The advancement of pretrained language models (LMs) has led to their increasing deployment in real-world applications, including coding assistants, virtual customer support, writing critiquers, and more. Such applications have become feasible due to the growing capabilities of LMs in understanding and reasoning across diverse types of texts.

While the increasing scale of pretrained LMs has led to notable advancements in their reasoning capabilities, their reliability remains concerning. LMs are prone to capturing surface patterns or dataset artifacts rather than learning deep reasoning process, rendering them susceptible to adversarial attacks and constraining their generalizability. Moreover, the intrinsic limitations of pretrained LMs, arising from their architectural constraints and the finite number of tokens they can process, impede their ability to scale to complex compositional reasoning tasks effectively. At the same time, when LMs fail to handle complex tasks, the internal reasoning process often lacks interpretability and is hard to debug.

My research aims to leverage explanations to steer LMs in performing complex reasoning reliably. I have developed methods that use explanations within two paradigms. The first paradigm uses post-hoc interventions on model predictions based on the explanations (Figure 1, top). Beginning with a model prediction and the corresponding explanation, we develop methods to calibrate model prediction based on the reliability of reasoning process revealed by the explanation. The second paradigm aims to incorporate explanations into the supervision alongside the correct predictions to the input (Figure 1, bottom), thereby enabling the model to learn from reasoning as demonstrated by the explanations. Following the two paradigms, I have explored the use of explanations across a diverse set of tasks requiring reasoning over various text types. In particular, my work has delved into program synthesis from natural language descriptions, wherein I have explored integrating programming language techniques to effectively utilize I/O examples, common specifications that users use to express their intents.

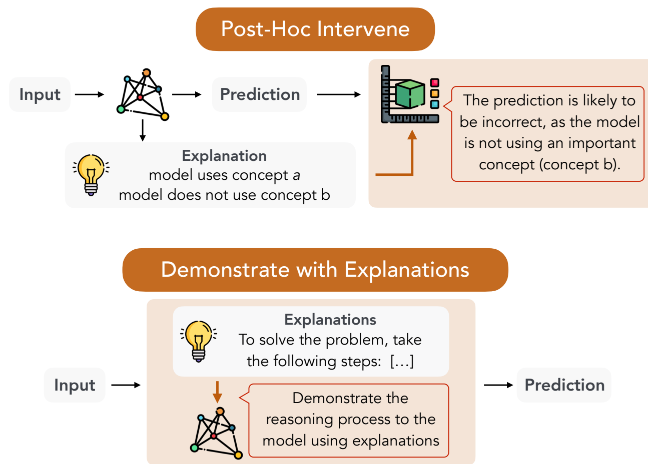

Figure 1: Two paradigms of my research on steering textual reasoning with explanations: intervening on the predictions post-hoc based on explanations or demonstrating the reasoning process to LMs with explanations.

My ultimate goal is to augment human capabilities with LMs in various tasks demanding deep reasoning (such as data analytics and programming), surpassing the efficacy and efficiency achievable by humans alone. I believe that leveraging explanations is crucial to enable robust reasoning skills and effective human-LM collaboration, which are essential for realizing this goal.

Calibrating Model Predictions with Explanations

A central desiderata of explanations for NLP models lies in assisting humans to interpret model predictions, particularly model failures, to propose potential improvements. My research on explanations extends beyond just providing interpretations of predictions; I further strive to automate the use of explanations to better understand LMs at a behavior level and to improve their predictions post-hoc, without requiring heavy human effort.

I first investigated how explanations can help construct a mechanistic understanding of model behavior, using challenging QA problems as a case study. We began by evaluating several explanation techniques based on whether they can reveal model behavior. Concretely, given a model prediction, the explanation for a prediction would hint about the model's operational mechanism (for example, how certain interactions within the input influence the model's prediction). We can perturb the input in a meaningful way, yielding realistic counterfactuals that disrupt the patterns. The model predictions on counterfactuals allow us to validate whether the hypothesized model behavior suggested by an explanation correctly aligns with the model's true behavior, hence providing a way for evaluating explanation techniques. Our work suggests that explanations generated by appropriate technique in a suitable format align well with model behavior in response to these counterfactuals (Ye et al., 2021b; Singhal et al., 2022).

