Scaffolding Targeted Generalization in Natural Language Processing

Ritam Dutt

February 2026

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Carolyn Penstein Rose (Carnegie Mellon University)
David Mortensen (Carnegie Mellon University)
Daniel Fried (Carnegie Mellon University)
Dan Roth (University of Pennsylvania, Oracle AI Labs)

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

Abstract

The holy grail of NLP is generalized success, though that has frequently meant gauging success primarily on the most frequent language phenomena and high-resource tasks. The rise of massive model architectures with billions of parameters, pretrained on trillions of tokens and instruction-tuned on thousands of tasks, has made the holy grail finally seem within reach for a broad range of tasks and domains. However, current systems (especially LLMs), designed in a task- and domain-agnostic manner, still cannot capitalize on the specific characteristics of target tasks and domains.

For many practitioners, small, bespoke systems remain advantageous from a cost-benefit perspective. However, within the core research community, SOTA large-scale models, often designed in a sequence-to-sequence paradigm, remain at the center stage. At the same time, they have a few shortcomings: the computational cost across multiple fronts is becoming formidable, and the transformation from raw data to input embeddings ignores key structures in the data that might be necessary to establish a solid basis for generalization. The result is that crutches are learned from massive data stores that enable high performance within the training distribution but fail to adapt beyond those frontiers.

This dissertation proposes a generalized framework to address both the computational expense problem and the learned crutch problem, leveraging different scaffolds to capture regularities between the source and target. These scaffolds are task-dependent and introduce inductive biases to facilitate generalization. We explore two main kinds of scaffolds: formal and informal. Formal scaffolds ground the information in text in an ontological structure, such as knowledge bases or linguistic frameworks. Informal scaffolds, on the other hand, expand upon the static text and provide an insight beyond what is explicitly mentioned.

In the first part of our thesis, we inspect the role of formal scaffolds for information extraction and question answering over knowledge bases (KBQA). We use linguistic frameworks such as dependency parses and abstract meaning representations for identifying the relation between different entity mentions. While these frameworks can bridge the gap between different domains (such as cooking recipes and materials science corpora) and provide substantial gains in a few-shot setting, the improvements are less pronounced in cross-lingual transfer. For KBQA, we explore how the internal graph structure and path information between source nodes and answers can provide additional support when generalizing to new graphs with unseen entities. We further characterize the reasoning complexity of a KBQA question using isomorphisms and investigate their role as both diagnostic tools and scaffolds to facilitate zero-shot generalization. This broadens the scope of generalization in KBQA to unseen classes and relations rather than just new entities.

In the second part of the dissertation, we investigate the utility of LLM-generated rationales to verbalize social cues implicit in a conversation. We observe that rationales reflecting the speakers' intentions serve as an excellent augmentation, significantly improving performance on two social meaning detection tasks in both

in-domain and cross-domain settings. Our follow-up work broadens the scope of this study along two axes: we explore rationales that capture multiple perspectives and their subsequent impact on cross-task generalization.

Finally, we design a comprehensive evaluation suite to assess whether models exhibit consistent generalization across scenarios, including domains, adversarial settings, and new compositions. In a nutshell, we propose a more holistic evaluation of the generalization capabilities of models.

Acknowledgments

To family, friends, and well-wishers.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Generalization in the context of NLP

"What does it mean for an NLP or AI system to generalize?"

Traditionally, in the context of machine learning, generalization is defined as the ability of models to adapt to new instances that were not observed during training. (Zhang et al., 2016; Freiesleben and Grote, 2023; Mohri et al., 2018). A closer inspection of the generalization definition assumes that all instances are sampled independently from an identical distribution or i.i.d. However, such a simplistic perspective of generalization is ill-suited for the field of language technologies, wherein the term has a much broader meaning.

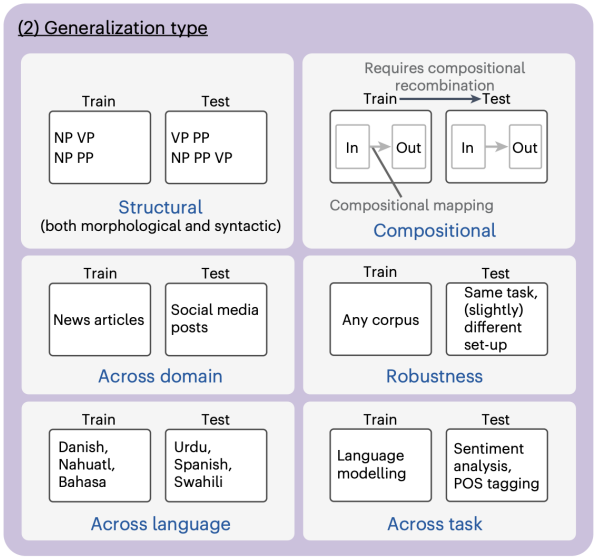


Figure 1.1: Different generalization categories in language technologies

A recent study of Hupkes et al. (2023) provides a broad taxonomic categorization of generalization research in NLP. They highlight how "generalization" refers to a broad umbrella term spanning multiple categories. Some prominent dimensions include generalizing to new

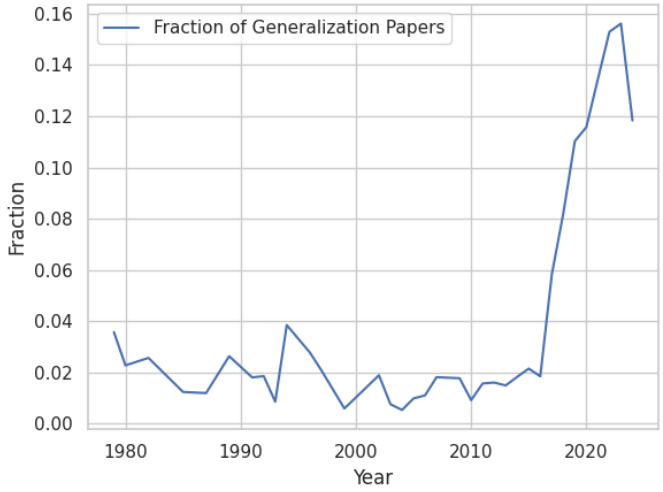


Figure 1.2: Fraction of generalization papers in *CL conferences annually.

tasks (transfer learning), new languages (cross-lingual transfer), new domains (domain adaptation), adversarial examples (robustness), and even new compositions. We illustrate the different generalization categories commonly used in language technologies in Figure 1.1.

Thus, the ability to *generalize* well is a primary desideratum of AI and NLP systems. We have observed tremendous progress in the adoption of language technologies in recent years, with a increased interest in generalization research (Naik et al., 2022). Figure 1.2 illustrates the fraction of papers (published in major *CL conferences annually) that allude to generalization in their abstract or title. [1] We see a sharp increase around 2018–19 with the trend having persisted since.

We observe a gradual shift in the distribution of papers corresponding to different generalization categories (Hupkes et al., 2023) in Figure 1.3, with greater emphasis on task-generalization in recent years compared to domain adaptation or robustness. We dive deeper to inspect the factors that have facilitated this upward trend.

1.2 Reasons behind Task-Agnostic Generalization Trend

The broadening of generalization research can be attributed in part to the rise and rise of foundation models spanning pre-trained LMs like BERT, GPT2, and T5 (Devlin et al., 2019a; Radford et al., 2019) to recent LLMs like ChatGPT (Achiam et al., 2023) and LLama (Touvron et al., 2023). The inherent power of foundation models led researchers to recast several NLP tasks as a sequence-to-sequence paradigm where both the input and output are treated as text sequences. A single unified model was subsequently trained on several NLP tasks; and not only achieved competitive

[1]We use keywords corresponding to “generalization”, “generalize”, “generalisation”, “generalise”, “domain adaptation”, “transfer learning”, “out of domain”, “out-of-domain”, “OOD”, ”task adaptation”, “task transfer”, “task generalization”, “adapt to new tasks”, “cross-lingual generalization”, “cross-lingual transfer”, “cross lingual generalization”, “zero-shot learning”, and “few-shot learning” for our analysis.

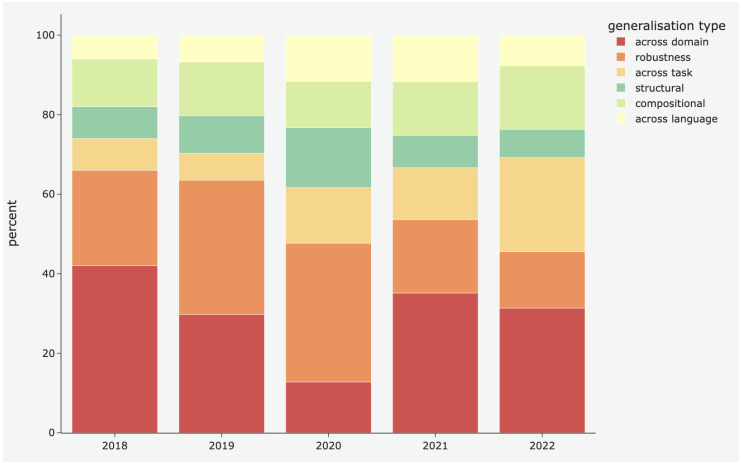


Figure 1.3: Distribution of different generalization categories. (Hupkes et al., 2023).

performance on seen tasks, but also demonstrated impressive instruction following capabilities on unseen tasks (Wang et al., 2023b, 2022c; Chung et al., 2022).

Nevertheless, these unified models generally fare worse than smaller stand-alone counterparts designed for a specific task or goal. It is partly because these models end up ignoring the structure inherent in the data that is crucial for solving the task at hand. For example prior work has observed improved performance on information extraction by incorporating AMRs for both generation (Hsu et al., 2023) and extractive tasks (Zhang et al., 2021a). Likewise Chen et al. (2024) observed the importance of capturing the permutation invariances and other structural properties for efficient reasoning over tabular data.

Additionally, another drawback of these unified seq2seq models is their susceptibility to variation in representations (learned from the text) (Madaan, 2024) especially for unobserved domains and tasks. Prior work has highlighted the poor evaluation performance of ChatGPT on tasks that require reasoning over structured knowledge (Zhuang et al., 2024). Consequently, when models are evaluated on a new distribution, the differences in the text representations are treated as natural representations of the domain and as challenges that should be overcome during training. We illustrate this phenomena pictorially how the trained model needs to adapt to changes in distribution of the input representation arising from difference in domains in Figure 1.4.

Finally, the biggest consequence of these unified models is that they still require thousands of instructions for fine-tuning to reach capabilities of their supervised standalone counterparts (Wang et al., 2022c; Zhuang et al., 2024). These models are also more computationally expensive and resource intensive even for inference and thus limits their accessibility for smaller use-cases.

In this dissertation, we emphasize the importance of enriching language models with information that can bridge the gap between the text and representations corresponding to different tasks. Prior knowledge of the particulars of the task can helps us understand apriori the deficiencies of the textual representations and thus guide our choice of information to enrich these models. We use an umbrella-term "scaffolds" to refer to this task-specific information and the rest of this dissertation helps unpack what, where, how, and why scaffolds can facilitate generalization.

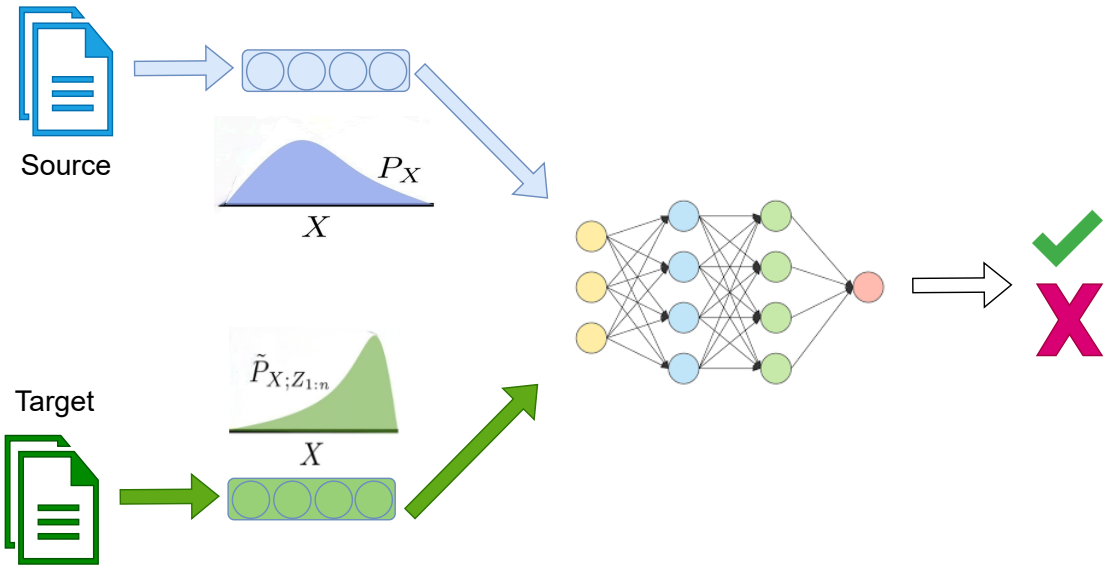


Figure 1.4: A pictorial depiction of the generalization framework in NLP. We use the source data (in blue) to train a system (say a neural network) for a task (say text classification). This trained model is then expected to adapt to an unseen target distribution (as shown in green). The emphasis is that when domains differ, so do the representations that are provided to the model as inputs, and so the model needs to adapt to these changes.

1.3 Scaffolds

The term scaffolds have been used previously in the context of NLP in the form of syntactic parses (Swayamdipta et al., 2018), linguistic frameworks (Gururaja et al., 2023a; Zhong et al., 2020), and document structure (Cohan et al., 2019), which when added to the baseline model improved downstream task performance.

We, however, use "scaffolds" as an umbrella term to refer to constructs that not only compensates for the deficiencies in the textual representations of the input but also bridges the differences between the source and the target. Informally, these scaffolds, can enable the model to operate at a higher level of abstraction to facilitate generalization across different but related tasks.

As opposed to the more prevalent task-agnostic paradigm where models are fine-tuned in a supervised setting over thousands of tasks (both natural or synthetic) to infuse them with instruction-following capabilities (Wang et al., 2022c, 2023b), we adopt a more task-aligned approach where the underlying scaffolds introduces a common schema that tasks can build upon, thereby facilitating transfer.

