified" queries, defined as follows:

- **fully specified**: The given query is totally clear and definitive, and it doesn't belong to any provided category of "underspecified" queries.

When analyzing a given query, think step by step in an explicit way:

First, provide "reasoning" regarding whether it belongs to any of the provided categories of "underspecified" queries or the category of "fully specified" queries.

Second, assign a "category" to it. It can belong to multiple categories of "underspecified" at the same time, but categories of "underspecified" and the category of "fully specified" are mutual-exclusive.

Lastly, based on the "category" assigned, determine a "judgment" on whether it is "underspecified" or "fully specified". Categories of "underspecified" can only lead to a "underspecified" judgment, while the category of "fully specified" can only lead to a "fully specified" judgment.

Now analyze the following query:

{"query": "When did the nuclear accident happen?"}

Think step by step as illustrated and provide your analysis in the following JSON format:
{"query": "When did the nuclear accident happen?", "reasoning": "[YOUR_DETAILED_REASONING]", "category": "[CATEGORY]", "judgment": "[fully specified/underspecified]"}

The natural language prompts: `NL LIGHT`

<system prompt> You are an expert analyst. Your task is to analyze and determine whether an input user query is “fully specified” or“underspecified”. </system prompt>

Here are examples of how to analyze user queries:

{"query": "When was the first world war broke out?", "reasoning": "The phrase “broke out” can be interpreted from multiple perspectives or granularities. It can be interpreted from the perspective of the political trigger event, which is the assassination of Archduke Franz Ferdinand, or from the perspective of the formal military actions and declarations of war between major powers. The question remains underspecified before a certain perspective is chosen.", "category": "Undetermined perspective or granularity", "judgment": "underspecified"}

. . . (the remaining 149 examples)

Now analyze the following input user query:

{"query": "Who started the white out in college football?", "reasoning": "[YOUR_DETAILED_REASONING]", "category": "[CATEGORY]", "judgment": "[fully specified/underspecified]"}

"category": select one or more labels (comma-separated) from ’Undetermined perspective or granularity’, ’Missing necessary components’, ’Undetermined lexicons or references’, ’Undetermined standard or preference’ if any undetermined factors exist. If the query is completely clear, output ’fully specified’.
"judgment": ’underspecified’ if any undetermined factors exist in the query, ’fully specified’ if the query is completely clear.

The natural language prompts: `NL MINI`

<system prompt> You are an expert analyst. Your task is to analyze and determine whether an input user query is “fully specified” or“underspecified”. </system prompt>

Analyze the following input user query:

{"query": "The idea that the united states was destined to extend westward across the continent?"}
Please provide your analysis in the following JSON format:
{"query": "The idea that the united states was destined to extend westward across the continent?", "reasoning": "[YOUR_DETAILED_REASONING]", "category": "[CATEGORY]", "judgment": "[fully specified/underspecified]"}
"category": select one or more labels (comma-separated) from ’Undetermined perspective or granularity’, ’Missing necessary components’, ’Undetermined lexicons or references’, ’Undetermined standard or preference’ if any undetermined factors exist. If the query is completely clear, output ’fully specified’.
"judgment": ’underspecified’ if any undetermined factors exist in the query, ’fully specified’ if the query is completely clear.

The DSPy prompts: `DSPy CoT FULL W1`

System message:

Your input fields are: 1. ‘request‘ (str): The input user question/request.

Your output fields are:
1. ‘reasoning‘ (str)
2. ‘category‘ (Literal[’Undetermined perspective or granularity’, ’Missing necessary components’, ’Undetermined lexicons or references’, ’Undetermined standard or preference’, ’fully specified’]): For "underspecified" questions/requests, they belong to 4 main categories:

- **Missing necessary components**: There is a strong perception that for at least one expression within the question/request, a commonly expected component conceptually tied to it is missing, thus its semantic interpretation is left with an unfilled slot. As a result, the meaning of the whole question/request is underspecified.

- **Undetermined lexicons or references**: The question/request contains at least one expression of lexical or referential ambiguity. For this lexical or referential ambiguous expression, multiple same-level entities or concepts can be mapped to it and serve as potential lexical entries or referents. It’s impossible to fully determine which one is intended based on the provided content. As a result, the meaning of the whole question/request is underspecified.

- **Undetermined perspective or granularity**: The question/request contains at least one expression where the general meaning is in place, but its specific interpretation can still vary based on different perspectives or granularities adopted. Multiple interpretations of different levels or natures are plausible for such an expression, and it’s impossible to fully determine which one is intended based on the provided content. As a result, the meaning of the whole question/request is underspecified.

- **Undetermined standard or preference**: The question/request contains at least one expression where the general meaning is in place, but its specific interpretation is vague due to unspecified contextual standards or subjective criteria. A wide range of fine-grained interpretations is possible according to contextual or subjective needs, and it’s impossible to fully determine which one is intended based on the provided content. As a result, the meaning of the whole question/request is underspecified.

There is only one same-name category of "fully specified" questions/requests, defined as follows:

- **fully specified**: The given question/request is totally clear and definitive, and it doesn’t belong to any provided category of "underspecified" questions/requests.

3. ‘judgment‘ (Literal[’underspecified’, ’fully specified’]): Categories of "underspecified" can only lead to a "underspecified" judgment, while the category of "fully specified" can only lead to a "fully specified" judgment.

All interactions will be structured in the following way, with the appropriate values filled in.

[[## request ##]] {request}
[[## reasoning ##]] {reasoning}

[[## category ##]] {category} # note: the value you produce must exactly match (no extra characters) one of: Undetermined perspective or granularity; Missing necessary components; Undetermined lexicons or references; Undetermined standard or preference; fully specified

[[## judgment ##]] judgment # note: the value you produce must exactly match (no extra characters) one of: underspecified; fully specified

[[## completed ##]]

In adhering to this structure, your objective is: You are an expert analyst. Your task is to analyze and determine whether an input user question/request is "fully specified" or "underspecified". When analyzing a given question/request, think step by step in an explicit way: First, provide "reasoning" regarding whether it belongs to any of the provided categories of "underspecified" questions/requests or the category of "fully specified" questions/requests. Second, assign a "category" to it. It can belong to multiple categories of "underspecified" at the same time, but categories of "underspecified" and the category of "fully specified" are mutual-exclusive. Lastly, based on the "category" assigned, determine a "judgment" on whether it is "underspecified" or "fully specified". Categories of "underspecified" can only lead to a "underspecified" judgment, while the category of "fully specified" can only lead to a "fully specified" judgment.

User message:

[[## request ##]] What's the latitude range where Oarcinidae is found?