Having evidence on the connection between model behavior and model explanations, we build a calibration framework that utilizes explanations to improve black-box models, which have become more and more accessible as API services throughout the Internet, in the selective prediction setting. The selective prediction setting allows models to selectively predict only on high-confident examples so as to abstain from making errors. At a high level, our framework assesses the correctness of model predictions based on the reliability of the reasoning process, revealed by explanations. As shown in Figure 2, we extract features that can describe the "reasoning process" disclosed by the explanations, which are then used by a trained calibrator to judge the robustness of predictions.

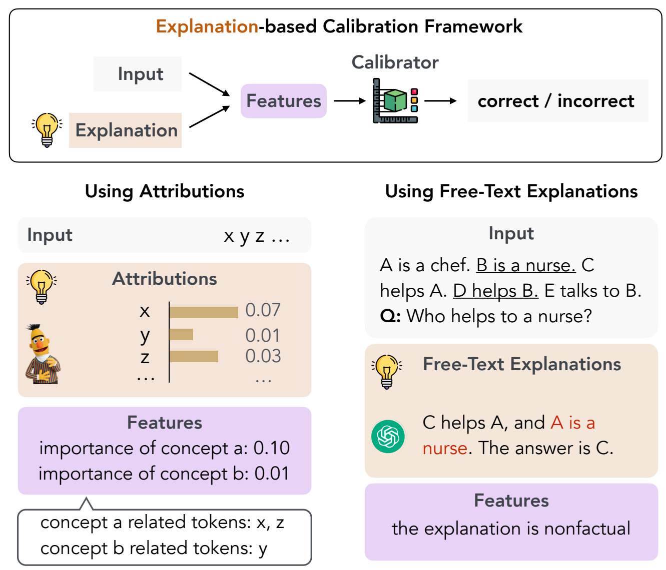

Figure 2: Calibration framework that assesses the correctness of model predictions based on explanations.

Such a calibration framework can be applied on calibrating BERT-based models using attributions generated (Ye and Durrett, 2022a), where we rely on features describing the importance of certain parts of input or certain linguistic features. Our framework can also be used on calibrating predictions of large language models (e.g., GPT-3) with free-text explanations. We prompted GPT-3 with free-text explanations for textual reasoning tasks and observed mild performance improvements compared to not using rationales in prompts (Ye and Durrett, 2022b). However, explanations generated by GPT-3 can be inconsistent and even nonfactual, contradicting the contexts specified in the prompt. But we still find flawed explanations to be useful, as their factuality correlates well with the accuracy of predictions, which we use as leverage to calibrate predictions.

Teaching Large Language Models to Reason with Explanations

My work has further explored how to closely integrate explanations to demonstrate the reasoning process for models, especially LLMs. We are among the first to study the usage of explanations for textual reasoning in prompting LLMs, showing their effectiveness as well as unreliability (Ye and Durrett, 2022b). As the benefits that LLMs can gain from explanations highly depend on the quality of explanations, my work particularly focuses on how to construct maximally effective explanations for LLMs.

One prominent way to construct explanations for LLMs is to showcase imperative procedures for solving reasoning problems. Our work found that while imperative explanations work well for tasks that only require forward reasoning (e.g., straightforward arithmetic), it tends to be inadequate for complex problems requiring more sophisticated planning and search, as the imperative explanations require LLMs to plan and execute complex reasoning procedure, which are tasks that LLMs are intrinsically challenged by. We propose a satisfiability-aided language modeling (SatLM) approach that supervises LLMs with declarative formal specifications as explanations and leverages an automated theorem prover to derive the final answer (Ye et al., 2023a). Our approach teaches LLMs to focus on understanding and parsing the NL problems while offloading the planning and execution task to an automated theorem prover, which leads to significant improvements over using imperative explanations (CoTs and programs) across a diverse set of reasoning problems.