Figure 1.5 illustrates how scaffolds (shown in grey) augment the representations coming from the source and target (in blue and green respectively). Scaffolds are designed to capture a theory (or inductive bias) unarticulated in the text which when supplemented with the model is crucial for generalizing across tasks or distributions. In our work, we make the distinction between two main categories of scaffolds: formal and informal.

Formal scaffolds refer to architectural structures or representational frameworks that ground or supplement the information obtained from the text in some formal structure. These structures

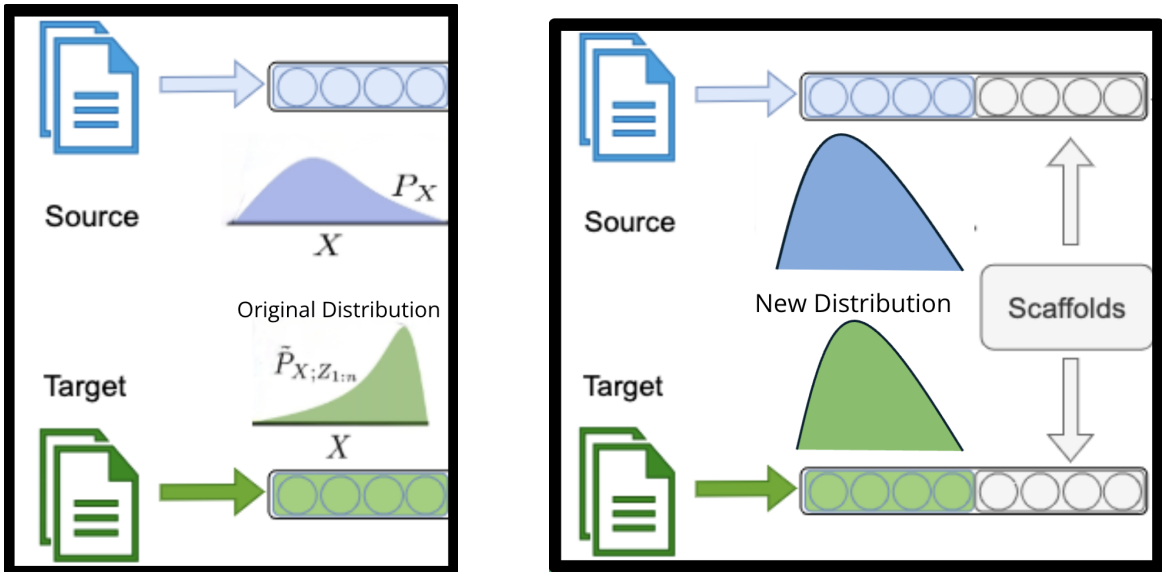


Figure 1.5: We present pictorially a simple illustration of scenarios where the target distribution differs considerably from the source. By injecting scaffolds, that explicitly highlights out the commonalities between the source and the target domain, the representations become more similar to one another thereby facilitating generalization. The old original distribution appears in the left and the new updated distribution is displayed on the right.

can either be external sources of ontological or encyclopedic knowledge like knowledge graphs, databases, or tables, or formalization of the internal structure of the text in the form of linguistic frameworks like syntactic and semantic parses. Formal scaffolds represent the most prevalent method of inducing structured knowledge into a model , usually in the form of graph (Wu et al., 2021), linearized data (Zhuang et al., 2024), or other like memory (Khandelwal et al., 2019).

Informal scaffolds, on the other hand, augment information to the system in the form of free text. These scaffolds aim to either supplement the model with additional world knowledge that is absent from the text or make explicit the subtle (or implicit) information present within the static text. In a way, these scaffolds verbalize the knowledge encoded in the parameters of an external system or even the same system. For example, rationales, explanations, chain-of-thought reasoning all fall under this broad terminology of informal scaffolds. (Majumder et al., 2022; Wiegreffe et al., 2021; Wei et al., 2022c).

Task	Scaffolds
Relation Extraction	Linguistic Frameworks
Question Answering over Knowledge Bases	Isomorphisms or Reasoning Patterns
Social Meaning Detection	Social Rationales

Table 1.1: Different Tasks and the scaffolds we explore for each task.

1.4 Thesis Outline

We present an overview of the different kinds of formal and informal scaffolds that can facilitate NLP generalization and organize them in the corresponding chapters of the dissertation as follows.

- Chapter 2 provides a literature review of the utility of different formal and informal scaffolds for several NLP tasks. Specifically, we touch upon how past work has leveraged different scaffolds for facilitating generalization and how we carry the baton forwards in investigating the cases or situations where these scaffolds are helpful. As a result, we ground the utility of formal scaffolds in the three tasks of interest in this thesis, i.e. relation extraction, question answering over knowledge graphs, and semantic parsing over knowledge bases. Likewise, we motivate the effectiveness of informal scaffolds for dialogue generalization tasks specifically those pertaining to social understanding. Finally, we explore in-depth the current generalization research landscape in NLP and how our work addresses a current under-represented area in this space, i.e. a holistic evaluation of generalization research.
- Chapter 3 defines formal scaffolds, i.e. structured frameworks that anchor text to formal representations either via an external ontology (e.g. knowledge graphs) or internal linguistic structures (syntactic and semantic parses). We operationalize formal scaffolds for three concrete tasks, i.e. relation extraction (RE), question answering over personalized knowledge graphs (PERKGQA) and isomorphism guided KBQA. For each task, we note the kind of generalization that is being investigated (i.e. cross-domain, cross-lingual, zero-shot schema generalization) and motivate how the particular scaffold can facilitate generalization.
- Chapter 4 explores the role of linguistic frameworks as formal scaffolds for few-shot relation extraction. We inspect generalization in two settings, i.e. generalization across domains for procedural text, and generalization across multiple languages. For the cross-domain experiments, the choice of our scaffolds were inspired by the observation that while different corpora for procedural text exhibit stark lexical and topical variations, they are united in the common theme of carrying out a set of step-wise instructions for a particular goal and thus exhibit similar semantics owing to their procedural nature. Our multilingual relation extraction experiments provides a contrast and enables us to investigate whether linguistic frameworks afford the same utility when the transfer takes place across languages with the domain being invariant. We augment popular transformer based models with linguistic frameworks using graph-aware neural architectures for both tasks.
- Chapter 5 introduces the task of question answering over knowledge graphs (KG) in a personalized setting. Unlike traditional KGQA, where there exists a single KG that is known apriori and is leveraged similarly across all queries, in PERKGQA a query is accompanied by a KG specific to the user. Consequently KGQA systems need to adapt to unseen queries over new entities for users, without having any access to prior information of the user. We demonstrate in this chapter how one can use the internal structure of the knowledge graph and the path information as scaffolds to facilitate generalization to these new settings over different cases.
- Chapter 6 establishes the dual-role of isomorphisms for the task of question answering over knowledge bases (KBQA). Firstly, we use isomorphisms to analyze the strengths and pitfalls of different KBQA systems such as whether a model is better equipped to handle questions with

longer hops or more constraints. Secondly, we leverage isomorphisms to improve KBQA performance of LLMs. We propose ISOCOD a framework for predicting the isomorphism category for a given KBQA query, by unifying text and graph representations of the constituent knowledge store. We can incorporate this isomorphism prediction module into a light-weight verification-and-regeneration loop. Using isomorphisms as scaffolds yields consistent improvements over the simple LLM baseline, with predicted isomorphisms delivering realistic gains.

- Chapter 7 scopes out the definition of informal scaffolds in the context of "rationales" i.e. free-form textual explanations designed to capture subtle pragmatic cues in conversations. We propose social meaning detection as a valid stress-test for generalization in dialogue since social meaning is subtly encoded, and models that rely on surface cues are prone to over-fitting while dealing with such tasks. We further outline two complementary generalization settings (cross-domain and cross-task) and realize them using different datasets. We thus lay the foundation of informal scaffolds or rationales in aiding generalization pertaining to social understanding tasks.
- Chapter 8 explores the ability of machine-generated rationales to capture the implicit meaning encoded in a social conversation. We subsequently investigate the capabilities of these rationales to facilitate cross-domain generalization for two social meaning detection tasks, i.e. identifying resisting strategies (RES) and recognizing emotions in conversations (ERC). We propose a prompting framework grounded in socio-linguistics theory to generate rationales that capture a user's intentions, assumptions, and biases. We carry out an extensive study over different models, few-shot settings, and tasks and observe significant gains from incorporating these rationales for both in domain and across domains, thus validating our hypothesis.
- Chapter 9 builds upon the past findings to investigate the capabilities of rationales in facilitating generalization across different social meaning understanding tasks. We introduce an automated framework SOCIAL SCAFFOLDS to generate task-agnostic social rationales that are capable of capturing perspectives corresponding to different points of view in narrative modeling. Our comprehensive evaluation suite demonstrate that rationales significantly improves task transfer performance for both fine-tuning and in-context learning experimental setups 31.3% and 44.0% of the times respectively. Our findings shows the promise of pragmatics-oriented data augmentation for social understanding and generalization.
- Chapter 10 presents a systematic study on the multi-faceted generalization abilities (i.e. domain adaptation, robustness, and compositional generalization) of common transformer-based models for two NLP tasks. Our findings reveal that generalizability is intrinsic to a model instance, i.e. a model instance typically does not generalize well in one dimension and poorly in others. Moreover, generalizability is well correlated with model size, with certain training strategies and model architectures achieving better generalization performance than others. In a nutshell, we propose a more holistic evaluation of generalization in NLP.
- Chapter 11 serves two purposes. Firstly, we highlight and outline the important contributions put forward in this dissertation. Secondly, we discuss potential avenues for future work, i.e. namely how personalization acts as an extension to generalization and how one can integrate structured knowledge within LLMs effectively.

1.5 Contributions

This dissertation has resulted in the following publications in *CL conferences.

- Dutt, R., Wu, Z. Ling, D., Gu, Y., & Rosé, C. "Leveraging Isomorphisms to facilitate Zero-Shot Generalization for Question Answering over Knowledge Bases." [Under submission]
- Wu, Z., Dutt, R., Breitfeller, L.M., Nourbakhsh, A., Parekh, S. and Rosé, C., 2025. "R^2-CoD: Understanding Text-Graph Complementarity in Relational Reasoning via Knowledge Co-Distillation." In Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers).
- Dutt, R., Rose, C. and Sap, M., 2025, November. "SOCIAL SCAFFOLDS: A Generalization Framework for Social Understanding Tasks." In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (pp. 29149-29185).
- Dutt, R., Sural, S. and Rose, C., 2025, May. "Can dependency parses facilitate generalization in language models? A case study of cross-lingual relation extraction." In Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing (pp. 338-358).
- Dutt, R., Choudhury, S.R., Rao, V.V., Rose, C. and Vydiswaran, V.V., 2024, November. Investigating the generalizability of pretrained language models across multiple dimensions: A case study of NLI and MRC. In Proceedings of the 2nd GenBench Workshop on Generalisation (Benchmarking) in NLP (pp. 165-182).
- Dutt, R., Wu, Z., Shi, J., Sheth, D., Gupta, P. and Rose, C., 2024, August. Leveraging machine-generated rationales to facilitate social meaning detection in conversations. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 6901-6929).
- Dutt, R., Khosla, S., Kumar, V.B. and Gangadharaiah, R., 2023, November. GrailQA++: A challenging zero-shot benchmark for knowledge base question answering. In Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 897-909).
- Khosla, S., Dutt, R., Kumar, V.B. and Gangadharaiah, R., 2023, May. Exploring the reasons for non-generalizability of KBQA systems. In Proceedings of the Fourth Workshop on Insights from Negative Results in NLP (pp. 88-93).
- Gururaja, S., Dutt, R., Liao, T. and Rose, C., 2023, July. Linguistic representations for fewer-shot relation extraction across domains. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 7502-7514).
- Dutt, R., Bhattacharjee, K., Gangadharaiah, R., Roth, D. and Rose, C., 2022, July. PerKGQA: Question answering over personalized knowledge graphs. In Findings of the Association for Computational Linguistics: NAACL 2022 (pp. 253-268).
- Dutt, R., Sinha, S., Joshi, R., Chakraborty, S.S., Riggs, M., Yan, X., Bao, H. and Rose, C., 2021, April. Resper: Computationally modelling resisting strategies in persuasive conversations. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (pp. 78-90).

Chapter 2

Literature Review

We situate the thesis in prior work of infusing scaffolds for different NLP tasks. We then do a deep dive into the specific tasks that the scaffolds play a role in, specifically relation extraction and question answering over knowledge bases for formal scaffolds, and understanding social meaning using informal scaffolds. We also revisit the current space of generalization research in NLP.

2.1 Formal Scaffolds

Formal scaffolds are architectures or representations that incorporate information from a pre-defined structured framework. The frameworks in question can either be grounded in external ontological knowledge sources like databases and graphs, or can be based on the internal structure of the text in the form of semantic or syntactic parses. The term "formal scaffolds" is closely associated with the broad idea of structure in language technologies, and hence we use these terms interchangeably while referring to past work in this space.

The role of structure in facilitating NLP development and progress in indisputable; language itself has an inherent structure (Cheng et al., 2016). Specifically, the way information is packaged is not an arbitrary phenomena (Croft, 2022). While the kind of structures employed, and the way they have been infused in NLP systems have transitioned over the years, their usage remains ubiquitous. Here, we discuss varied forms of structure such as syntactic and dependency parses, semantic parses, and ontological knowledge, memory, the structure inherent in the task, and neurosymbolic approaches.

Syntactic structures, conceptualized by Chomsky, formalize the arrangement of words in natural language text. These structures, mostly in the form of constituent and dependency parses, have been used extensively for a wide variety of NLP tasks such as language modelling (Du et al., 2020; Shen et al., 2021), machine translation (Post and Gildea, 2008; Bastings et al., 2017; Chen et al., 2017; Egea Gómez et al., 2021), information extraction (Grishman, 1996; Vashishth et al., 2018; Duan et al., 2022) that includes relation extraction (Tian et al., 2022b; Gururaja et al., 2023b), and semantic role labelling (Cai and Lapata, 2019; Sachan et al., 2021; Kasai et al., 2019) amongst others.