Respond with the corresponding output fields, starting with the field '[[## reasoning ##]]', then '[[## category ##]]' (must be formatted as a valid Python Literal['Undetermined perspective or granularity', 'Missing necessary components', 'Undetermined lexicons or references', 'Undetermined standard or preference', 'fully specified']), then '[[## judgment ##]]' (must be formatted as a valid Python Literal['underspecified', 'fully specified']), and then ending with the marker for '[[## completed ##]]'.

Assistant message:

[[## reasoning ##]] The request is underspecified because no identifiable referent or scientific concept for "Oarcinidae" is available, preventing meaningful retrieval of geographic or ecological data.

[[## category ##]] Undetermined lexicons or references

[[## judgment ##]] underspecified

[[## completed ##]]

. . . (the remaining 149 examples)

User message:

[[## request ##]] Who is the president?

Respond with the corresponding output fields, starting with the field '[[## reasoning ##]]', then '[[## category ##]]' (must be formatted as a valid Python Literal['Undetermined perspective or granularity', 'Missing necessary components', 'Undetermined lexicons or references', 'Undetermined standard or preference', 'fully specified']), then '[[## judgment ##]]' (must be formatted as a valid Python Literal['underspecified', 'fully specified']), and then ending with the marker for '[[## completed ##]]'.

Response:

The DSPy prompts: `DSPy CoT DEF W1`

System message:

Your input fields are:
1. ‘request‘ (str): The input user question/request.

Your output fields are:
1. ‘reasoning‘ (str)
2. ‘category‘ (Literal[’Undetermined perspective or granularity’, ’Missing necessary components’, ’Undetermined lexicons or references’, ’Undetermined standard or preference’, ’fully specified’]): For "underspecified" questions/requests, they belong to 4 main categories:

- **Missing necessary components**: There is a strong perception that for at least one expression within the question/request, a commonly expected component conceptually tied to it is missing, thus its semantic interpretation is left with an unfilled slot. As a result, the meaning of the whole question/request is underspecified.

- **Undetermined lexicons or references**: The question/request contains at least one expression of lexical or referential ambiguity. For this lexical or referential ambiguous expression, multiple same-level entities or concepts can be mapped to it and serve as potential lexical entries or referents. It’s impossible to fully determine which one is intended based on the provided content. As a result, the meaning of the whole question/request is underspecified.

- **Undetermined perspective or granularity**: The question/request contains at least one expression where the general meaning is in place, but its specific interpretation can still vary based on different perspectives or granularities adopted. Multiple interpretations of different levels or natures are plausible for such an expression, and it’s impossible to fully determine which one is intended based on the provided content. As a result, the meaning of the whole question/request is underspecified.

- **Undetermined standard or preference**: The question/request contains at least one expression where the general meaning is in place, but its specific interpretation is vague due to unspecified contextual standards or subjective criteria. A wide range of fine-grained interpretations is possible according to contextual or subjective needs, and it’s impossible to fully determine which one is intended based on the provided content. As a result, the meaning of the whole question/request is underspecified.

There is only one same-name category of "fully specified" questions/requests, defined as follows:

- **fully specified**: The given question/request is totally clear and definitive, and it doesn’t belong to any provided category of "underspecified" questions/requests.

3. ‘judgment‘ (Literal[’underspecified’, ’fully specified’]): Categories of "underspecified" can only lead to a "underspecified" judgment, while the category of "fully specified" can only lead to a "fully specified" judgment.

All interactions will be structured in the following way, with the appropriate values filled in.

[[## request ##]] {request}

[[## reasoning ##]] {reasoning}

[[## category ##]] {category} # note: the value you produce must exactly match (no extra characters) one of: Undetermined perspective or granularity; Missing necessary components; Undetermined lexicons or references; Undetermined standard or preference; fully specified

[[## judgment ##]] {judgment} # note: the value you produce must exactly match (no extra characters) one of: underspecified; fully specified

[[## completed ##]]

In adhering to this structure, your objective is: You are an expert analyst. Your task is to analyze and determine whether an input user question/request is "fully specified" or "underspecified". When analyzing a given question/request, think step by step in an explicit way: First, provide "reasoning" regarding whether it belongs to any of the provided categories of "underspecified" questions/requests or the category of "fully specified" questions/requests. Second, assign a "category" to it. It can belong to multiple categories of "underspecified" at the same time, but categories of "underspecified" and the category of "fully specified" are mutual-exclusive. Lastly, based on the "category" assigned, determine a "judgment" on whether it is "underspecified" or "fully specified". Categories of "underspecified" can only lead to a "underspecified" judgment, while the category of "fully specified" can only lead to a "fully specified" judgment.

User message:

[[## request ##]] Who is the president?

Respond with the corresponding output fields, starting with the field ‘[[## reasoning ##]]‘, then ‘[[## category ##]]‘ (must be formatted as a valid Python Literal[’Undetermined perspective or granularity’, ’Missing necessary components’, ’Undetermined lexicons or references’, ’Undetermined standard or preference’, ’fully specified’]), then ‘[[## judgment ##]]‘ (must be formatted as a valid Python Literal[’underspecified’, ’fully specified’]), and then ending with the marker for ‘[[## completed ##]]‘.

Response:

The DSPy prompts: `DSPy CoT LIGHT W1`

System message:

Your input fields are: 1. ‘request‘ (str): An input user question/request.

Your output fields are: 1. ‘reasoning‘ (str)
2. ‘category‘ (Literal[’Undetermined perspective or granularity’, ’Missing necessary components’, ’Undetermined lexicons or references’, ’Undetermined standard or preference’, ’fully specified’]): Select one or more labels (comma-separated): Undetermined perspective or granularity, Missing necessary components, Undetermined lexicons or references, Undetermined standard or preference. If fully clear, output ’fully specified’
3. ‘judgment‘ (Literal[’underspecified’, ’fully specified’]): Final judgment: ’underspecified’ if any undetermined factors exist, ’fully specified’ if completely clear.

All interactions will be structured in the following way, with the appropriate values filled in.

[[## request ##]] {request}

[[## reasoning ##]] {reasoning}

[[## category ##]] {category} # note: the value you produce must exactly match (no extra characters) one of: Undetermined perspective or granularity; Missing necessary components; Undetermined lexicons or references; Undetermined standard or preference; fully specified

[[## judgment ##]] {judgment} # note: the value you produce must exactly match (no extra characters) one of: underspecified; fully specified

[[## completed ##]]

In adhering to this structure, your objective is: Given the fields ‘request‘, produce the fields ‘category‘, ‘judgment‘.

User message:

[[## request ##]] What’s the latitude range where Oarcinidae is found?