Even when knowing the suitable formalisation of explanations, crafting good explanations typically requires expertise and manual engineering. Subtly different explanations can yield widely varying downstream task accuracy. My work also tackles the problem of optimizing explanation-infused prompts in a black-box fashion. We first generate sets of candidate explanations for each example in the prompt using a leave-one-out scheme, then find an effective combination of these explanations by searching over combinations of explanations to find one that yields high performance against a silver-labeled development set (Ye and Durrett, 2023). Our optimization technique can effectively improve prompts over crowdworker annotations. In addition to the verbalization of explanations, the exemplars included in the prompts also impact the effectiveness (Lee et al., 2023). We study how to form maximally effective sets of explanations for solving a given test query. Through a series of probes, we find that LLMs can benefit from the complementarity of the explanation set: diverse reasoning skills shown by different exemplars can lead to better performance (Ye et al., 2023b). Therefore, we propose a maximal marginal relevance-based exemplar selection approach for constructing exemplar sets that are both relevant as well as complementary, which successfully improves the in-context learning performance across several real-world tasks on multiple LLMs.

Program Synthesis from Natural Language and Examples

Being able to parse NL descriptions for complex tasks into executable logical forms is an important and promising reasoning capability of LMs. However, directly generating complex programs from NL descriptions is challenging, as language is inherently ambiguous (Ye et al., 2020a). E.g., a short description *"comma separated columns of two or three digits and letters."* refer to multiple structurally complex regexes that possibly match its intents. Users often include additional I/O examples to further convey their intents. We are among the first to use multimodal inputs consisting of both NL descriptions and I/O examples, which scales up code generation to more complex programs.

Solving multimodal synthesis problems requires efficient searching under a combination of hard constraints (I/O) and soft constraints (NL). We tackled this by integrating program synthesis techniques into code generation models. We proposed a sketch-driven approach that first parses NL descriptions into sketches, and then employs an enumerative synthesizer to search for I/O-satisfying programs under execution guidance (Ye et al., 2020b). In this way, we use neural models to generate high-level structure of the programs, and let a program synthesizer efficiently resolve underspecifity and ambiguity in NL with the guidance of I/O examples. We also built a neural program synthesizer that directly uses scores from models and efficiently searches for optimal programs defined by model scores (Ye et al., 2021a). Our synthesizers successfully scale to a complex regex synthesis dataset (Ye et al., 2020a) and even real-world regexes from StackOverflow posts (Chen et al., 2020). In addition to program synthesis, I also explored the integration of enumeration and execution guidance in KBQA, which also sees substantial benefits (Ye et al., 2022).

Future Directions

Effective human-LM collaboration with explanations as the vehicle Despite the variety of explanation forms and generation techniques available, extensive research suggests that explanations have only achieved limited success in aiding humans across many tasks. To this end, I am interested in developing a more effective protocol for human-LM collaboration, where LMs can take initiatives to seek explanations towards collaboratively solving a problem. Specifically, LMs can actively express their uncertainties and request clarifications on sub-tasks or data instances where their confidence is low. In turn, humans can inspect their explanations on these instances as well and provide targeted feedback to guide LM behavior. This protocol raises intriguing questions regarding the most effective forms of explanations (such as case-based or contrastive explanations) and feedback (like natural language instructions or preferences regarding explanations). Building on my previous research on interactive systems that assist experts in improving models through insights into data instances or model parameters (Xiang et al., 2019; Yang et al., 2022), I believe that explanations can be the vehicle to enable more effective human-LM collaboration.

Learning to interact with symbolic executors with explanations When writing a complex formal specification or a complex program, experts typically operate iteratively, engaging actively with symbolic executors (like SMT solvers or program interpreters) to receive feedback and refine their solutions. My past work has shown that equipping models with program synthesizers (Ye et al., 2020b) leads to more efficient program generation. I believe that teaching LLMs to utilize more granular feedback will further scale up the complexity of problems they can handle. For instance, LMs can learn to use program profiling tools or debuggers (like GDB) to analyze the bottlenecks or failures in model-generated programs; LMs can also learn from the feedback from SMT solvers (such the unsat core) to pinpoint the errors in specification. This presents a challenging task due to the difficulty of collecting high-quality supervision and complex action spaces for using certain tools. I am interested in developing effective explanations to facilitate LMs to acquire such sophisticated skills.