In a complementary sense, semantic structures in the form of frame-semantics or abstract meaning representations have also been explored for several NLP tasks. Past work has investigated

the role of infusing semantic structures in for language modelling (Prange et al., 2022; Vashishth et al., 2019), information extraction (Bassignana et al., 2023; Hsu et al., 2023; Zhang and Ji, 2021), and other NLU tasks (Wu et al., 2021; Guan et al., 2021; Ma et al., 2023).

NLP systems also require access to world or ontological knowledge to ensure generalization to domains beyond which the models have been trained. Consequently, incorporating external knowledge in the form of knowledge bases, data stores, and tables has also been explored for NLP tasks including language modelling (Zhang et al., 2019; Wang et al., 2021b), factual checking (Feng et al., 2023), and open-domain QA (Han et al., 2020; Yu et al., 2022b). Infusing such external knowledge has also demonstrated utility across domains like biomedical (Meng et al., 2021; Hao et al., 2020) and finance (Nararatwong et al., 2022). The source of knowledge usually comes in the form of generic knowledge bases like Freebase (Bollacker et al., 2008) and WikiData (Vrandečić and Krötzsch, 2014), task-specific knowledge sources like ConceptNet (Speer et al., 2017), NormBank (Ziems et al., 2023), and SOCIAL-CHEM-101 (Forbes et al., 2020) for social and commonsense reasoning, or domain-specific knowledge banks like UMLS for medicine (Bodenreider, 2004).

Beyond these linguistic frameworks or external knowledge stores, the notion of structure often spans to encompass the structure inherent in the task such as hierarchical structure for extractive text summarization (Ruan et al., 2022), document structure for document-level tasks like QA and NLI (Buchmann et al., 2024), discourse structure for document modelling (Koto et al., 2019), and dialogue structure for conversational tasks (Jiao et al., 2019; Ghosal et al., 2019).

Likewise there has also been a renewed interest in leveraging the internal memory of architectures for NLP applications. Some recent strides involve using memory for improving feedback (Madaan et al., 2022a; Tandon et al., 2022), or memory as a knowledge store to retrieved previously seen cases (Das et al., 2021a; Khandelwal et al., 2019; Sarch et al., 2023) or using memory as a parametric architectural component during training and inference (Jiao et al., 2020; Wang et al., 2020a; Jain and Lapata, 2021).

Finally, in the current era of massive LLMs, the role of structure has broadened to accommodate executable programs and codes, drawing parallels to prior work in neuro-symbolic processing. Some recent advancements in this field include converting natural language into a proto-language to extract polarity from text (Cambria et al., 2022), or prompting LLMs to generate code for procedural reasoning (Madaan et al., 2022b), narrative understanding (Dong et al., 2023), and logical reasoning (Olausson et al., 2023).

We now deep dive into how formal scaffolds have been employed in two NLP applications, i.e. relation extraction (RE) and question answering over knowledge bases (KBQA).

2.1.1 Relation Extraction

The goal of relation extraction (RE) is to detect and classify the relation between specified entities in a text according to some predefined schema. Current research in RE has mostly been carried out in a few-shot or a zero-shot setting to address the dearth of training data (Liu et al., 2022b) and the "long-tail" problem of skewness in relation classes. (Ye and Ling, 2019b). Salient work in that direction includes (i) designing RE-specific pretraining objectives for learning better representations (Baldini Soares et al., 2019; Zhenzhen et al., 2022; Wang et al., 2022a), (ii) incorporating meta-information such as relation descriptions (Yang et al., 2020; Chen and Li,

2021) a global relation graph, (Qu et al., 2020), or entity types (Peng et al., 2020b), and (iii) leveraging additional information in the form of dependency parses (Yu et al., 2022c), translated texts for multilingual RE (Nag et al., 2021), or distantly supervised instances (Zhao et al., 2021; Ye and Ling, 2019a).

Recent works such as Prange et al. (2022) have demonstrated the significant potential of using human-annotated linguistic information as scaffolding for learning language models. Other works such as Zhang and Ji (2021) and Bai et al. (2021) use automatically generated semantic annotations. These works depend on the idea that the structure that the linguistic frameworks provide allows models to better learn salient features of the input.

Supplementing training data with explicit linguistic structure, in the form of syntactic and semantic parses has led to substantial improvements in the in-domain performance on several NLP tasks. Sachan et al. (2021) challenges the utility of syntax trees over pre-trained transformers for IE and observed that one can only obtain meaningful gains with gold parses. Semantic parses, in the form of AMRs, have shown to be beneficial for IE (Zhang et al., 2021b; Zhang and Ji, 2021; Xu et al., 2022), even when the parses employed are not human-annotated. In this work, we raise the question of the utility of either kind of parse for few-shot RE in a cross-domain setting.

Recently, LLMs have shown promise in zero-shot relation extraction. Challenging cases such as overlapping relations and none-of-the-above (nota) relations have been handled effectively by LLMs in zero-shot settings (Li et al., 2023). LLMs have also outperformed smaller models for RE with larger, document-level context sizes in models such as AutoRE (Xue et al., 2024). All of these techniques seek to alleviate the need of using expensive human-annotated training data, which is where linguistic frameworks come into the picture.

2.1.2 Question Answering over Knowledge Graphs (KGQA)

Question Answering (Q&A) requires gathering and aggregating information from various domain-specific knowledge sources, including unstructured or textual content (e.g. Wikipedia articles or FAQ pages) or structured sources such as ontologies, database tables, or knowledge graphs. The aim of KGQA is to answer natural language question by querying over a pre-defined knowledge graph. The task has evolved from a simple-classification setting (Mohammed et al., 2018) to an information retrieval paradigm (Wang et al., 2020c; Saxena et al., 2020; Yasunaga et al., 2021; Sun et al., 2019; Xiong et al., 2019).

Progress in KGQA research has addressed several challenges, such as answering complex questions, multi-hop reasoning, (Lan and Jiang, 2020; Ren et al., 2021), conversational KGQA (Dieleman et al., 2015), and multi-lingual KGQA (Zhou et al., 2021), and has also found applications in tax, insurance, and healthcare (Lüdemann et al., 2020; Huang et al., 2021; Park et al., 2020).

While other approaches such as semantic parsing (Lan and Jiang, 2020; Ding et al., 2019; Maheshwari et al., 2019; Zhu et al., 2020; Ren et al., 2021) and reinforcement learning (Das et al., 2018; Lin et al., 2018; Saha et al., 2019; Ansari et al., 2019) are popular alternatives, graph-based information retrieval methods are preferred for cases where the underlying ground-truth semantic parses are unavailable. For example, work on KGQA generalizability such as that of Gu et al. (2021), and Chen et al. (2021b) requires logical forms (s-expressions or SPARQL queries) during training; information often unavailable in real-world.

Most KGQA approaches that operate in an information retrieval setting over predefined knowledge graphs follow a similar procedure to make the problem computationally feasible. (Sun et al., 2018, 2019; Wang et al., 2020c,b). They first construct a smaller sub-graph for each question from the base graph, using the Personalized PageRank algorithm (Haveliwala, 2003), and re-use the base graph's node representation to initialize the nodes in the sub-graph. Thus during inference, the model has prior knowledge of the nodes. However, in such a setting, if the system encounters a new KG during inference, they would need to learn the representations of those unseen nodes from scratch, a challenge we aim to solve in Chapter 5.

2.1.3 Question Answering over Knowledge Bases (KBQA)

While we use the term "knowledge graphs" and "knowledge bases" interchangeably in this thesis, the main distinction between the two is that knowledge bases allow for more varied ways of storing and representing information rather than the entity-relationship (ER) model in knowledge graphs. Knowledge bases arranges information in the form of a schema and hence allows greater flexibility in representing information. A more detailed description can be found in (Gu et al., 2022a). Here we present an overview of the task formulation of KBQA in formal notation and the different kinds of KBQA generalization. Unlike the previous section, we exclusively refer to the semantic-parsing based approaches for KBQA.

Semantic-parsing based KBQA: Given the KB, $\mathcal{K}$, and a natural language question q, the objective of KBQA is to find a set of entities (answers $\mathcal{A}$) that satisfies the question q. In a semantic-parsing or translation based setting, the task of KBQA involves converting q into its corresponding logical form L_q. This L_q is executed over the $\mathcal{K}$ to obtain the answers. Examples of logical forms include S-expressions (Gu et al., 2021), SPARQL queries (Saha et al., 2019), and λ-calculus.

Each logical form L_q has a particular schema $\mathcal{S}_q$ that includes elements from the set of relations, classes, and other constructs specific to the logical-form. The specific composition of items in $\mathcal{S}_q$ forms a logical template or $\mathcal{T}_q$. E.g., the questions "Who wrote Pride and Prejudice?" and "Who was the author of Oliver Twist?" have the same template but different logical forms since they refer to different novels. However the questions "Who wrote Pride and Prejudice?" and "Which author wrote both the Talisman and It?" have the same schema but different logical templates since the former involves only one constraint or entity ("Pride and Prejudice") while the latter specifies two ("Talisman" and "It"),

Case Based Reasoning (CBR) in KBQA: Similar to semantic parsing based methods, case-based reasoning or CBR is a popular way of KBQA, wherein one combines or composes individual queries to form the final answer. For example, one of the first works of CBR is that of Das et al. (2020), which performs relation linking such as (Delhi, capital_of, _?_). They first retrieve entities similar to the query entity and the corresponding reasoning paths that lead to an answer for those retrieved entities. Finally, then apply reasoning paths to the query entity. In a similar vein, Das et al. (2021b) uses a neuro-symbolic case-based reasoning approach for answering complex, multi-hop questions.

KBQA Generalization

Gu et al. (2021) puts forward the three levels of generalization based on how the schema $\mathcal{S}_q$ and logical template $\mathcal{T}_q$ for a question q differs from the set of all possible schema items and templates seen during training, i.e. $\mathcal{S}_{train}$ and $\mathcal{T}_{train}$ respectively. These include.

(i) **I.I.D.** generalization occurs when $\mathcal{S}_q \subset \mathcal{S}_{train}$ and $\mathcal{T}_q \in \mathcal{T}_{train}$.

(ii) **Compositional** generalization occurs when $\mathcal{S}_q \subset \mathcal{S}_{train}$ but $\mathcal{T}_q \notin \mathcal{T}_{train}$. Thus the questions operate upon a subset of schema items seen during training but they have new templates.

(iii) **Zero Shot** generalization occurs when $\exists s \in \mathcal{S}_q$ such that $s \notin \mathcal{S}_{train}$. Thus the questions operate upon novel schemas, mostly new classes and relations that were not encountered during training.

Conceptually, these three levels of generalization could be stacked in an hierarchical fashion in increasing order of difficulty; with I.I.D. being the least challenging since it operates over templates seen during training, followed by Compositional, which occurs over unseen templates, and then Zero Shot which have unseen schema items. In Chapter 6 and **??**, we exclusively focus on the task of Zero Shot generalization in KBQA.

2.2 Informal Scaffolds

As opposed to formal scaffolds which capture information in a formal structure, we define informal scaffolds as constructs that augment information to the system in the form of free text. Based on the origin of these free texts, these scaffolds aim to bolster the current information encoded in the text with additional contextual knowledge akin to retrieval augmented systems (Lewis et al., 2020), or verbalize the information encoded in the language model's internal parameters, akin to current work on Theory of Mind (Sap et al., 2022), and Chain of Thought Reasoning (Wei et al., 2022c). In a way, we can distinguish these two situations based on whether the source of information was "external" or "internal" to the model, i.e., whether the information source is non-parametric or parametric respectively.

Providing textual information to NLP systems grounded in external knowledge stores has seen promise on several tasks. These approaches range from extracting free-text explanations from an external knowledge base for NLI and commonsense QA (Schuff et al., 2021; Majumder et al., 2022; Chen et al., 2020, 2021a; Ghosal et al., 2023), to retrieving relevant text for open-domain QA (Lewis et al., 2020; Izacard and Grave, 2021), dialogue understanding (Feng et al., 2021), reading comprehension (Thai et al., 2023), summarization (Cao et al., 2018), and translation (Khandelwal et al., 2020), to querying a larger LLM for relevant information that is subsequently used for model distillation (Hsieh et al., 2023).

In a complementary light, large-scale pretraining enables language models to encode information about the world in their parameters (Petroni et al., 2019; Zhang et al.; Wang et al., 2021a; Singhal et al., 2023). Consequently, as an alternate to retrieval or extraction, prior work investigates how one can generate relevant information which serves as an auxiliary input for downstream tasks. A commonplace example is to generate natural language "rationales" or "explanations" for NLU tasks like entailment or reasoning (Wiegreffe et al., 2021; Kumar and Talukdar, 2020; Rajani et al., 2019). Another line of research that has sparked wide interest is

the ability of LLMs to self-rationalize about the task in a zero-shot or few-shot setting through specific prompt designs such as chain-of-thought prompting (Wei et al., 2022b; Zelikman et al., 2022; Zhou et al., 2023a), or using the generated outputs to refine their predictions (Madaan et al., 2024), to align themselves to unseen tasks (Wang et al., 2023c), or improve the faithfulness and consistency of their outputs (Huang et al., 2023; Wang et al., 2022b)

We thus see several use-cases of informal scaffolds such as facilitating commonsense and social reasoning (Zelikman et al., 2022; Majumder et al., 2022), explaining the predictions of neural models (Wiegreffe et al., 2021; Jayaram and Allaway, 2021; Zaidan et al., 2007), assisting humans in their tasks (Das and Chernova, 2020; Joshi et al., 2023), and even contributing to OOD generalization of models(Majumder et al., 2022; Xiong et al., 2023; Joshi et al., 2022).

Building upon this foundation, we approach informal scaffolds or rationales through a under-explored lens: the elicited verbalization of social meaning in a conversation, which makes explicit the underlying social signals and helps overcome some limitations of static text like omission of communicative intent (Sap et al., 2022). We make a distinction from prior works on social reasoning (Rao et al., 2023; Sap et al., 2020a) which uses rationales as means of contextualizing a task with preconceived social norms, whereas we use rationales to elicit the implicit intentions and assumptions of the speaker.

The remainder of this subsection hones into the role of social meaning in NLP and how rationales can help facilitate generalization of social meaning, both across domains and across tasks in Chapters 8 and 9 respectively.