Respond with the corresponding output fields, starting with the field ‘[[## reasoning ##]]‘, then ‘[[## category ##]]‘ (must be formatted as a valid Python Literal[’Undetermined perspective or granularity’, ’Missing necessary components’, ’Undetermined lexicons or references’, ’Undetermined standard or preference’, ’fully specified’]), then ‘[[## judgment ##]]‘ (must be formatted as a valid Python Literal[’underspecified’, ’fully specified’]), and then ending with the marker for ‘[[## completed ##]]‘.

Assistant message:

[[## reasoning ##]] The request is underspecified because no identifiable referent or scientific concept for "Oarcinidae" is available, preventing meaningful retrieval of geographic or ecological data.
[[## category ##]] Undetermined lexicons or references
[[## judgment ##]] underspecified
[[## completed ##]]

. . . (the remaining 149 examples)

User message:

[[## request ##]] Who is the president?

Respond with the corresponding output fields, starting with the field '[[## reasoning ##]]', then '[[## category ##]]' (must be formatted as a valid Python Literal['Undetermined perspective or granularity', 'Missing necessary components', 'Undetermined lexicons or references', 'Undetermined standard or preference', 'fully specified']), then '[[## judgment ##]]' (must be formatted as a valid Python Literal['underspecified', 'fully specified']), and then ending with the marker for '[[## completed ##]]'.

Response:

The DSPy prompts: `DSPy CoT MINI W1`

System message:

Your input fields are: 1. 'request' (str): An input user question/request.

Your output fields are: 1. 'reasoning' (str)
2. 'category' (Literal['Undetermined perspective or granularity', 'Missing necessary components', 'Undetermined lexicons or references', 'Undetermined standard or preference', 'fully specified']): Select one or more labels (comma-separated): Undetermined perspective or granularity, Missing necessary components, Undetermined lexicons or references, Undetermined standard or preference. If fully clear, output 'fully specified'
3. 'judgment' (Literal['underspecified', 'fully specified']): Final judgment: 'underspecified' if any undetermined factors exist, 'fully specified' if completely clear.

All interactions will be structured in the following way, with the appropriate values filled in.

[[## request ##]] {request}

[[## reasoning ##]] {reasoning}

[[## category ##]] {category} # note: the value you produce must exactly match (no extra characters) one of: Undetermined perspective or granularity; Missing necessary components; Undetermined lexicons or references; Undetermined standard or preference; fully specified

[[## judgment ##]] judgment # note: the value you produce must exactly match (no extra characters) one of: underspecified; fully specified

[[## completed ##]]

In adhering to this structure, your objective is: Given the fields ‘request‘, produce the fields ‘category‘, ‘judgment‘.

User message:
[[## request ##]] Who is the president?

Respond with the corresponding output fields, starting with the field ‘[[## reasoning ##]]‘, then ‘[[## category ##]]‘ (must be formatted as a valid Python Literal['Undetermined perspective or granularity', 'Missing necessary components', 'Undetermined lexicons or references', 'Undetermined standard or preference', 'fully specified']), then ‘[[## judgment ##]]‘ (must be formatted as a valid Python Literal['underspecified', 'fully specified']), and then ending with the marker for ‘[[## completed ##]]‘.

Response:

Due to the page constraint, we leave out the specific example prompts of `DSPy CoT FULL W2`, `DSPy CoT DEF W2`, `DSPy CoT LIGHT W2`, and `DSPy CoT MINI W2`. They differ from their `W1` counterparts in the input field description: `‘query’ (str): An input user query`, and all the occurrences of the input name “request” are replaced by “query”. Everything else in the prompt content is identical.

A.3 Prompting SotA Proprietary LLMs for QA

QA System Prompt: Long Answer

Answer the question thoroughly and helpfully. Provide context, explanations, and relevant details from Wikipedia that would help the user understand the topic better.

QA System Prompt: Short Answer

Answer the question with a concise response. Return answers as a list of strings. If there's only one answer, return a single-item list. Each answer should be brief and direct.

Appendix B

Additional Performance Results

	Qwen3-4B NL	Qwen3-8B NL	Qwen3-14B NL	Qwen3-32B NL
FS F1	0.65	0.61	0.65	0.67
UND F1	0.41	0.44	0.37	0.34
accuracy	0.56	0.54	0.55	0.56
macro F1	0.53	0.53	0.51	0.51
	DeepSeek-R1-Distill-Qwen-1.5B NL	DeepSeek-R1-Distill-Qwen-7B NL	DeepSeek-R1-Distill-Qwen-14B NL	DeepSeek-R1-Distill-Qwen-32B NL
FS F1	0.2	0.53	0.69	0.61
UND F1	0.55	0.47	0.47	0.52
accuracy	0.43	0.5	**0.61**	0.57
macro F1	0.38	0.5	0.58	0.57
	DeepSeek-V3-0324-671B (API) NL	DeepSeek-R1-0528-671B (API) NL	Llama-3.2-3B NL	Llama-3.3-70B NL
FS F1	0.7	0.69	0.13	0.7
UND F1	0.31	0.49	0.62	0.24
accuracy	0.58	**0.61**	0.47	0.57
macro F1	0.5	0.59	0.37	0.47
	Qwen3-4B DSPy-Predict W1	Qwen3-4B DSPy-CoT W1	Llama-3.3-70B-Instruct DSPy-Predict W1	Llama-3.3-70B-Instruct DSPy-CoT W1
FS F1	0.69	0.68	0.69	0.71
UND F1	0.22	0.31	0.31	0.41
accuracy	0.56	0.56	0.57	**0.61**
macro F1	0.46	0.49	0.5	0.56
	Qwen3-4B DSPy-Predict W2	Qwen3-4B DSPy-CoT W2	Llama-3.3-70B-Instruct DSPy-Predict W2	Llama-3.3-70B-Instruct DSPy-CoT W2
FS F1	0.7	0.69	0.67	0.71
UND F1	0.37	0.32	0.38	0.39
accuracy	0.59	0.58	0.57	**0.61**
macro F1	0.54	0.51	0.53	0.55

Table B.1: An overview of the performance on the **AmbigNQ** subset of **UND-QA-MS** obtained from Off-the-shelf LLMs, using both natural language (NL) prompt and DSPy prompt.