Combining NL explanations and formal explanations for flexible and robust reasoning While utilizing formal specifications to teach LMs guarantees soundness in reasoning, there are many reasoning tasks that involve both "hard" constraints as well as rules that are tricky to articulate solely through formal specifications. For instance, in legal reasoning, certain prerequisites must be met for a verdict of guilt, yet whether a suspect in a case fulfills these criteria can be debatable and difficult to verify with formal specifications alone. Furthermore, some problems might require a blend of reasoning types, some that are suitable for solvers like deductive reasoning, and others less so, such as defeasible reasoning and reasoning by analogy. I believe a system that can make use of both hard formal specifications (like SMT formulas) as well NL statements (NL proposition based on commonsense) would take benefit of both the reliability of symbolic systems as well as the flexibility of LM's capabilities in NL reasoning.

Building resources towards complex reasoning in real-world applications LMs have significantly advanced in their reasoning capabilities. However, there is a notable gap in resources for benchmarking and enhancing LMs' reasoning abilities in a manner that aligns with actual user needs in real-world scenarios. Some of my previous work has involved compiling datasets that test reasoning with natural narrative text (Sprague et al., 2023) or real user queries (Ye et al., 2020a). Moving forward, my aim is to develop resources that facilitate the application of LLMs in real-world settings where language-based reasoning is crucial. One area of focus is data analysis tasks, which require in-depth examination of data to extract meaningful insights. For instance, how can we enable LMs to analyze sales data, pinpointing key factors and customer segments that could drive revenue growth? This task demands both data-driven reasoning and commonsense reasoning. I intend to establish datasets and platforms that support research in this direction. Furthermore, I view explanations as a powerful force to enable LMs to aid humans in various real-world applications like data analytics, potentially leading to results beyond human capabilities alone.

References

Qiaochu Chen, Xinyu Wang, Xi Ye, Greg Durrett, and Isil Dillig. 2020. Multi-Modal Synthesis of Regular Expressions. In *Conference on Programming Language Design and Implementation (PLDI)*.

Yoonsang Lee, Pranav Atreya, Xi Ye, and Eunsol Choi. 2023. Crafting in-context examples according to lms' parametric knowledge. *ArXiv*.

Prasann Singhal, Jarad Forristal, Xi Ye, and Greg Durrett. 2022. Assessing out-of-domain language model performance from few examples. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. 2023. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. *ArXiv*.

Shouxing Xiang, Xi Ye, Jiazhi Xia, Jing Wu, Yang Chen, and Shixia Liu. 2019. Interactive correction of mislabeled training data. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 57–68.

Weikai Yang, Xi Ye, Xingxing Zhang, Lanxi Xiao, Jiazhi Xia, Zhongyuan Wang, Jun Zhu, Hanspeter Pfister, and Shixia Liu. 2022. Diagnosing ensemble few-shot classifiers. *IEEE Transactions on Visualization and Computer Graphics*, 28(9):3292–3306.

Xi Ye and Greg Durrett. 2022a. Can Explanations Be Useful for Calibrating Black Box Models? In *Proceedings of the Annual Meeting of the Association for Computation Linguistics (ACL)*.

Xi Ye and Greg Durrett. 2022b. The (Un)reliability of Explanations in Few-shot Prompting for Textual Reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Xi Ye and Greg Durrett. 2023. Explanation Selection using Unlabeled Data for Effective Chain-of-Thought Prompting. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2020a. Benchmarking multimodal regex synthesis with complex structures. In *Proceedings of the Annual Meeting of the Association for Computation Linguistics (ACL)*.

Xi Ye, Qiaochu Chen, Xinyu Wang, Isil Dillig, and Greg Durrett. 2020b. Sketch-Driven Regular Expression Generation from Natural Language and Examples. In *Transactions of the Association for Computational Linguistics (TACL)*.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2021a. Optimal Neural Program Synthesis from Multimodal Specifications. In *Findings of the Conference on Empirical Methods for Natural Language Processing (Findings of EMNLP)*.

Xi Ye, Rohan Nair, and Greg Durrett. 2021b. Connecting Attributions and QA Model Behavior on Realistic Counterfactuals. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. 2022. RNG-KBQA: Generation augmented iterative ranking for knowledge base question answering. In *Proceedings of the Annual Meeting of the Association for Computation Linguistics (ACL)*.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2023a. SatLM: Satisfiability-Aided Language Models Using Declarative Prompting. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Ves Stoyanov, Greg Durrett, and Ramakanth Pasunuru. 2023b. Complementary explanations for effective in-context learning. In *Findings of the Annual Meeting of the As- sociation for Computational Linguistics (ACL Findings)*.