2.2.1 Social Meaning Detection tasks in NLP

According to sociologist Erving Goffman (Goffman, 2002) language conveys two forms of "social meaning", namely, one that is *given* or intentional, and one that is *given off* or unintentional, often thought of as "reading between the lines". The former embodies the idea of linguistic agency, the deliberate choices people make to protect their identity (Gee, 2014) or to accomplish social goals (Martin and Rose, 2003). The latter encompasses involuntary cues which signals their disposition, like mental illness (Kayi et al., 2017; Alqahtani et al., 2022), personality (Mairesse et al., 2006; Moreno et al., 2021), attitude (Martin and White, 2003), or emotion (Hazarika et al., 2018).

Social meaning is thus defined as the signaling people do during interactions to maintain positioning in terms of identity and relationship (e.g., practices of signaling are defined in detail in Gee (2014), with additional operationalizations in Martin and White (2003) and Meyerhoff (2019)). While originally defined in the context of socio-linguistics, the term "social meaning" has been heavily used in the computational linguistics community. It can refer to different ways or styles people interact (Jurafsky et al., 2009), or the social background and identity of a user that can be predicated from linguistic variation (Nguyen et al., 2021a), or the meaning that emerges through human interaction on social media in the form of emotion, sarcasm, irony and the like (Zhang and Abdul-Mageed, 2022).

Given the myraid definitions, we use "social meaning" as an umbrella term to refer to tasks that infer the intentions of the users or their characteristics in a social setting. Some instances of social meaning we explore in our thesis include identifying (i) strategies to resist persuasion (Dutt et al., 2021), (ii) formulations of politeness (Dutt et al., 2020b), (iii) emotions and empathetic exchanges expressed in conversations (Sharma et al., 2020; Busso et al., 2008), (iv) negotiation and

persuasive attempts (Wang et al., 2019b; He et al., 2018b; Chawla et al., 2021), (v) argumentative strategies (Jo et al., 2020), and (vi) implicit forms of hate speech (ElSherief et al., 2021; Breitfeller et al., 2019). We contextualize the tasks of social meaning detection in conversational exchanges and why generalization for dialogue tasks remains particularly challenging.

2.2.2 Generalization in Dialogue

Generalization in the context of dialogue tasks is challenging because interactions are typically structured to accomplish a task rather than simply conveying information. Such a task-centered organization enables participants to rely heavily on implicit cues by omitting information they know to be shared among all participants (Dutt et al., 2024).

Mehri (2022) outlines different types of generalization imperative for dialogue. These include (i) new inputs arising from covariate shift or stylistic variation (Khosla and Gangadharaiah, 2022), (ii) new problems in dialogue modeling such as evaluation and response generation (Peng et al., 2020a), (iii) new outputs and schemas corresponding to out-of-domain shift (Larson et al., 2019) and (iv) new tasks such as controlled generation or fact verification (Gupta et al., 2022).

Politeness is a good example of a social meaning where work on generalizability has been frequent, and in fact, the theory itself was designed with the intention of generalizability (Brown et al., 1987). This particular theory has been operationalized computationally using a wide variety of approaches as the field has evolved (Danescu-Niculescu-Mizil et al., 2013; Li et al., 2020; Dutt et al., 2020a). In practice, generalizability is still challenging (Khan et al., 2023b), because the features that garner the most influence within trained models tend to domain-specific or the relatively infrequent, strongly overt forms of politeness.

Another notable work on transfer for social meaning detection is that of Hazarika et al. (2021) where they designed a hierarchical dialogue model, pretrained on multi-turn conversations and subsequently adapted for emotion classification. Previous work on few-shot generalization in dialogue has benefited from large-scale multitask pretraining (Wu et al., 2020; Peng et al., 2021; Hosseini-Asl et al., 2020) or instruction tuning (Gupta et al., 2022; Wang et al., 2025; Sanh et al.; Wang et al., 2022c). In this thesis, we propose an efficient solution that uses the underlying social cues in a dialogue as augmentations to unify multiple tasks without the need for large-scale pretraining.

2.3 Generalization Research through the lens of NLP

Recent years have witnessed a tremendous interest in generalization research in the language technologies community, with a marked distinction from how generalization research is distinct and complementary from other fields like machine learning or computer vision. In this section, we briefly touch upon the advancements in NLP generalization by drawing on past work and surveys conducted in the field in recent times.

The most seminal work we explore is that of Hupkes et al. (2023) where the authors design a thorough and meticulous taxonomy of generalization research in the context of NLP. The taxonomy, rooted in a extensive literature review, investigates generalization research across five key axes based upon the main motivation for carrying out the research, the type or kind

of generalization that is being investigated, the type of data shift that is characteristic of the generalization, the source of this data shift, and the locus of the shift in the modelling pipeline. We defer the reader to the original paper for a comprehensive overview of each aspect. While this work presents a comprehensive taxonomy of how researchers can summarize their own generalization research, it does not inspect deeper into the categorization of the generalization/adaptation methods.

As a complementary line of work, we present a hierarchy of adaptation methods common in NLP domain. We start off with the unsupervised domain adaptation strategies proposed by (Ramponi and Plank, 2020) which were expanded in a follow-up work of Naik et al. (2022). The updated framework organizes the adaptation methods into (i) model-centric which perform adaptation by modifying the model architecture such as feature augmentation (Blitzer et al., 2006; Daumé III, 2007) or loss augmentation (Ruder, 2017; Zhang et al., 2017), (ii) data-centric which carries out adaptation by leveraging additional labelled or unlabelled data in the source or target domain such as pretraining (Gururangan et al., 2020; Devlin et al., 2019a) or psuedolabeling (Nishida and Matsumoto, 2022; Chen et al., 2021c) and (iii) hybrid strategies which perform a combination of both such as instance weighting (Chen et al., 2021d; Jiang and Zhai, 2007) and data selection (Swayamdipta et al., 2020; Attendu and Corbeil, 2023; van der Wees et al., 2017) and hence is a separate category. The main objective of this work was to highlight how prevalent unsupervised domain adaptation techniques were concentrated for a few specific NLP tasks leaving the long-tail problem unaddressed.

With the prevalence of massive scale pretraining and development of LLMs, there has been a gradual shift in paradigm in moving towards a task-agnostic set-up where several tasks are unified into a single sequence2sequence paradigm. The consequence of scale also led to the popularization of parameter efficient fine-tuning approaches (PEFT) where instead of full-model training, only a subset of it was modified such as adapters, (Houlsby et al., 2019a; Liu et al., 2022a), low-rank adaptation (Hu et al., 2021a; Dettmers et al., 2024), and prefix-tuning (Li and Liang, 2021; Xu et al., 2021). In a similar vein, prompt-tuning or in-context learning approaches where a prompt, is tuned in a zero-shot/ few-shot setting to adapt to the particular task has also gained popularity (Gu et al., 2022b; Hambardzumyan et al., 2021; Vu et al., 2022; Lester et al., 2021; Jin et al., 2022).

There has also been an interest in characterizing the transferability for different NLP tasks in different settings. Recent work in this space investigates how to select better source tasks for multi-task NLP (Kim et al., 2023; Albalak et al., 2022), what intermediate tasks to fine-tune on (Poth et al., 2021; Pruksachatkun et al., 2020), and whether and how such cross-task generalization ability can be acquired (Ye et al., 2021a). There has also been a keen interest in measuring and characterizing domain robustness (Calderon et al., 2023) and how the difference in performance across domains can be attributed to the choice of the target rather than the source. Similarly, for *robustness generalization*, many papers have proposed adversarial attacks to perturb the input to fool the model. These attacks can be white-box (Ebrahimi et al., 2018), i.e., the attacker has access to the model parameters or not (black-box (Jin et al., 2020), see Goyal et al. (2023) for a survey).

We thus emphasize the rich history of generalization research in NLP, and how we aim to carry it forward. In this dissertation we emphasize the importance of task-specific scaffolds and its role in different kinds of generalization in NLP.

Part I

Formal Scaffolds

Chapter 3

Formal Scaffolds: Tasks and Datasets

3.1 Formal Scaffolds: Definition

We define formal scaffolds as structured frameworks that anchor the textual information to a predefined formal representation which can be either ontological sources external to the text (e.g. knowledge bases or taxonomies) or internal linguistic formalisms (e.g., syntax or semantic parses). Subsequently, these frameworks enrich the text with factual information and help in information extraction tasks such as named-entity recognition, question answering, and relation extraction. We explore three main tasks in our thesis pertaining to formal scaffolds. These include relation extraction (RE), information extraction over knowledge graphs (KGQA), and isomorphism prediction for question answering over knowledge bases (KBQA).

3.2 Relation Extraction

The goal of relation extraction (RE) is to detect and classify the relation between specified entities in a text according to some predefined schema. We present a detailed account of past work in this space in Chapter 2. In this section, we specifically discuss the two kinds of relation extraction tasks for which we show the utility of linguistic frameworks as formal scaffolds. These include relation extraction over procedural text and multi-lingual relation extraction.

3.2.1 Relation Extraction over Procedural Text

A procedural text instructs the audience on how to complete a specific task. For example, a simple cooking recipe such as "Gently mix eggs, milk and flour in a bowl. Heat the mixture in an oven" is an example of a valid procedural text. Since the instructions have a clear chronological order in which they need to be carried out, procedural texts are often be represented as narrative flow graphs (Jiang et al., 2020; Yamakata et al., 2020a). We present an example of a narrative flow graph corresponding to the aforementioned cooking instruction below. The events or actions are represented as ovals, the entities are represented as circles or diamonds, while the arrows connecting these events/entities represent the relation between them.

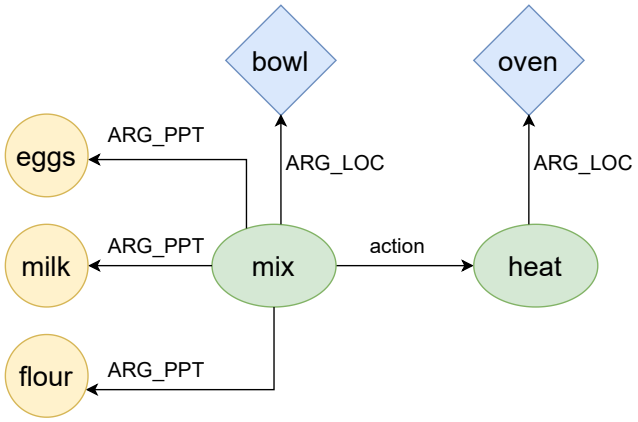


Figure 3.1: An example of a procedural text with the corresponding narrative flow graph. The green ovals (i.e. mix and heat) refer to the events, the entities in yellow circles refer to the participants of the action (i.e. eggs, milk, and flour), while the entities in blue diamonds refer to the location of the action, i.e. bowl and oven for the events mix and heat respectively.

We choose relation extraction over procedural text as the task of interest because of the *implicit schemas* afforded by procedural text. Across domains, the datasets chosen describe the process of combining ingredients under certain conditions in a sequential fashion to produce a desired product. These range from preparing a cooking recipe to synthesizing a chemical compound to extracting materials from ores. As a result, the relations that we derive from each dataset share a *loose semantic correspondence*, both to each other and to basic semantic relations such as verb arguments and locations. For example, the actions "boil" and "heat" in "Boil the mixture in a medium saucepan" and "Heat the solvent in the crucible" are similar.

We use linguistic frameworks as scaffolds for the task of relation extraction because we hypothesize that the underlying semantics of different datasets are similar enough that models should be able to better generalize across domains from the explicit inclusion of syntactic and semantic structural features. We operate over three datasets across two domains, where each dataset defines the task of generating a comprehensive, descriptive graph representation of a procedure. We thus simplify this task into relation extraction in order to better compare the impact of different linguistic formalisms.

Additionally, we hypothesize that linguistic structures offer an abstraction over the way natural language varies over different domains, providing representations that might express meaning in domain-general ways. We therefore investigate whether including linguistic representations encourages learning domain-agnostic representations of relations such that models can generalize better in a few-shot setting, i.e. learning from less high-quality data in **new domains**. We now discuss the datasets and the linguistic frameworks we explore for relation extraction over procedural text, both in an in-domain and transfer setting.

Datasets

We consider three datasets across two different domains for this transfer: cooking and materials science procedures. Our cooking datasets are the RISeC (Jiang et al., 2020) and English Recipe Flow Graph (EFGC) (Yamakata et al., 2020b) corpora, and we introduce a much wider domain gap with the Materials Science Procedural Text Corpus (MSCorpus) from Mysore et al. (2019). We do not standardize labels across datasets; we retain the original labels from each dataset, though we combine some relations in MSCorpus to make it more comparable to the other datasets (see below for details). Summary statistics for each dataset (including for the definition of "relation") are shown in Table 3.1, and we describe salient features for each of our datasets below. Notably, all three of our datasets exhibit a high degree of concentration in their label distributions, with infrequent classes being found sometimes as much as 200× less than the most frequent classes.

Dataset	Documents	# Relations	Labels	Label Distribution
RISeC	260	7,591	11	
EFGC	300	15,681	13	
MSCorpus	230	18,399	11	

Table 3.1: Dataset Statistics. The label distribution column visualizes sorted frequencies of labels in each dataset.

The **RISeC** dataset is the most explicitly aligned with existing semantic frameworks: the authors build upon Propbank (Kingsbury and Palmer, 2002), which is also the framework that underlies AMR. However, because the relations in the dataset do not correspond strictly to verbal frames, relations use Propbank roles, rather than numbered arguments. Additionally, while these relations are *inspired* by Propbank, the authors' definitions of the labels do not always correspond to Propbank's, rendering this correspondence somewhat loose.

The **EFGC** dataset takes a more domain-specific approach, and defines a labeling schema specialized for cooking, including coreference relations segmented by whether the coreferent entities are tools, foods, or actions. Many of the descriptors of actions that are given explicit labels in RISeC such as temporal relations and descriptions of manner, are collapsed into a single class in this dataset, with the authors choosing to focus on physical components, their amounts, and operational relationships.

The **MSCorpus** dataset splits its relations into three categories: relations between operations and entities, relations between entities, and one relation indicating the flow of operations. MSCorpus defines a rich set of relations between entites, which is atypical for the other datasets. We thus combine some of these labels to bring MSCorpus into alignment with the other annotation schemas.