Appendix C

Additional Statistical Tests

Model	CoCoNot F1 CI	IN3 F1 CI	CLAMBER F1 CI
Qwen3-4B NL	[0.7048, 0.8430]	[0.6823, 0.7792]	[0.5944, 0.6874]
Qwen3-32B NL	[0.7174, 0.8571]	[0.6688, 0.7666]	[0.5824, 0.6751]
Qwen3-4B DSPy-CoT W1	[0.7026, 0.8362]	[0.5771, 0.6914]	[0.5290, 0.6275]
Qwen3-4B DSPy-CoT W2	[0.7632, 0.8838]	[0.6475, 0.7502]	[0.5888, 0.6843]
DeepSeek-R1-Distill-Qwen-14B NL	[0.6638, 0.8178]	[0.5808, 0.6938]	[0.6020, 0.7014]
DeepSeek-R1-0528-671B (API) NL	[0.7056, 0.8455]	[0.6943, 0.7862]	[0.6091, 0.7019]
DeepSeek-V3-0324-671B(API) NL	[0.7669, 0.8859]	[0.6638, 0.7703]	[0.5711, 0.6682]
Llama-3.3-70B-Instruct NL	[0.7374, 0.8716]	[0.7077, 0.8107]	[0.5250, 0.6285]
Llama-3.3-70B-Instruct DSPy-CoT W2	[0.6522, 0.7969]	[0.6873, 0.7869]	[0.5910, 0.6854]

Table C.1: The bootstrapped 95% confidence intervals (CIs) for macro F1 from nine selected off-the-shelf LLM runs regarding their cross-subset performance.

Model	Overall χ^2 (p)	Comparison	Accuracy A	Accuracy B	p-value	Significant (Bonf.)
Qwen3-4B NL	15.92 (0.0003)	CoCoNot vs IN3	0.81	0.73	0.0961	
		CoCoNot vs CLAMBER	0.81	0.65	0.0003	✓
		IN3 vs CLAMBER	0.73	0.65	0.0176	✗
Qwen3-32B NL	18.10 (0.0001)	CoCoNot vs IN3	0.81	0.72	0.0346	✗
		CoCoNot vs CLAMBER	0.81	0.64	0.00007	✓
		IN3 vs CLAMBER	0.72	0.64	0.0239	✗
Qwen3-4B DSPy-CoT W1	14.30 (0.0008)	CoCoNot vs IN3	0.78	0.67	0.0237	✗
		CoCoNot vs CLAMBER	0.78	0.61	0.00027	✓
		IN3 vs CLAMBER	0.67	0.61	0.1124	✗
Qwen3-4B DSPy-CoT W2	19.82 (0.0000)	CoCoNot vs IN3	0.84	0.7	0.002	✓
		CoCoNot vs CLAMBER	0.84	0.65	0.00001	✓
		IN3 vs CLAMBER	0.7	0.65	0.1484	✗
DeepSeek-R1-Distill-Qwen-14B NL	9.50 (0.0086)	CoCoNot vs IN3	0.78	0.65	0.0048	✓
		CoCoNot vs CLAMBER	0.78	0.66	0.0069	✓
		IN3 vs CLAMBER	0.65	0.66	0.8204	✗
DeepSeek-R1-0528-671B (API) NL	15.03 (0.0005)	CoCoNot vs IN3	0.8077	0.7458	0.1722	✗
		CoCoNot vs CLAMBER	0.8077	0.655	0.0006	✓
		IN3 vs CLAMBER	0.7458	0.655	0.0126	✓
DeepSeek-V3-0324-671B(API) NL	25.87 (0.0000)	CoCoNot vs IN3	0.8462	0.7358	0.0107	✓
		CoCoNot vs CLAMBER	0.8462	0.635	0.000002	✓
		IN3 vs CLAMBER	0.7358	0.635	0.0061	✓
Llama-3.3-70B-Instruct NL	34.52 (0.0000)	CoCoNot vs IN3	0.8205	0.7826	0.407	✗
		CoCoNot vs CLAMBER	0.8205	0.615	0.000006	✓
		IN3 vs CLAMBER	0.7826	0.615	0.000003	✓
Llama-3.3-70B-Instruct DSPy-CoT W2	9.71 (0.0078)	CoCoNot vs IN3	0.75	0.7391	0.8896	✗
		CoCoNot vs CLAMBER	0.75	0.645	0.023	✗
		IN3 vs CLAMBER	0.7391	0.645	0.0102	✓

Table C.2: Results of chi-square (χ^2) tests on nine selected off-the-shelf LLM runs about their cross-subset performance.

Bibliography

Aliannejadi, M., J. Kiseleva, A. Chuklin, J. Dalton, and M. Burtsev (2020). *ConvAI3: Generating Clarifying Questions for Open-Domain Dialogue Systems (ClariQ)*. arXiv: 2009.11352 [cs.CL]. https://arxiv.org/abs/2009.11352.

Bach, K. (1982). “Semantic Nonspecificity and Mixed Quantifiers”. In: *Linguistics and Philosophy* 4.4, pp. 593–605. ISSN: 01650157, 15730549. http://www.jstor.org/stable/25001076 (visited on 07/05/2025).

— (2004). “Context Ex Machina”. In: *Semantics Versus Pragmatics*. Ed. by Z. G. Szabo. Oxford University Press UK, pp. 15–44.

Banerjee, S. and A. Lavie (June 2005). “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments”. In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ed. by J. Goldstein, A. Lavie, C.-Y. Lin, and C. Voss. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 65–72. https://aclanthology.org/W05-0909/.

Belleri, D. (2014). *Semantic Under-Determinacy and Communication*. London/Basingstoke: Palgrave-Macmillan.

Brahman, F., S. Kumar, V. Balachandran, P. Dasigi, V. Pyatkin, A. Ravichander, S. Wiegreffe, N. Dziri, K. R. Chandu, J. Hessel, Y. Tsvetkov, N. A. Smith, Y. Choi, and H. Hajishirzi (2024). “The Art of Saying No: Contextual Noncompliance in Language Models”. In: *CoRR* abs/2407.12043. https://doi.org/10.48550/ARXIV.2407.12043. arXiv: 2407.12043. https://doi.org/10.48550/arXiv.2407.12043.

Bunt, H. (2007). “Semantic Underspecification: Which Technique For What Purpose?” In: *Computing Meaning*. Ed. by H. Bunt and R. Muskens. Dordrecht: Springer Netherlands, pp. 55–85. ISBN: 978-1-4020-5958-2. https://doi.org/10.1007/978-1-4020-5958-2_4. https://doi.org/10.1007/978-1-4020-5958-2_4.

Cardillo, A. (June 13, 2025). *Best 44 Large Language Models (LLMs) in 2025*. https://explodingtopics.com/blog/list-of-llms (visited on 07/12/2025).

Carston, R. (2002). “Pragmatics and Linguistic Underdeterminacy”. In: *Thoughts and Utterances*. John Wiley & Sons, Ltd. Chap. 1, pp. 15–93. ISBN: 9780470754603. https://doi.org/https://doi.org/10.1002/9780470754603.ch2. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470754603.ch2. https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470754603.ch2.

Chang, Y., X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie (Mar. 2024). “A Survey on Evaluation of Large Language Models”. In: *ACM Trans. Intell. Syst. Technol.* 15.3. ISSN: 2157-6904. https://doi.org/10.1145/3641289. https://doi.org/10.1145/3641289.

Clark, P., I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord (2018). *Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge*. arXiv: 1803.05457 [cs.AI]. https://arxiv.org/abs/1803.05457.

Comanici, G., E. Bieber, M. Schaekermann, I. Pasupat, N. Sachdeva, I. Dhillon, M. Blistein, O. Ram, D. Zhang, E. Rosen, L. Marris, S. Petulla, C. Gaffney, A. Aharoni, N. Lintz, T. C. Pais, and H. Jacobsson (2025). *Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities*. arXiv: 2507.06261 [cs.CL]. https://arxiv.org/abs/2507.06261.

DeepSeek-AI, D. Guo, et al. (2025). *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. arXiv: 2501.12948.

DeepSeek-AI, A. Liu, et al. (2024). *DeepSeek-V3 Technical Report*. arXiv: 2412.19437.

Dong, Q., L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, B. Chang, X. Sun, L. Li, and Z. Sui (Nov. 2024). "A Survey on In-context Learning". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Y. Al-Onaizan, M. Bansal, and Y.-N. Chen. Miami, Florida, USA: Association for Computational Linguistics, pp. 1107–1128. https://doi.org/10.18653/v1/2024.emnlp-main.64. https://aclanthology.org/2024.emnlp-main.64/.

Egg, M. (2010). "Semantic Underspecification". In: *Language and Linguistics Compass* 4.3, pp. 166–181. https://doi.org/https://doi.org/10.1111/j.1749-818X.2010.00188.x. eprint: https://compass.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1749-818X.2010.00188.x. https://compass.onlinelibrary.wiley.com/doi/abs/10.1111/j.1749-818X.2010.00188.x.

Frisson, S. (Feb. 2009). "Semantic Underspecification in Language Processing". In: *Language and Linguistics Compass* 3, pp. 111–127. https://doi.org/10.1111/j.1749-818X.2008.00104.x.

Garg, S., T. Vu, and A. Moschitti (Apr. 2020). "TANDA: Transfer and Adapt Pre-Trained Transformer Models for Answer Sentence Selection". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34, pp. 7780–7788. https://doi.org/10.1609/aaai.v34i05.6282.

Gillon, B. S. (1990). "Ambiguity, Generality, and Indeterminacy: Tests and Definitions". In: *Synthese* 85.3, pp. 391–416. https://doi.org/10.1007/bf00484835.

Grice, H. P. (1969). "Utterer's Meaning and Intentions". In: *Philosophical Review* 78.2, pp. 147–177. https://doi.org/10.2307/2184179.

— (1975). "Logic and Conversation". In: *The logic of grammar*. Ed. by D. Davidson. Dickenson Pub. Co., pp. 64–75.

Grice, H. P. (1957). "Meaning". In: *Philosophical Review* 66.3, pp. 377–388. https://doi.org/10.2307/2182440.

Gross, S. (2001). *Essays on Linguistic Context-sensitivity and Its Philosophical Significance*. Literary Criticism and Cultural Theory: The Interaction of T. Routledge. ISBN: 9780815340386. https://books.google.co.uk/books?id=egYEcp1eSA4C.

Guo, M., M. Zhang, S. Reddy, and M. Alikhani (2021). "Abg-CoQA: Clarifying Ambiguity in Conversational Question Answering". In: *3rd Conference on Automated Knowledge Base Construction, AKBC 2021, Virtual, October 4-8, 2021*. Ed. by D. Chen, J. Berant, A. McCallum, and S. S. 0001. https://doi.org/10.24432/C5F30Z. https://doi.org/10.24432/C5F30Z.

Harris, D. W. (2020a). “Intention Recognition and its Psychological Underpinnings”. Unpublished book manuscript, CUNY Graduate Center. https://danielwharris.com/book/DanielWHarris-IntentionRecognition.pdf.

— (2020b). “What makes human communication special?” Unpublished book manuscript, CUNY Graduate Center. https://danielwharris.com/book/DanielWHarris-WhatMakesHumanCommunicationSpecial.pdf.

Heim, I. (Sept. 2000). “Degree Operators and Scope”. In: *Proceedings from Semantics and Linguistic Theory* 10, pp. 214–239. https://doi.org/10.3765/salt.v10i0.2722.

Heim, I. and A. Kratzer (1998a). “More on English: Nonverbal Predicates, Modifiers, Definite Descriptions”. In: *Semantics in Generative Grammar*. Malden, MA: Blackwell, pp. 61–86. ISBN: 978-0-631-19713-3.

— (1998b). “Quantification and Grammar”. In: *Semantics in Generative Grammar*. Malden, MA: Blackwell, pp. 178–208. ISBN: 978-0-631-19713-3.

Hepp, A. (May 2020). “Artificial companions, social bots and work bots: communicative robots as research objects of media and communication studies”. In: *Media, Culture & Society* 42, p. 016344372091641. https://doi.org/10.1177/0163443720916412.

Herlihy, C., J. Neville, T. Schnabel, and A. Swaminathan (2024). “On Overcoming Miscalibrated Conversational Priors in LLM-based Chatbots”. In: *ArXiv* abs/2406.01633. https://arxiv.org/abs/2406.01633.

Huber, P., A. Aghajanyan, B. Oguz, D. Okhonko, S. Yih, S. Gupta, and X. Chen (July 2022). “CCQA: A New Web-Scale Question Answering Dataset for Model Pre-Training”. In: *Findings of the Association for Computational Linguistics: NAACL 2022*. Ed. by M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz. Seattle, United States: Association for Computational Linguistics, pp. 2402–2420. https://doi.org/10.18653/v1/2022.findings-naacl.184. https://aclanthology.org/2022.findings-naacl.184/.

Jiang, A. Q., A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed (2023). *Mistral 7B*. arXiv: 2310.06825 [cs.CL]. https://arxiv.org/abs/2310.06825.

Joshi, M., E. Choi, D. Weld, and L. Zettlemoyer (July 2017). “TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by R. Barzilay and M.-Y. Kan. Vancouver, Canada: Association for Computational Linguistics, pp. 1601–1611. https://doi.org/10.18653/v1/P17-1147. https://aclanthology.org/P17-1147/.