We refer the readers to the original papers for the definition of the relations used in this work.

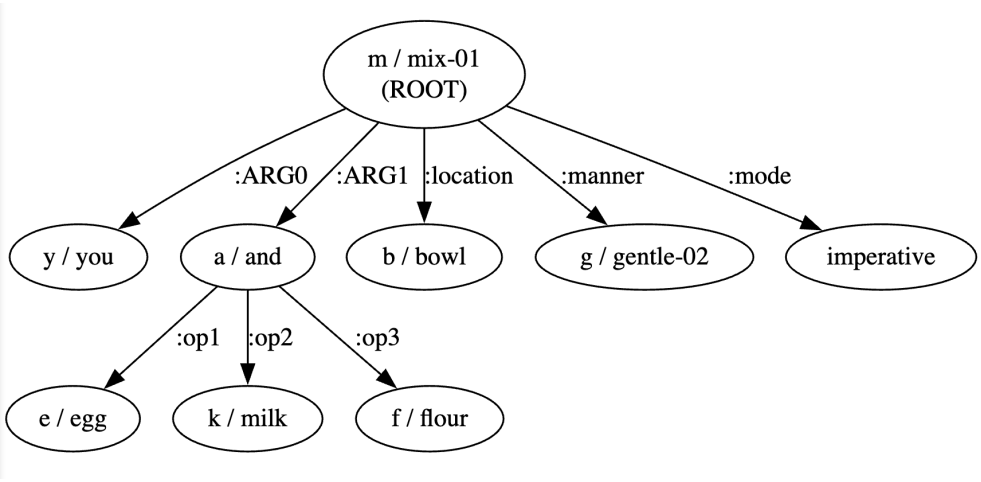


Figure 3.2: AMR for the sentence “Gently mix eggs, milk and flour in a bowl.”

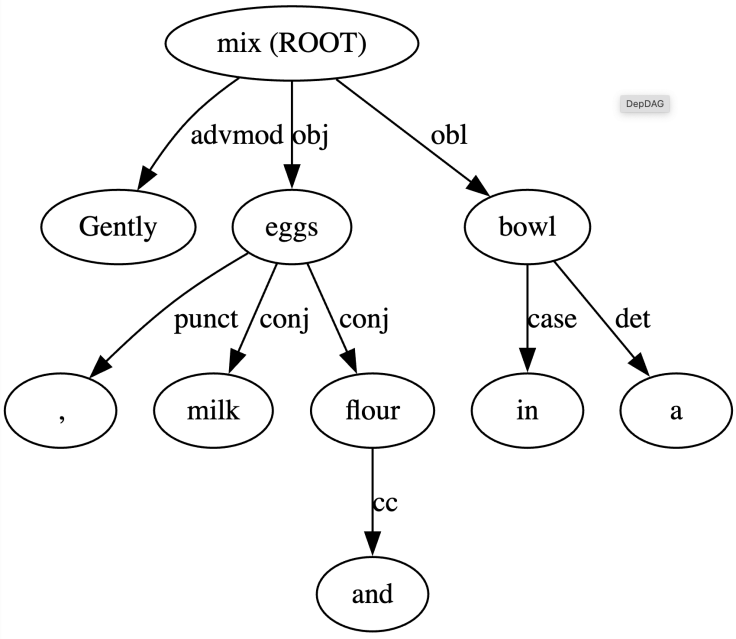


Figure 3.3: Dependency parse for the sentence “Gently mix eggs, milk and flour in a bowl.”

Linguistic Frameworks

We explore the role of linguistic frameworks as augmentations to the text on the task of relation extraction. We specifically focus on automatically generated linguistic annotations, to evaluate their impact on downstream task without expensive human annotation of parse data. We thus compare and contrast two linguistic formalisms, i.e. to evaluate and compare their impact on the task: dependency parses, and abstract meaning representations (AMR).

AMR or semantic parsing (Banarescu et al., 2013) seeks to represent meaning at the level of a sentence in the form of a rooted, directed graph. AMR is based on Propbank (Kingsbury and Palmer, 2002), and factors out syntactic transformations due to verb alternations, passivization, and relativization, leading to a less sparse expression of textual variance. Dependency parsing, by contrast, remains at a low level of abstraction, with structures that do not nest outside of the words in the original text. We illustrate the distinction between AMR and dependency parse for the sentence "Gently mix eggs, milk and flour in a bowl" in Figures 3.2 and 3.3 respectively.

3.2.2 Multilingual Relation Extraction

Language	Sentence	Relation
English	The <e1>Battle of al-Qaryatayn (2016)</e1> was a military operation launched by Syrian government forces, supported by <e2>Russia</e2>n airstrikes, to recapture the mainly Christian town of Al-Qaryatayn from the Islamic State of Iraq and the Levant.	Participant
Telugu	<e2>లక్ష్మణుడు</e2> చెప్పిన దానికి <e1>ఊర్మిళ</e1> చాలా ధైర్యంగా.. సమాధానం చెప్పింది. రాముడు, సీతకు తన భర్త చేయాల్సిన సేవలను అర్థం చేసుకుని, ఈ వనవాసం గడిపే రోజుల్లో తన గురించి కనీసం ఆలోచించకూడదని లక్ష్మణుడి నుంచి ప్రామిస్ తీసుకుంది.	Spouse
Hindi	इस प्रस्ताव को पाकिस्तान के साथ-साथ सऊदी अरब, पाकिस्तान, यमन, <e1>संयुक्त अरब अमीरात</e1>, मोरक्को, ओमान, बहरीन और मिस्र ने साथ पेश किया था जिसे <e2>संयुक्त राष्ट्र</e2> महासभा ने आम सहमति से स्वीकार कर लिया है.	Member_of

Figure 3.4: Examples of MLRE in different languages

The task of multilingual relation extraction (MLRE), involves identifying the nature of the relationship between two annotated entity spans in a multi-lingual text document. Subsequently, in the context of generalization or transfer, we investigate whether a given RE system achieves similar transfer performance on a target language as on the original source language, and whether linguistic frameworks can act as effective scaffolds to facilitate transfer across languages. MLRE as a task thus serves as a contrast to RE over procedural text since MLRE focuses on generalization

Language	#Sentences	#Relations
IndoRE		
English (en)	8486	51
Hindi (hi)	6963	51
Telugu (te)	8154	51
REDFM		
English (en)	10899	32
Spanish (es)	6538	32
French (fr)	7383	32
Italian (it)	6812	32
German (de)	7497	32
Arabic (ar)	1846	32
Chinese (zh)	1384	32

Table 3.2: Statistics for IndoRE and RedFM Datasets in terms of the number of sentences and relations. It is to be noted that the ar and zh languages for REDFM is used only for evaluation/inference.

across languages while the latter emphasizes generalization across domains. We present examples of MLRE in the Figure 3.4.

Datasets: We conduct our experiments for MLRE on two datasets i.e. IndoRE and REDFM. We present initial statistics for the datasets in Table **??**.

- **IndoRE (Nag et al., 2021):** The IndoRE dataset covers a diverse and rich set of entity and relation annotated sentences in three low resource Indian languages — Bengali (bn), Hindi (hi) and Telugu (te). To study protocols for transferring RE capability across languages, it also has labeled RE instances in english (en) as an example of a resource-high language. The dataset consists of 32,610 sentences combining all four languages from Wikidata where each language contains 51 unique relations. Out of these languages, we exclude Bengali from our experiments because the dependency parsers' inability to parse the language.
- **REDFM (Huguet Cabot et al., 2023):** The REDFM dataset consists of RE examples in 7 languages. These languages include English (en), Arabic (ar), Spanish (es), German (de), Italian (it), French (fr), and Chinese (zh), which are hand-annotated. There are a total of about 15,400 examples in the dataset with a total of 32 types of relations. We use the languages en, es, de, it, and fr for training (i.e. source languages), and all the 7 languages for testing in a zero-shot setting (i.e. target languages). We exclude Arabic and Chinese as source language due to the unavailability of a training split in the REDFM dataset. We use the train/validation/test splits as in the original paper.

Linguistic Frameworks We choose dependency parses as our choice of linguistic framework for MLRE. We experiment with Trankit (Nguyen et al., 2021b) and Stanza (Qi et al., 2020a) dependency parsers for both datasets across all languages. We did not explore semantic parses due to the unavailability of AMR parsers for certain low-resource languages.

3.3 Question Answering over structured knowledge sources

The task of question answering over knowledge graphs (KGQA) or knowledge bases (KBQA) involves extracting or generating responses for a natural language query that is contextualized (restricted) to a situated knowledge store or database.

Formally, a knowledge base or a KB (Gu et al., 2021; Dutt et al., 2023) is defined mathematically as $\mathcal{K} = (\mathcal{O}, \mathcal{M})$, where $\mathcal{O}$ defines the ontology of the KB and $\mathcal{M}$ specifies the set of relational facts present in $\mathcal{K}$ on the basis of $\mathcal{O}$. The ontology is a subset of all possible relations $\mathcal{R}$ that can exists between two classes, which are denoted by $\mathcal{C}$ i.e., $\mathcal{O} \subseteq \mathcal{C} \times \mathcal{R} \times \mathcal{C}$. Likewise, the set of facts is represented as $\mathcal{M} \subseteq \mathcal{E} \times \mathcal{R} \times (\mathcal{L} \cup \mathcal{E} \cup \mathcal{C})$, where $\mathcal{E}$ and $\mathcal{L}$ denote the set of possible entities and literals respectively.

Likewise, as defined in Dutt et al. (2022), a knowledge graph is a specific subcase of a KB where the data is organized in the form of a triple store or heterogeneous graph, with nodes representing entities and edges corresponding to relations between the entities. Formally a KG is represented as $\mathcal{K} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, where $\mathcal{V}$ is the set of entities, $\mathcal{R}$ is the set of relations, and $\mathcal{E}$ is the set of triplets. (e_1, r, e_2), $e_1, e_2 \in \mathcal{V}$, and $r \in \mathcal{R}$. Thus $\mathcal{E} \subset (\mathcal{V} \times \mathcal{R} \times \mathcal{V})$. Given a natural language question q, the objective of KGQA is to retrieve answer entities from $\mathcal{V}$.

The two common approaches for KBQA/KGQA as shown in Figure 3.5 include (i) an Information Retrieval (IR) based approach where the final answer is extracted from the corresponding knowledge graph (KG), and (ii) a Semantic Parsing (SP) based approach where the natural language query is converted into an equivalent logical form (such as s-expression or SPARQL query) that is executed over the KB to generate the answer. We explore both information-retrieval and semantic-parsing based approaches for QA over structured knowledge stores in this thesis.

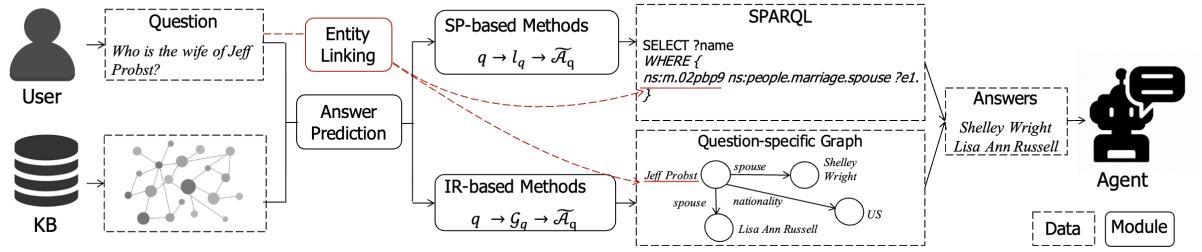


Figure 3.5: An illustration of different techniques for KBQA taken from Lan et al. (2022). These include both semantic parsing (SP) based methods where the objective is to convert a question into an equivalent logical form like SPARQL, and information retrieval (IR) based methods where the objective is to extract the answer given the corresponding question-specific graph.

3.3.1 PERKGQA: Question Answering over Personalized Knowledge Graphs

Motivation

Prior studies on KGQA have typically operated over a single knowledge graph (KG). This KG is assumed to be known a priori and is leveraged similarly for all users' queries during inference.

Such an assumption is not applicable to real-world settings, such as healthcare, where one needs to handle queries of new users over unseen KGs during inference, especially for user's queries that require situated knowledge such as personal information. Leveraging a single global KG for users' queries introduces the following concerns.

- **Scalability:** The massive size of the global KG makes it computationally expensive to apply sophisticated neural architectures over it.
- **Privacy:** The unfettered access to information of all individuals raises ethical or legal concerns.

We thus propose the task of question answering over knowledge graphs in a personalized setting, which we term PERKGQA. In such a setting, for a given user, the system has access to the specific knowledge graphs that contains information only relevant to the user. We are only restricted to the user's KG to answer only their queries both during training and inference. PERKGQA thus exhibits departure from prior KGQA research which has focused more on generalizable or generic knowledge, i.e. involves reasoning over all possible users' information.

PERKGQA appears deceivingly simple in conception since we afford access to a subset of the larger global KG. One can claim that our setting is similar to the KGQA subtask where subgraphs and questions are predefined, and thus, traditional KGQA methods are applicable. However, information retrieval based KGQA methods employ knowledge graph completion techniques like TransE (Bordes et al., 2013) to learn node representations over the global KG and reuse them during inference. Alternately, other approaches leverage additional information such as semantic parses, logical forms, and query graphs to answer queries.

This sets PERKGQA apart because we lack access to any prior information, be it text, semantic parses, or prior representations of KG nodes. Our setting requires learning node representations from scratch for each KG to handle unknown entities during inference. Moreover, other challenges prevalent in KGQA settings, namely multi-hop reasoning or answering complex/constraint-based questions, are also applicable to PERKGQA. To the best of our knowledge, we are the first to address the challenges of KGQA over unseen KGs in the absence of any additional information.

Task Formulation

A KG is defined mathematically as $\mathcal{K} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, where $\mathcal{V}$ is the set of entities, $\mathcal{R}$ is the set of relations, and $\mathcal{E}$ is the set of triplets, while the objective of information-retrieval based KGQA is to retrieve answer entities from $\mathcal{V}$ for a natural language question q. For PERKGQA, we treat each question as posed by a separate user, and each question is associated with its corresponding knowledge graph, $\mathcal{K}_q$. A given $\mathcal{K}_q$ has a subset of nodes, $\mathcal{V}_q$ and relations, $\mathcal{R}_q$. Two knowledge graphs, $\mathcal{K}_q$ and $\mathcal{K}_{q^*}$ associated with questions q and q^* can have a varying degree of overlap, even being distinctly different.

Running Example

We now demonstrate the applicability of PERKGQA for a cloud service provider (e.g. Microsoft Azure) in Figure 3.6. Here, users (blue and red) can create cloud resources (yellow), and index them using a unique system identifier. These resources have a corresponding user-specific tag (green), are located in a specific region (orange), and have predefined services deployed on them

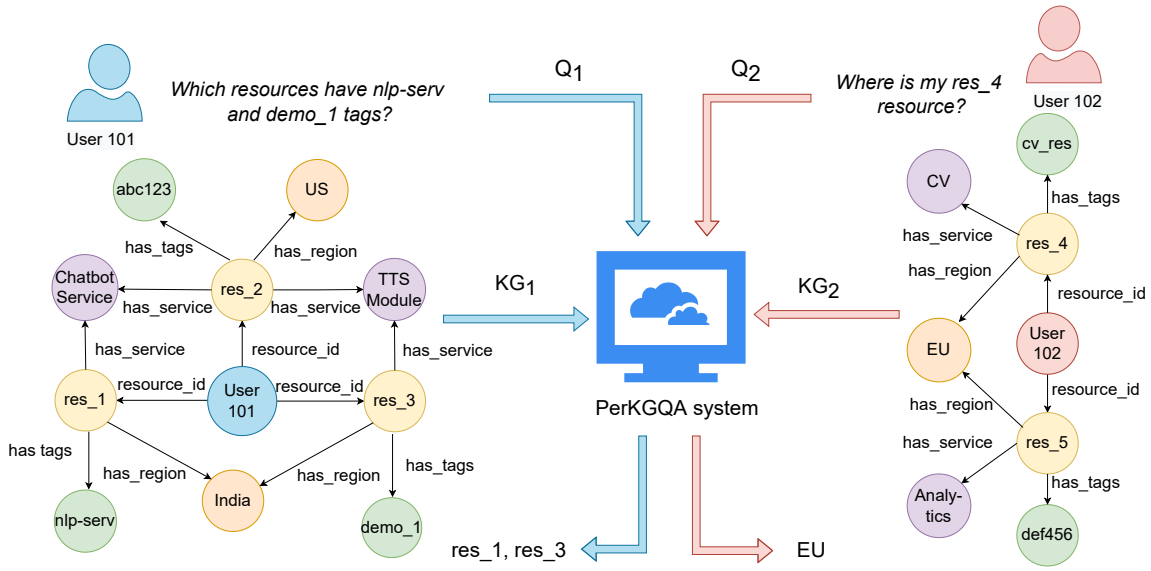


Figure 3.6: PERKGQA for a cloud service provider setting. The two users (in blue and red) create cloud resources (in yellow) in specific regions (in orange), and deploy services e.g. *Chatbot service*, or *Analytics* (in purple) on them. The users assign customized tags (in green) to the resources. Each user has their unique KG. The system should scale to support queries of new users over unseen KGs without any retraining or additional knowledge.

(purple). The entire system can be envisioned as a knowledge graph (CloudKG) where nodes represent concepts (users and services), and edges define the relations between concepts. Due to confidentiality, user names are replaced with anonymous identifiers, while concept and relation names in CloudKG are modified. The underlying schema is unchanged.

Deploying a chatbot-based assistant that performs QA over CloudKG would facilitate use, especially by novice users. It would enable users to navigate the system and glean information by posing natural language questions. In Figure 3.6, when User 101 asks "Which resources have nlp-serv and demo_1 tags?", the system is expected to answer "res_1, res_3". We refer to Figure 3.6 as a running example in subsequent sections. As new users become a part of CloudKG, the QA system should accommodate their requests over the corresponding KG without any training. KGQA approaches that operate upon the entire CloudKG would be computationally infeasible due to the massive size of the user-base [1]. [1] Moreover, the approach should be privacy-preserving wherein a given user's information is not revealed to another.

Datasets

We explore two distinct datasets; an internal dataset, CloudKGQA, built on top of CloudKG, and an academic dataset called Mod-WebQSP designed to mimic the PERKGQA setting. An instance in either dataset follows the same task formulation as described earlier, namely, for each question q, there exists a corresponding KG, K_q, which contains all the necessary information. Also, each question q is associated with one or more source entities; these correspond to nodes in the K_q

[11] https://www.statista.com/statistics/321215/global-consumer-cloud-computing-users/

linked through salient mentions of entities in q. E.g., the source entity for, "Who was responsible for Lincoln's assassination?" is the node corresponding to Abraham Lincoln.

CloudKGQA: The internal dataset, which we refer to as CloudKGQA, entails question-answering of a customer's queries on their respective cloud resources. We refer the readers to Figure 3.6 as we present examples that outline the key characteristics of CloudKGQA. Due to confidentiality, all customer examples in this paper are composed by the authors while explaining the characteristics of our data.

- **Multiple Answers:** A question can have one or more correct answers.
- **Varying Complexity:** A question can either be simple or complex.

  (i) **Simple:** The question can be answered by a single-hop relation, e.g. "Which resource has the tag nlp_serv?"

  (ii) **Complex :** The question involves logical operations like union or intersection, e.g."Show me resources in US and India" or contains multiple constraints, e.g. "Which resource has the TTS and MongoDB service and is located in US?" has three constraints, TTS, MongoDB, and US.
- **Multi-Hop distance:** The distance between the source entities and the answers is variable (e.g., the number of hops for "Show me tags for resources in US" is 2 in Figure 3.6).
- **Variable graph size:** The size of the KG varies in terms of the number of nodes, edges, and relations for each question.
- **Unseen nodes:** Nodes that appear in the KG during inference might not be seen while training.

Mod-WebQSP or Modified WebQSP: We mimic our setting on the publicly-available WebQSP dataset (Yih et al., 2016), which operates on the Freebase KG, $\mathcal{F}$, [2]. We use the pruned version of the dataset provided by Saxena et al. (2020). To completely mimic our setting, we construct a graph $\mathcal{F}_q$ associated with each question q with the caveat that a significant fraction of nodes we encounter during inference are not observed during training.

Each question is associated with a source entity as noted in the dataset of Saxena et al. (2020). The question's corresponding KG, comprises all nodes, a distance of k-hop from the source entity, where k is the shortest distance between the source and the answer. We limit ourselves to k=2 similar to Saxena et al. (2020). Furthermore, to constrain the size of $\mathcal{F}_q$, we randomly sample 1000 paths at a k-hop distance from the source entity; these are inclusive of all paths that lead from the source to an answer.

We observe a small fraction of questions ($\approx 5\%$), which have ≥ 100 answers; these correspond to simple 1-hop questions like *"What did Roald Dahl write?"*, or *"Who are famous people from Spain?"*. We remove such questions to constrain the size of the KG. Since, our objective was to retrieve all possible answers in the KG for a given question, there are no missing answers in the KG corresponding to the question in the Mod-WebQSP.

To achieve low overlap between nodes we encountered during training and inference, we modify them by assigning new identifiers. For example, a node, "m.0gzh" corresponding to "Abraham Lincoln", was modified to "**KG_i**_m.0gzh" and "**KG_j**_m.0gzh" for questions q_i and q_j

[2] `https://developers.google.com/freebase/data`

in their respective KG $\mathcal{F}_{q_i}$ and $\mathcal{F}_{q_j}$. Although these nodes have the same underlying entity name in the original KG, $\mathcal{F}$, their node representations are different in these two questions. We rank all relations in $\mathcal{F}$, based on the decreasing order of frequency, and chose the top 39 relations that occur in 95% of all triplets in $\mathcal{F}$. We modify only those nodes which are associated with these 39 relations.

We added the graph-identifiers to the most frequent relations to ensure a small degree of overlap between the training and the test sets, similar to the CloudKGQA dataset, where certain entities were universal like names of regions (India, USA). Our modification achieves a low overlap of 4.0% between entities across training and test splits, implying that 96% of entities remain unseen.

Dataset Comparison: We present the descriptive statistics of the two datasets in Table 3.3 corresponding to the mean number of nodes, edges, relations, answers, and hops for a KG. We also depict the degree of overlap between nodes in training and test splits. The number of instances in CloudKGQA and Mod-WebQSP are 800 and 4468, respectively. Moreover, we split the data into train, development, and test for both datasets in the ratio of 8:1:1.

We observe that CloudKGQA is comparatively smaller in size, had significantly fewer relations, but had longer reasoning chains. Moreover, CloudKGQA had more complex questions in terms of logical operations and multiple-constraints. Specifically, CloudKGQA had one or more source entities for each question, q, whereas Mod-WebQSP had only one source entity. The KGs in CloudKGQA had a similar underlying schema; different KGs had the same set of relations but different entities. However, the questions in the test data had distinct question templates from those during training, as seen in Figure 5.1. The Mod-WebQSP dataset, on the other hand, had KGs with different relations, but questions in the test data were similar to those asked during training. We chose these two datasets because they capture two different scenarios.

Dataset	CloudKGQA	Mod-WebQSP
Nodes	23.39	518.21
Edges	35.59	1334.10
Relations	8.00	36.20
Answers	1.99	4.94
Hops	1.75	1.36
Overlap	3.21%	4.01%

Table 3.3: An overview of the statistics of the two datasets, CloudKGQA and Mod-WebQSP. We present the mean number of nodes, edges, relations, answers, and hops, and the overlap between nodes during test and train.

Scaffolds

We propose techniques to tackle the nuanced challenges of PERKGQA. We leverage the internal structure of a new or unseen knowledge graph and the path information between the source nodes and answers as scaffolds to prevent reliance on learning node representations of the graph apriori,

and enables us to generalize over unseen KGs for new users. We investigate the utility of these scaffolds in detail in Chapter 5.

3.3.2 ISOKBQA: Isomorphisms for zero-shot generalization in KBQA

As opposed to information retrieval based approaches for KBQA, where we extract the answer for a given natural language query directly, semantic parsing based approaches operate by translating the natural language query (q) into a corresponding logical form (L_q). This logical form is executed over the KB to yield an answer. The logical form is composed of the following elements; (i) the classes and relations that constitute the schema of this L_q, (ii) the way these schema elements and operations are arranged in the form of a template i.e. T_q and (iii) the instantiation of the classes and relations in the query. Subsequently, based on the elements that comprise this logical form, we can extend the generalization capabilities of KBQA systems.

Zero-shot generalization in KBQA

We explore the generalization capabilities of KBQA systems beyond the i.i.d setting, where one might encounter unseen entities or KBs during inference, but still operate over the same schema items such as in the PERKGQA setting. This challenging "zero-shot generalization" setting requires operating over schema items such as classes and relations that were unobserved during training. Formally, Gu et al. (2021) proposed a hierarchical view of generalization in KBQA, going from i.i.d. to compositional to zero-shot generalization. The three levels of generalization are based on how the schema $\mathcal{S}_q$ and logical template $\mathcal{T}_q$ for a question q differs from the set of all possible schema items and templates seen during training, i.e. $\mathcal{S}_{train}$ and $\mathcal{T}_{train}$ respectively. We provide a pictoral representation of this hierarchical organization here in Figure 3.7.

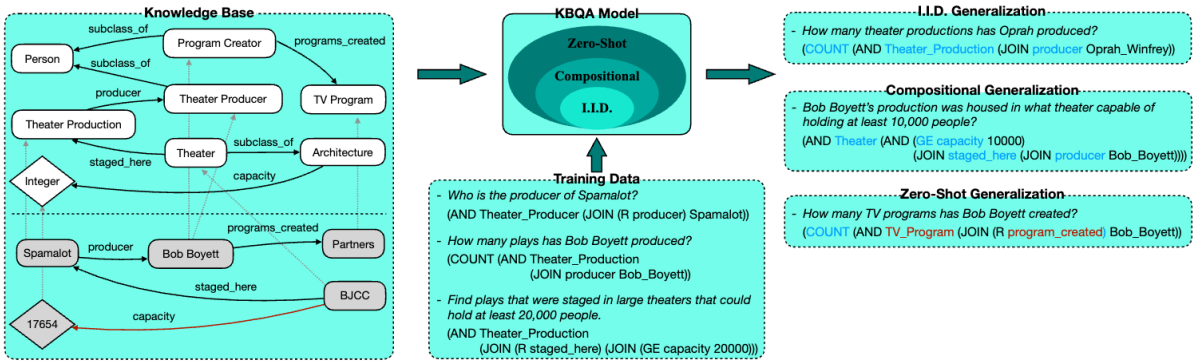


Figure 3.7: The three levels of generalization in KBQA as referenced in the paper of Gu et al. (2021).

(i) **I.I.D.** generalization occurs when $\mathcal{S}_q \subset \mathcal{S}_{train}$ and $\mathcal{T}_q \in \mathcal{T}_{train}$.

(ii) **Compositional** generalization occurs when $\mathcal{S}_q \subset \mathcal{S}_{train}$ but $\mathcal{T}_q \notin \mathcal{T}_{train}$. Thus the questions operate upon a subset of schema items seen during training but they have new templates.

(iii) **Zero Shot** generalization occurs when $\exists s \in \mathcal{S}_q$ such that $s \notin \mathcal{S}_{train}$. Thus the questions operate upon novel schemas, mostly new classes and relations that were not encountered during training.

Beyond the generalization levels, we propose a technique to categorize the different kinds of logical forms for KBQA using the concept of isomorphisms.

Isomorphisms

Each logical form L_q has an equivalent graphical notation $\mathcal{G}_q$, where the set of vertices V_q correspond to the different constraints $(\mathcal{E}, \mathcal{L})$ and classes $\mathcal{C}$ in the L_q while edges E_q represents the relations $\mathcal{R}$ present in L_q. This notation is similar to the design of query-graphs (Lan and Jiang, 2020) but where the operations (aggregation or comparative) do not have any specialized vertices. We however denote one of the vertices in V_q that correspond to the answers as $\mathcal{A}_q$ and call it the root. The nodes corresponding to the root and the constraints are denoted in green and red respectively in Figure 3.8.