Jurafsky, D. and J. H. Martin (2025). “Context-Free Grammars and Constituency Parsing”. In: *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd. Online manuscript released January 12, 2025. https://web.stanford.edu/~jurafsky/slp3/18.pdf.

Kennedy, C. (1997). *Projecting the Adjective: The Syntax and Semantics of Gradability and Comparison*. Routledge. ISBN: 9780203055458. https://doi.org/10.4324/9780203055458.

— (2011). “23. Ambiguity and vagueness: An overview”. In: *Volume 1*. Ed. by C. Maienborn, K. von Heusinger, and P. Portner. Berlin, Boston: De Gruyter Mouton, pp. 507–535. ISBN:

9783110226614. https://doi.org/doi:10.1515/9783110226614.507. https://doi.org/10.1515/9783110226614.507.

Kennedy, C. and L. Mcnally (2010). “Color, Context, and Compositionality”. In: *Synthese* 174.1, pp. 79–98. https://doi.org/10.1007/s11229-009-9685-7.

Khashabi, D., S. Min, T. Khot, A. Sabharwal, O. Tafjord, P. Clark, and H. Hajishirzi (Nov. 2020). “UNIFIEDQA: Crossing Format Boundaries with a Single QA System”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Ed. by T. Cohn, Y. He, and Y. Liu. Online: Association for Computational Linguistics, pp. 1896–1907. https://doi.org/10.18653/v1/2020.findings-emnlp.171. https://aclanthology.org/2020.findings-emnlp.171/.

Khattab, O., A. Singhvi, P. Maheshwari, Z. Zhang, K. Santhanam, S. Vardhamanan, S. Haq, A. Sharma, T. T. Joshi, H. Moazam, H. Miller, M. Zaharia, and C. Potts (2024). “DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines”. In: *The Twelfth International Conference on Learning Representations*. https://arxiv.org/abs/2310.03714.

Kim, H. J., Y. Kim, C. Park, J. Kim, C. Park, K. M. Yoo, S.-g. Lee, and T. Kim (Nov. 2024). “Aligning Language Models to Explicitly Handle Ambiguity”. In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Y. Al-Onaizan, M. Bansal, and Y.-N. Chen. Miami, Florida, USA: Association for Computational Linguistics, pp. 1989–2007. https://doi.org/10.18653/v1/2024.emnlp-main.119. https://aclanthology.org/2024.emnlp-main.119/.

Köpf, A., Y. Kilcher, D. von Rütte, S. Anagnostidis, Z.-R. Tam, K. Stevens, A. Barhoum, N. M. Duc, O. Stanley, R. Nagyfi, S. ES, S. Suri, D. Glushkov, A. Dantuluri, A. Maguire, C. Schuhmann, H. Nguyen, and A. Mattick (2023). *OpenAssistant Conversations – Democratizing Large Language Model Alignment*. arXiv: 2304.07327 [cs.CL]. https://arxiv.org/abs/2304.07327.

Kuhn, L., Y. Gal, and S. Farquhar (Dec. 2022). “CLAM: Selective Clarification for Ambiguous Questions with Large Language Models”. In: *ArXiv*. https://doi.org/10.48550/arXiv.2212.07769.

Kwiatkowski, T., J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov (Aug. 2019). “Natural Questions: A Benchmark for Question Answering Research”. In: *Transactions of the Association for Computational Linguistics* 7, pp. 453–466. ISSN: 2307-387X. https://doi.org/10.1162/tacl_a_00276. eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00276/1923288/tacl_a_00276.pdf. https://doi.org/10.1162/tacl%5C_a%5C_00276.

Lee, D., S. Kim, M. Lee, H. Lee, J. Park, S.-W. Lee, and K. Jung (Dec. 2023). “Asking Clarification Questions to Handle Ambiguity in Open-Domain QA”. In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by H. Bouamor, J. Pino, and K. Bali. Singapore: Association for Computational Linguistics, pp. 11526–11544. https://doi.org/10.18653/v1/2023.findings-emnlp.772. https://aclanthology.org/2023.findings-emnlp.772/.

Lee, K., M.-W. Chang, and K. Toutanova (July 2019). “Latent Retrieval for Weakly Supervised Open Domain Question Answering”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by A. Korhonen, D. Traum, and L. Màrquez. Florence, Italy: Association for Computational Linguistics, pp. 6086–6096. https://doi.org/10.18653/v1/P19-1612. https://aclanthology.org/P19-1612/.

Levinson, S. C. (Apr. 2000). *Presumptive Meanings: The Theory of Generalized Conversational Implicature*. The MIT Press. ISBN: 9780262278256. https://doi.org/10.7551/mitpress/5526.001.0001.

Lin, C.-Y. (July 2004). "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, pp. 74–81. https://aclanthology.org/W04-1013/.

Lin, S., J. Hilton, and O. Evans (May 2022). "TruthfulQA: Measuring How Models Mimic Human Falsehoods". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by S. Muresan, P. Nakov, and A. Villavicencio. Dublin, Ireland: Association for Computational Linguistics, pp. 3214–3252. https://doi.org/10.18653/v1/2022.acl-long.229. https://aclanthology.org/2022.acl-long.229/.

Liu, A., Z. Wu, J. Michael, A. Suhr, P. West, A. Koller, S. Swayamdipta, N. Smith, and Y. Choi (Dec. 2023). "We're Afraid Language Models Aren't Modeling Ambiguity". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by H. Bouamor, J. Pino, and K. Bali. Singapore: Association for Computational Linguistics, pp. 790–807. https://doi.org/10.18653/v1/2023.emnlp-main.51. https://aclanthology.org/2023.emnlp-main.51/.

Meta (2024a). *Llama 3.2 Model Card*. Accessed: 2025-07-13. https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md.

— (2024b). *Llama 3.2: Revolutionizing edge AI and vision with open, customizable models*. Accessed: 2025-07-13. https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/.

— (2024c). *Llama 3.3 Model Card*. Accessed: 2025-07-13. https://github.com/meta-llama/llama-models/blob/main/models/llama3_3/MODEL_CARD.md.

Meta AI (2024). *LLaMA 3.3-70B-Instruct*. Hugging Face model page. Retrieved June 17, 2025. https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct.

Min, S., J. Michael, H. Hajishirzi, and L. Zettlemoyer (Nov. 2020). "AmbigQA: Answering Ambiguous Open-domain Questions". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by B. Webber, T. Cohn, Y. He, and Y. Liu. Online: Association for Computational Linguistics, pp. 5783–5797. https://doi.org/10.18653/v1/2020.emnlp-main.466. https://aclanthology.org/2020.emnlp-main.466/.

Nieuwland, M. S. and J. J. A. Van Berkum (2008). "The Neurocognition of Referential Ambiguity in Language Comprehension". In: *Language and Linguistics Compass* 2.4, pp. 603–630. https://doi.org/https://doi.org/10.1111/j.1749-818X.2008.00070.x. https://compass.onlinelibrary.wiley.com/doi/abs/10.1111/j.1749-818X.2008.00070.x.

OpenAI (2022). *Introducing ChatGPT*. Retrieved June 15, 2025. https://openai.com/index/chatgpt/%5C#OpenAI.

— (2024a). *Memory and new controls for ChatGPT*. Retrieved June 16, 2025. https://openai.com/index/memory-and-new-controls-for-chatgpt/.

— (2024b). *OpenAI o1*. Accessed: 2025-07-13. https://openai.com/o1/.

OpenAI, J. Achiam, et al. (2024). *GPT-4 Technical Report*. arXiv: 2303.08774 [cs.CL]. https://arxiv.org/abs/2303.08774.

OpenAI, A. Hurst, et al. (2024). *GPT-4o System Card*. arXiv: 2410.21276 [cs.CL]. https://arxiv.org/abs/2410.21276.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu (July 2002). "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Ed. by P. Isabelle, E. Charniak, and D. Lin. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp. 311–318. https://doi.org/10.3115/1073083.1073135. https://aclanthology.org/P02-1040/.

Partee, B. H. (1986). "Noun Phrase Interpretation and Type-Shifting Principles". In: *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers*. Ed. by J. Groenendijk, D. de Jongh, and M. Stokhof. Berlin, Boston: De Gruyter, pp. 115–144. ISBN: 9783112420027. https://doi.org/doi:10.1515/9783112420027-006. https://doi.org/10.1515/9783112420027-006.

Peter, J., T. Araujo, C. Ischen, S. J. Shaikh, M. J. van der Goot, and C. L. van Straten (2024). "12. Human–Machine Communication". In: *Fundamental Insights from the Amsterdam School of Communication Research*. Ed. by T. Araujo and P. Neijens. Amsterdam: Amsterdam University Press, pp. 205–220. ISBN: 9789048560608. https://doi.org/doi:10.1515/9789048560608-013. https://doi.org/10.1515/9789048560608-013.

Pezzelle, S. (July 2023). "Dealing with Semantic Underspecification in Multimodal NLP". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by A. Rogers, J. Boyd-Graber, and N. Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 12098–12112. https://doi.org/10.18653/v1/2023.acl-long.675. https://aclanthology.org/2023.acl-long.675/.

Piantadosi, S. T., H. Tily, and E. Gibson (2012). "The Communicative Function of Ambiguity in Language". In: *Cognition* 122.3, pp. 280–291. https://doi.org/10.1016/j.cognition.2011.10.004.

Qian, C., B. He, Z. Zhuang, J. Deng, Y. Qin, X. Cong, Z. Zhang, J. Zhou, Y. Lin, Z. Liu, and M. Sun (Aug. 2024). "Tell Me More! Towards Implicit User Intention Understanding of Language Model Driven Agents". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by L.-W. Ku, A. Martins, and V. Srikumar. Bangkok, Thailand: Association for Computational Linguistics, pp. 1088–1113. https://doi.org/10.18653/v1/2024.acl-long.61. https://aclanthology.org/2024.acl-long.61/.

Rein, D., B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman (2023). *GPQA: A Graduate-Level Google-Proof Q&A Benchmark*. arXiv: 2311.12022 [cs.AI]. https://arxiv.org/abs/2311.12022.

Roberts, A., C. Raffel, and N. Shazeer (Nov. 2020). "How Much Knowledge Can You Pack Into the Parameters of a Language Model?" In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by B. Webber, T. Cohn, Y. He, and Y. Liu. Online: Association for Computational Linguistics, pp. 5418–5426. https://doi.org/10.18653/v1/2020.emnlp-main.437. https://aclanthology.org/2020.emnlp-main.437/.

Schwarzchild, R. and K. Wilkinson (Mar. 2002). "Quantifiers in Comparatives: A Semantics of Degree Based on Intervals". In: *Natural Language Semantics* 10, pp. 1–41. https://doi.org/10.1023/A:1015545424775.

Sennet, A. (2023). “Ambiguity”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by E. N. Zalta and U. Nodelman. Summer 2023. Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/sum2023/entries/ambiguity/.

Shakudo (July 7, 2025). *Top 9 Large Language Models (LLMs) as of July 2025*. https://www.shakudo.io/blog/top-9-large-language-models (visited on 07/12/2025).

Shi, Z., G. Castellucci, S. Filice, S. Kuzi, E. Kravi, E. Agichtein, O. Rokhlenko, and S. Malmasi (May 2025). “Ambiguity Detection and Uncertainty Calibration for Question Answering with Large Language Models”. In: *Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP 2025)*. Ed. by T. Cao, A. Das, T. Kumarage, Y. Wan, S. Krishna, N. Mehrabi, J. Dhamala, A. Ramakrishna, A. Galystan, A. Kumar, R. Gupta, and K.-W. Chang. Albuquerque, New Mexico: Association for Computational Linguistics, pp. 41–55. ISBN: 979-8-89176-233-6. https://doi.org/10.18653/v1/2025.trustnlp-main.4. https://aclanthology.org/2025.trustnlp-main.4/.

Sorensen, R. (2023). “Vagueness”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by E. N. Zalta and U. Nodelman. Winter 2023. Metaphysics Research Lab, Stanford University.

Stalnaker, R. (2002). “Common Ground”. In: *Linguistics and Philosophy* 25.5, pp. 701–721. https://doi.org/10.1023/a:1020867916902.

Tamkin, A., M. McCain, K. Handa, E. Durmus, L. Lovitt, A. Rathi, S. Huang, A. Mountfield, J. Hong, S. Ritchie, M. Stern, B. Clarke, L. Goldberg, T. Sumers, J. Mueller, W. McEachen, W. Mitchell, S. Carter, J. Clark, and D. Ganguli (Dec. 2024). “Clio: Privacy-Preserving Insights into Real-World AI Use”. In: https://doi.org/10.48550/arXiv.2412.13678.

Tanjim, M. M., X. Chen, V. Bursztyn, U. Bhattacharya, M. Tùng, V. Muppala, A. Maharaj, S. Mitra, E. Koh, Y. Li, and K. Russell (Feb. 2025). “Detecting Ambiguities to Guide Query Rewrite for Robust Conversations in Enterprise AI Assistants”. In: *arXiv*. https://doi.org/10.48550/arXiv.2502.00537.