We say two logical forms for questions q_i and q_j belong to the same isomorphism category, iff their equivalent graphs $\mathcal{G}_{q_i}$ and $\mathcal{G}_{q_j}$ are isomorphic. Subsequently, two graphs $\mathcal{G}_{q_i}$ and $\mathcal{G}_{q_j}$ are isomorphic iff there exists a mapping function ψ from V_{q_i} to V_{q_j} such that $\forall m, n$ nodes in V_{q_i}, that correspond to an edge in $\mathcal{G}_{q_i}$ i.e $(m, n) \in E_{q_i}$, the mapping of the nodes should also correspond to an edge in $\mathcal{G}_{q_j}$ or, $(\psi(m), \psi(n)) \in E_{q_j}$. This mapping is bijective. Furthermore the roots in the two graphs also share the same mapping, i.e. $\mathcal{A}_{q_j} = \psi(\mathcal{A}_{q_i})$.

Isomorphisms describe how the constraints in the query graph are connected to the root (or the answer). It obfuscates any specific information such as the name of the entities or classes in the graph. They provide a unified way to characterize a query graph (and subsequently a logical form) based on the number of constraints, and the number of hops required to reach the answer from said constraints. For example, in Figure 3.8, the green Tea node corresponds to Ans while the red constraint nodes, Fujian and White tea, corresponds to E1 and E2 respectively. Thus the given logical form is an instance of Iso-2. The distribution of isomorphisms spanning all datasets appears in Table 3.6.

While the notion of isomorphisms is similar in concept to the idea of reasoning paths (Das et al., 2022) or semantic structures (Li and Ji, 2022), we use the generic definition of "isomorphisms" to take into account that these isomorphisms can also be cyclic. For example, in Table 3.6, we note instances of isomorphisms (CIso-0 to CIso-4) where at least one cycle is present.

Isomorphism distribution in pre-existing KBQA datasets.

The most salient work in the space of zero-shot generalization is that of Gu et al. (2021). The authors created GrailQA, a dataset to benchmark the different levels of generalizability of KBQA models. This dataset has garnered significant research interest with state-of-the-art KBQA models (Ye et al., 2021b; Yu et al., 2022a; Gu and Su, 2022; Shu et al., 2022; Liu et al., 2022c) achieving remarkable performance on the leaderboard, specifically on the zero-shot setting. [3]

However, a closer inspection of the GrailQA dataset reveals that it is biased towards simpler isomorphisms. We first categorize the questions in GrailQA according to the isomorphism type of

[3] https://dki-lab.github.io/GrailQA/

the corresponding logical form. We refer to isomorphisms with fewer than 3 relations as simple and the rest as complex isomorphisms. The simple isomorphisms for the remainder of the chapter are Iso-0,1,2. We show the distribution of the isomorphisms in the zero-shot development data of GrailQA in Table 3.4. A similar tale holds for the other KBQA datasets as we see in Table 3.6.

We observe that the simple isomorphisms (Iso-0, 1, 2) comprise more than 97% of all zero-shot examples in the development set. A similar story holds true for the train set where 95% of all isomorphisms belong to these three classes (See Table 3.6). We hypothesize that this skewness could exaggerate the perceived generalization capabilities of KBQA models, such that the staggering numbers on the leaderboard reflect the performance on these simpler isomorphisms. We thus propose creating a more challenging dataset to accurately benchmark the generalization capabilities of KBQA systems.

GrailQA++ Dataset Creation

To gauge whether KBQA models exhibit zero-shot generalization capabilities across different isomorphisms, we propose GrailQA++, a challenging dataset with an equal distribution of simple and complex isomorphisms. To create GrailQA++, we not only employ annotators with prior expertise in KBQA, but also leverage pre-existing KBQA datasets. We outline the creation process below and illustrate the same in Figure 3.8, using both pre-existing datasets as well as expert annotations.

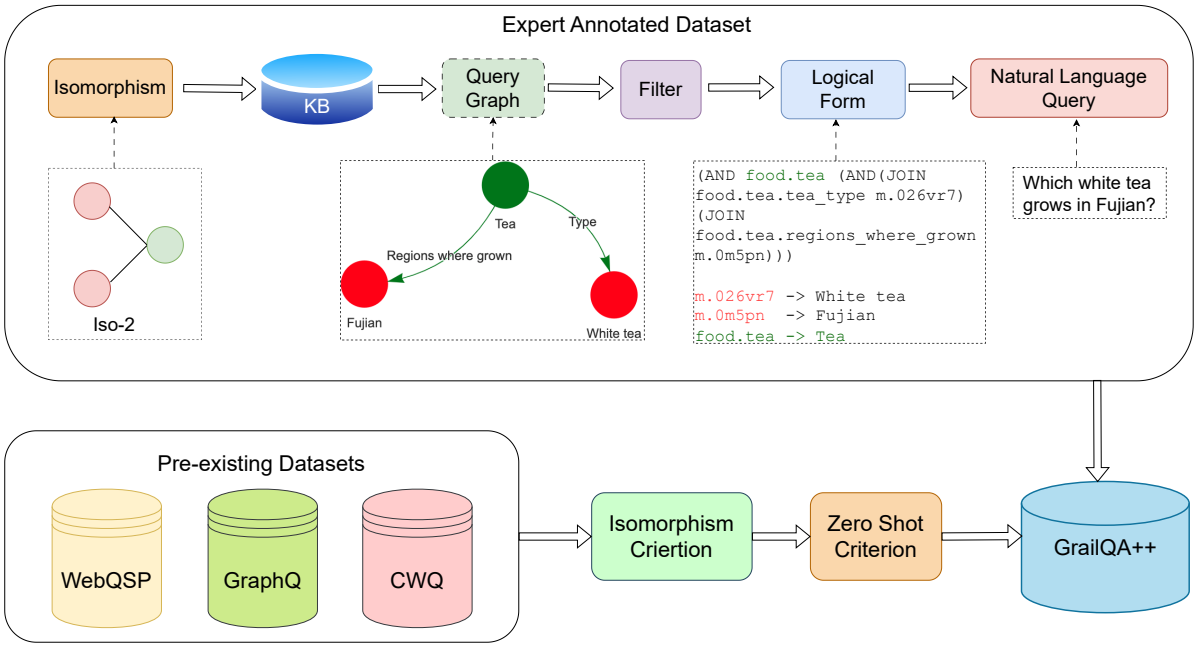


Figure 3.8: Schematic diagram that outlines the GrailQA++ dataset creation. The dataset comprises of question and corresponding logical forms, from two different sources. The former are instances which are hand-annotated by domain experts, and the latter are instances obtained from pre-existing datasets (WebQSP, CWQ, and GraphQ) which also operate over Freebase KB).

Expert Annotated Datapoints

We describe our controlled approach to sample and annotate instances of different isomorphism classes. Our process is similar to that of GrailQA albeit with a few differences, namely in terms of query sampling and natural language query generation.

Query Graph Sampling: GrailQA was created using the OVERNIGHT process (Su et al., 2016) which extracts templates by traversing Freebase and obtains a query graph. Since traversal is easier for simpler hops and subsequently simpler isomorphisms, they appear higher in GrailQA. We, however, follow a more controlled algorithm to sample the query graph.

We first choose a particular isomorphism, which determines the number of constraints. If there is exactly one constraint (Iso-0, 1, and 5), we first choose a class at random and then sample an entity randomly from that class. We then follow the relations that originate from the instantiated entity and continue our traversal of the KB till we reach the answer node. In case of multiple constraints (Iso-2, 3, 4, and 11), we first randomly sample the answer class and then traverse the KB by adding relations in a manner that conforms with the isomorphism structure. At each expansion step, we ensure that there exists an entity which can be instantiated using the new relation. This ensures executability of the current sub-query and thus of the main query.

The authors chose to sample instances corresponding to Iso-0,1,2,3,4,5 since these were already present in the zero-shot split of GrailQA. Additionally, we also sampled and annotated instances of Iso-11, since it was the simplest isomorphism that could be formed with three constraints.

Filtering: We filter query graphs that do not conform with the zero-shot generalizability criteria. Specifically, the query graph should have at least one class or relation absent from the GrailQA training split. Later, we employ the filtering techniques proposed in Gu et al. (2021) to discard illegal relations, and ignore instances with entities or relations written in a language other than English.

Logical Form: Once we obtain the filtered query graph, we convert it to its canonical logical form using the deterministic algorithm of Gu et al. (2021). We then execute this logical form over Freebase to obtain the answers, and discard instances where the logical form was inexecutable or unanswerable.

Natural Language Query Annotation: To create the corresponding natural language question we choose annotators who are fluent in English, are current working professionals with a graduate degree to their name, and have prior domain expertise in KBQA. The annotators are first provided with a design document with examples of query graphs and their corresponding logical form. We also provide the annotators with aliases of the constraints and relations to better interpret the query graph as they compose the corresponding question. Examples of these screenshots appear in Figure 3.9.

We randomly select 35 instances (5 from each isomorphism) to include in the pilot study after which the annotators meet to discuss their interpretations and resolve any differences. We find

Iso-2

S-expression

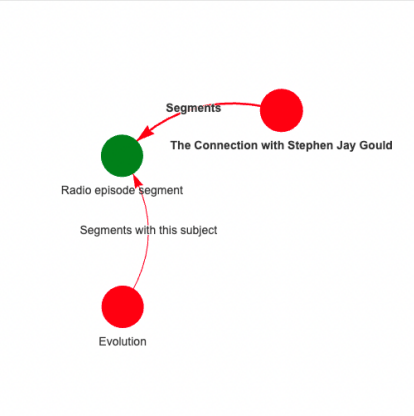
```
(AND radio.radio_episode_segment (AND (JOIN
(R radio.radio_subject.segments_with_this_subject) m.02j8z)
 (JOIN (R radio.radio_program_episode.segments) m.0blhhfg)))
```

Constraints and Relations

```
m.02j8z -->  Evolution
m.0blhhfg -->  The Connection with Stephen Jay Gould
radio.radio_episode_segment -->  Radio episode segment
```

Answer

```
m.0blhk6j --> Christopher Lydon with Stephen Jay Gould
```



Iso-3

S-expression

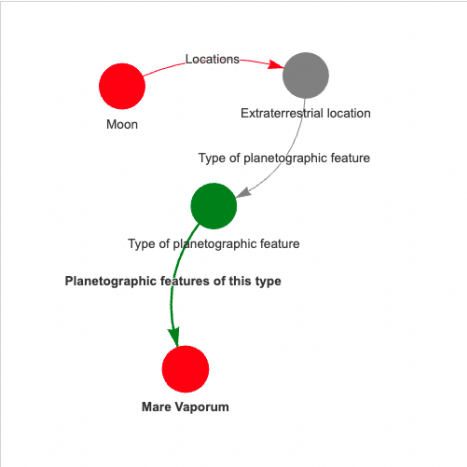
```
(AND astronomy.type_of_planetographic_feature (AND (JOIN
(R astronomy.extraterrestrial_location.
type_of_planetographic_feature)
(JOIN (R astronomy.celestial_object.locations) m.04wv_))
(JOIN astronomy.type_of_planetographic_feature.
 planetographic_features_of_this_type m.01d80r)))
```

Constraints and Relations

```
astronomy.type_of_planetographic_feature -->
Type of planetographic feature
astronomy.extraterrestrial_location -->  Extraterrestrial location
m.04wv_ -->  Moon
m.01d80r -->  Mare Vaporum
```

Answer

```
m.03dy13 --> Lunar mare
```



Iso-5

S-expression

```
(AND metropolitan_transit.transit_line (JOIN
(R metropolitan_transit.transit_stop.terminus_for_lines)
(JOIN (R metropolitan_transit.transit_line.stops)
(JOIN metropolitan_transit.transit_line.service_type m.0452jfk))))
```

Constraints and Relations

```
m.0452jfk -->  Van
metropolitan_transit.transit_line -->  Transit Line
metropolitan_transit.transit_stop -->  Transit Stop
metropolitan_transit.transit_line -->  Transit Line
```

Answer

```
m.0452j59 --> Quartzsite Transit Service
m.0403pn4 --> Walnut Line
```

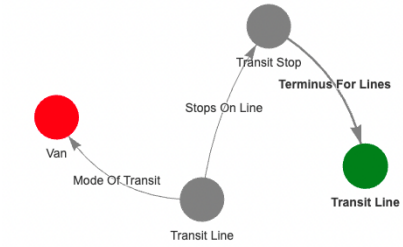


Figure 3.9: Annotation screenshots for three isomorphism categories; Iso-2 (top), Iso-3 (middle), and Iso-5 (bottom). For each instance, we provide the S-expression, the friendly name for each entity and relation, as well as the answers.

that all three annotators agree on 75% of the examples, while at least two agree on 97%. The main causes of disagreement was determining how explicitly the entities should be referred in the NL query. The annotators decided to be explicit in specifying the hidden nodes to facilitate evaluation. Finally, we sample a large set with 1000 unique query-graphs equally distributed among the three annotators. We ensure a balanced distribution between the different kinds of isomorphisms (see Table 3.4). All annotations were carried out by domain experts and we did not employ any crowd-workers unlike in Gu et al. (2021) for paraphrasing.

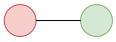
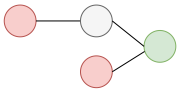
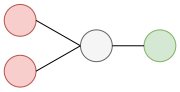
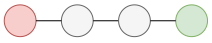
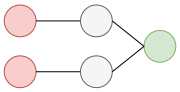
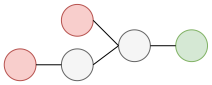
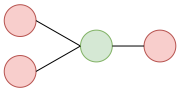
				GrailQA++									
Code	Pictoral Desc.	GrailQA		EAD		GraphQ		WebQSP		CWQ		Tot	
		Freq	Perc	Freq	Perc	Freq	Perc	Freq	Perc	Freq	Perc	Freq	Perc
Iso-0		2809	77.9	83	11.9	292	43.9	245	43.2	0	0.0	620	16.1
Iso-1		559	15.5	151	21.7	237	35.6	177	31.2	324	16.8	889	23.0
Iso-2		135	3.8	96	13.8	33	5.0	6	1.1	289	14.9	424	11.0
Iso-3		18	0.5	81	11.6	31	4.7	3	0.5	695	35.9	810	21.0
Iso-4		61	1.7	101	14.5	39	5.9	136	24.0	0	0.0	276	7.2
Iso-5		22	0.6	98	14.1	33	5.0	0	0.0	252	13.0	383	9.9
Iso-6		0	0.0	0	0.0	0	0.0	0	0.0	302	15.6	302	7.8
Iso-8		0	0.0	0	0.0	0	0.0	0	0.0	72	3.7	72	1.9
Iso-11		0	0.0	85	12.2	0	0.0	0	0.0	0	0.0	85	2.2

Table 3.4: Distribution of isomorphisms in the GrailQA (Dev) set and our curated GrailQA++ dataset (Tot). We show the total count of isomorphisms for each of the datasets (Freq) and their corresponding proportion in % (Perc). Note that complex isomorphisms belonging to Iso-6, Iso-8, and Iso-11 do not occur in the original GrailQA dataset. The red and green nodes in each isomorphism correspond to the constraints and the final answer respectively.