Tanjim, M. M., Y. In, X. Chen, V. Bursztyn, R. Rossi, S. Kim, G.-J. Ren, V. Muppala, S. Jiang, Y. Kim, and C. Park (May 2025). “Disambiguation in Conversational Question Answering in the Era of LLM: A Survey”. In: *arXiv*. https://doi.org/10.48550/arXiv.2505.12543.

The DSPy Team (2025). *DSPy: Programming, Not Prompting, LMs.* Accessed: 2025-07-13. https://dspy.ai/.

Touvron, H., L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, and T. Scialom (July 2023). “Llama 2: Open Foundation and Fine-Tuned Chat Models”. In: *arXiv*. https://doi.org/10.48550/arXiv.2307.09288.

Travis, C. (1997). “Pragmatics”. In: *A Companion to the Philosophy of Language*. Ed. by B. Hale, C. Wright, and A. Miller. Wiley Blackwell, pp. 127–150.

Trott, S. and B. K. Bergen (2022). “Languages are efficient, but for whom?” In: *Cognition* 225. https://doi.org/10.1016/j.cognition.2022.105094.

van Rooij, R. (2011). “Vagueness and Linguistics”. In: *Vagueness: A Guide*. Ed. by G. Ronzitti. Dordrecht: Springer Netherlands, pp. 123–170. ISBN: 978-94-007-0375-9. https://doi.org/10.1007/978-94-007-0375-9_6. https://doi.org/10.1007/978-94-007-0375-9_6.

Von Werra, L., L. Tunstall, A. Thakur, S. Luccioni, T. Thrush, A. Piktus, F. Marty, N. Rajani, V. Mustar, and H. Ngo (Dec. 2022). “Evaluate & Evaluation on the Hub: Better Best Practices for Data and Model Measurements”. In: *Proceedings of the 2022 Conference on Empirical Methods in*

Natural Language Processing: System Demonstrations. Ed. by W. Che and E. Shutova. Abu Dhabi, UAE: Association for Computational Linguistics, pp. 128–136. https://doi.org/10.18653/v1/2022.emnlp-demos.13. https://aclanthology.org/2022.emnlp-demos.13/.

Wahde, M. and M. Virgolin (2022). "Conversational Agents: Theory and Applications". In: *Handbook on Computer Learning and Intelligence*. World Scientific. Chap. Chapter 12, pp. 497–544. https://doi.org/10.1142/9789811247323_0012. eprint: https://www.worldscientific.com/doi/pdf/10.1142/9789811247323_0012. https://www.worldscientific.com/doi/abs/10.1142/9789811247323_0012.

Wang, J., W. Ma, P. Sun, M. Zhang, and J.-Y. Nie (2024). "Understanding User Experience in Large Language Model Interactions". In: *ArXiv* abs/2401.08329. https://arxiv.org/abs/2401.08329.

Wang, Y., Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi (July 2023). "Self-Instruct: Aligning Language Models with Self-Generated Instructions". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by A. Rogers, J. Boyd-Graber, and N. Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 13484–13508. https://doi.org/10.18653/v1/2023.acl-long.754. https://aclanthology.org/2023.acl-long.754/.

Wei, J., N. Karina, H. W. Chung, Y. J. Jiao, S. Papay, A. Glaese, J. Schulman, and W. Fedus (2024). *Measuring short-form factuality in large language models*. arXiv: 2411.04368 [cs.CL]. https://arxiv.org/abs/2411.04368.

Wei, J., X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou (2023). *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. arXiv: 2201.11903 [cs.CL]. https://arxiv.org/abs/2201.11903.

Wildenburg, F., M. Hanna, and S. Pezzelle (Aug. 2024). "Do Pre-Trained Language Models Detect and Understand Semantic Underspecification? Ask the DUST!" In: *Findings of the Association for Computational Linguistics: ACL 2024*. Ed. by L.-W. Ku, A. Martins, and V. Srikumar. Bangkok, Thailand: Association for Computational Linguistics, pp. 9598–9613. https://doi.org/10.18653/v1/2024.findings-acl.572. https://aclanthology.org/2024.findings-acl.572/.

Xiang, Y. (2017). *Quantifier Raising and Scope Ambiguity*. Accessed: 2025-07-28. https://scholar.harvard.edu/files/yxiang/files/ho14-quantifier_raising_and_scope_ambiguity.pdf.

Yang, A., A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al. (2025). *Qwen3 Technical Report*. arXiv: 2505.09388 [cs.CL]. https://arxiv.org/abs/2505.09388.

Yang, Z., P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning (2018). "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Ed. by E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii. Brussels, Belgium: Association for Computational Linguistics, pp. 2369–2380. https://doi.org/10.18653/v1/D18-1259. https://aclanthology.org/D18-1259/.

Zhang, M. and E. Choi (Nov. 2021). "SituatedQA: Incorporating Extra-Linguistic Contexts into QA". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 7371–7387. https://doi.org/10.18653/v1/2021.emnlp-main.586. https://aclanthology.org/2021.emnlp-main.586/.

Zhang, M. J. and E. Choi (Apr. 2025). “Clarify When Necessary: Resolving Ambiguity Through Interaction with LMs”. In: *Findings of the Association for Computational Linguistics: NAACL 2025*. Ed. by L. Chiruzzo, A. Ritter, and L. Wang. Albuquerque, New Mexico: Association for Computational Linguistics, pp. 5526–5543. ISBN: 979-8-89176-195-7. https://doi.org/10.18653/v1/2025.findings-naacl.306. https://aclanthology.org/2025.findings-naacl.306/.

Zhang, T., V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi (2020). *BERTScore: Evaluating Text Generation with BERT*. arXiv: 1904.09675 [cs.CL]. https://arxiv.org/abs/1904.09675.

Zhang, T., P. Qin, Y. Deng, C. Huang, W. Lei, J. Liu, D. Jin, H. Liang, and T.-S. Chua (Aug. 2024). “CLAMBER: A Benchmark of Identifying and Clarifying Ambiguous Information Needs in Large Language Models”. In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by L.-W. Ku, A. Martins, and V. Srikumar. Bangkok, Thailand: Association for Computational Linguistics, pp. 10746–10766. https://doi.org/10.18653/v1/2024.acl-long.578. https://aclanthology.org/2024.acl-long.578/.

Zwicky, A. M. and J. M. Sadock (1975). “Ambiguity Tests and How to Fail Them”. In: *Syntax and Semantics: volume 4*. https://brill.com/display/book/edcoll/9789004368828/BP000002.xml.