Pre-existing Datasets

We also leveraged pre-existing public datasets that were built over the same Freebase KB as GrailQA. These datasets were chosen since they were designed to evaluate progress on KBQA.

- **WebQSP** (Yih et al., 2016) uses Amazon Mechanical Turk to answer questions from non-experts collected using the Google Suggest API. Since the dataset is restricted to to "wh" questions from non-experts the questions tend to more colloquial.

- **GraphQ** (Su et al., 2016) was created in a fashion similar to GrailQA with questions exhibiting variation in terms of complexity, topic space, and number of answers.
- **ComplexWebQuestions (CWQ)** (Talmor and Berant, 2018) was created on top of WebQSP with the intention of generating complex questions by incorporating compositions (more hops), conjuctions (more constraints), and superlatives and comparatives (more function types).

Zero-shot splits: We consider only the questions in the test splits of the pre-existing datasets which satisfy the zero-shot crieteria of Gu et al. (2021). Specifically, zero-shot instances have at least one schema item (class or relation) that were not seen during training in the training data of GrailQA. Following Khosla et al. (2023), we also exclude questions if a relation's corresponding inverse relation was observed during training to make the task more challenging. We follow the same criteria for the expert annotated dataset as well.

Isomorphism crietrion: We sample instances corresponding to the following isomorphisms, Iso-0,1,2,3,4,5,6,8. The selection of these isomorphisms were driven by two criteria, namely (i) the isomorphisms should be present in the training split of the GrailQA dataset and (ii) there should be sufficient representation of these isomorphisms in the combined test-split of GrailQA++(¿50).

Distribution of isomorphisms for GrailQA++

We present the distribution of isomorphisms corresponding to our curated GrailQA++ in Table 3.4. We see that simple and complex isomorphisms are equally represented in the dataset, where the simple isomorphsims that correspond to Iso-0,1,2 comprise 50.1% of the dataset. We also include isomorphisms corresponding to Iso-6, Iso-8, and Iso-11 which are absent in the original dev split of GrailQA. This enables us to evaluate the zero-shot generalization performance of KBQA models on these unseen isomorphism categories. Subsequently, we thus use isomorphisms as a diagnostic tool to investigate the zero-shot generalization capabilities of KBQA systems.

Isomorphisms as Scaffolds

Beyond diagnostic tools, we hypothesize that isomorphisms can also serve as scaffolds to facilitate zero-shot generalization in KBQA systems without any additional training. The idea is that isomorphisms capture the overall structure of the reasoning path from the starting nodes(or constraints) to the final answer. Subsequently, we hypothesize that isomorphisms should serve as a strong prior to improve the performance of KBQA systems. This can be achieved either by filtering out candidate logical forms that do not conform to the isomorphism category or by guiding the exploration in a manner that aligns with the form of the isomorphism. As we show later in the next chapter(s), integrating the isomorphism information with pre-existing systems or LLMs does indeed bring about performance gains.

However, a caveat of this approach requires one to provide the isomorphism category as part of the input, information that is unlikely to be available during inference. We thus propose the task of **isomorphism prediction** where the objective is to predict the isomorphism category for a KBQA question in the context of the KB schema that the question is based on. We need to ground

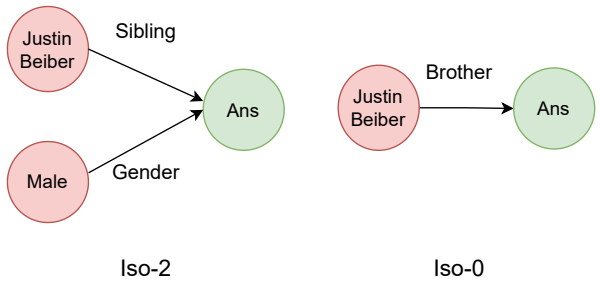


Figure 3.10: A simple example to show how the underlying schema impacts the isomorphism category for the same query.

the query into a given KB schema since the operationalization of a logical form is dependent on the particulars of the KB schema. For example, the question "Who is Justin Beiber's brother?" could give rise to different isomorphism categories as we show in Figure 3.10. Provided the relation "brother" exists in the schema, the corresponding isomorphism is simply Iso-0. However, if the "brother" relation is subsumed by a more generic "sibling" relation, one would need the additional constraint of the gender being "male" to represent the same query, yielding an instance of Iso-2. Thus the same query can give rise to two different isomorphisms if they operate over different KB schemas.

Isomorphism Prediction Task and Datasets

We now formally define the task of isomorphism prediction. Given a natural language query, q over a knowledge base, its corresponding logical form is denoted as L_q. One can deterministically convert the logical form into its equivalent graphical notation G_q or query graph, and subsequently the isomorphism category that represents the overall graph structure. While theoretically there can be an infinite number of isomorphism categories, for sake of simplicity, we frame the task of isomorphism prediction as a multi-class classification task by restricting the number of possible isomorphism categories.

The categories we consider for isomorphism prediction include an equal distribution of simple and complex isomorphisms. These include Iso-0, Iso-1, and Iso-2 corresponding to the simple isomorphisms and Iso-3, Iso-4, and Iso-5 corresponding to the complex ones. We restrict ourselves to the aforementioned candidates since these are the only isomorphism categories that occur in the public development dataset of GrailQA (Gu et al., 2021). Thus, formally for a model or system f, the task is to classify a given q and the schema associated with the question or KB_q into one of the six isomorphism categories.

$$f[q; KB_q] \rightarrow C, \;\; C \in \{\text{Iso-0,1,2,3,4,5}\}$$

We realize the task of isomorphism prediction on two datasets commonly used for KBQA, i.e. WebQSP (Yih et al., 2016) and GrailQA (Gu et al., 2021). Despite operating over the same Freebase, these datasets are organized differently and enables us to demonstrate the applicability

WebQSP

Question: what is the king of spain's name?

Structured Input: Spain: m.06mkj | m.06mkj location.location.contains m.0g3qgy | m.06mkj location.location.contains m.02qf5mh | m.0j5_3sv government.government_position_held.office_position_or_title m.0j5_3sz | m.06mkj location.location.contains m.02zb43k | .... | m.06mkj government.governmental_jurisdiction.governing_officials m.010wswjc | m.06mkj location.location.contains m.09k5hy | m.010wswjc government.government_position_held.office_position_or_title m.0j5_3sz | m.06mkj location.location.contains m.02z98t5 | m.06mkj location.location.contains m.03qcr60

S-expression: (JOIN (R government.government_position_held.office_holder) (AND (JOIN government.government_position_held.time_macro 2015^^http://www.w3.org/2001/XMLSchema#date) (AND (JOIN government.government_position_held.office_position_or_title m.0j5_3sz) (JOIN (R government.governmental_jurisdiction.governing_officials) m.06mkj))))

Isomorphism: Iso-4

GrailQA

Question: what is the role of opera designer gig who designed the telephone / the medium?

Structured Input: m.0pm2fgf: opera.opera_production: The Telephone / The Medium | opera.opera_role opera.opera_role.opera opera.opera_production | opera.opera_production opera.opera_role.opera opera.opera_role | ... opera.opera_character opera.opera_role.role opera.opera_role | opera.opera_production opera.opera_production.cast opera.opera_role | opera.opera_production opera.opera_production.designers opera.opera_designer_gig

S-expression: (AND opera.opera_designer_role (JOIN (R opera.opera_designer_gig.design_role) (JOIN (R opera.opera_production.designers) m.0pm2fgf)))

Isomorphism: Iso-1

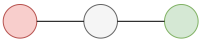
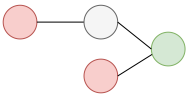
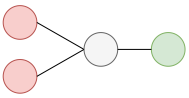
Figure 3.11: An example of the isomorphism prediction task for the WebQSP and GrailQA dataset

of the isomorphism prediction task in different scenarios. We show a representative example from each dataset in Figure 3.11.

WebQSP: We adopt the WebQSP dataset of Xie et al. (2022) for our isomorphism prediction task. Here, each natural language query, q is accompanied by a subset of the knowledge graph with the corresponding s-expression as the answer. The knowledge graph is organized as a set of tuples, where each tuple is of the form (e_h, r, e_t), i.e. a head entity e_h and a tail entity e_t is connected by a relation r . The head and tail entities such as "`m.06mkj` or Spain" constitutes the nodes in the KG while the relation r such as "location.location.contains" represents the edge. The WebQSP dataset is designed in such a manner that for a given query, the relations and entities that constitute the corresponding s-expression, as well as the final answer are present within the provided knowledge graph (KG). To extend this dataset for our isomorphism prediction task, we deterministically convert the s-expression into the corresponding isomorphism category.

We consider instances whose isomorphism category conforms to the aforementioned six classes; accounting for $\geq 97\%$ of the dataset. We also filter the instances where the resultant s-expression or answer is not self-contained in the accompanying KG. Moreover, an exploratory analysis of the dataset highlighted a significant overlap of relations and classes across the train and test splits. Subsequently, we employ the approach of Jiang and Usbeck (2022) to obtain development and test splits that characterize different generalization levels in almost equal proportion, i.e. i.i.d, compositional, and zero-shot.

GrailQA: We also explore GrailQA as a potential dataset for the isomorphism prediction task;

Isomorphism	WebQSP Train	WebQSP Dev	GrailQA Train	GrailQA Dev
Iso-0 	1884	164	20814	8972
Iso-1 	517	61	5385	2209
Iso-2 	150	10	1407	585
Iso-3 	169	6	588	279
Iso-4 	294	48	259	126
Iso-5 	-	-	274	142

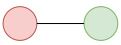
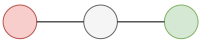
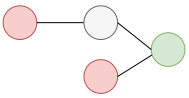
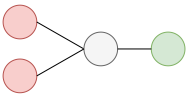
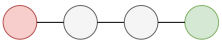
Isomorphism	WebQSP i.i.d	WebQSP Comp	WebQSP ZS	GrailQA i.i.d	GrailQA Comp	GrailQA ZS	GrailQA++ ZS
Iso-0 	368	-	364	1080	905	2782	87
Iso-1 	118	140	58	270	212	456	156
Iso-2 	12	33	25	62	100	121	100
Iso-3 	25	2	3	29	74	10	83
Iso-4 	25	159	11	9	-	43	107
Iso-5 	-	-	-	26	9	17	99

Table 3.5: Distribution of isomorphisms across the train and validation splits for WebQSP and GrailQA (on top) and the corresponding test split (in the bottom) .

we use the train split of GrailQA for finetuning models and subsequently evaluate them on the publicly available GrailQA development split and GrailQA++. To avoid data leakage in evaluating LLMs for KBQA, we restrict the GrailQA++ dataset to the expertly annotated instances and not the publicly available datasets. Each instance in GrailQA follows a similar design to WebQSP; the input consists of a question and the structured knowledge (organized as a set of tuples), while the output is the s-expression and corresponding isomorphism category.

However, unlike WebQSP, a given query in GrailQA is not accompanied by their corresponding knowledge graph; we need to create the structured information for each instance. Moreover, the knowledge tuples in GrailQA consists of Freebase classes rather than the entities directly. We thus employ the approach of Shu et al. (2022) to retrieve the top 10 classes and the top 10 relations for a given query. We can then construct a corresponding knowledge store by organizing the retrieved schema items as tuples. We also simultaneously query the Freebase ontology to fill in any missing

information, such as the absence of a given relation between two retrieved classes. As opposed to retrieving the subgraph directly from the KB ontology, (Zhang et al., 2022a) our approach yields significantly higher coverage across both datasets. The mean coverage across classes and relations is 97.0% and 96.8% respectively for GrailQA and 97.3% and 96.4% for GrailQA++, when we retrieve the top 10 instances for each schema category.

We present the distribution of isomorphisms in the training and test dataset in Tables 3.5. For the test split, we show the distribution of isomorphisms over the different generalization levels, i.e. i.i.d, compositional (Comp), and zero-shot (ZS). As referenced earlier, the distribution of isomorphisms is more skewed towards the simple categories for WebQSP and GrailQA. Subsequently, we anticipate GrailQA++ to be a more challenging dataset for the isomorphism prediction task. We delve deeper in Chapter 6 on how we design modules to solve isomorphism prediction and subsequently used the predicted isomorphisms to improve the performance of LLMs on KBQA.

3.4 Conclusion and Takeaways

In this chapter, we define formal scaffolds: structured frameworks that anchor text to formal representations either via external ontologies (e.g. knowledge graphs) or internal linguistic structures (syntactic and semantic parses). We operationalize formal scaffolds for three concrete tasks, i.e. relation extraction (RE), question answering over personalized knowledge graphs (PERKGQA) and isomorphism prediction. For each task, we note the kind of generalization that is being investigated (i.e. cross-domain, cross-lingual, zero-shot schema generalization) and motivate how the particular scaffold can facilitate generalization.

This chapter serves as preamble to the experimental methodology and the corresponding results/analyses that will be carried out in the subsequent chapters. Chapter 4 highlights whether internal linguistic frameworks can aid relation extraction. Likewise Chapter 5 outlines our methodology for PERKGQA and how graph structure and path information can facilitate generalization to new users. Finally we conclude the first part of our thesis with Chapter 6 where we inspect the dual role of isomorphisms both as diagnostic tools as well as scaffolds in zero-shot generalization for KBQA.

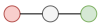
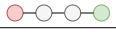
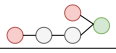
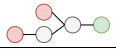
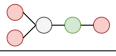
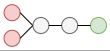
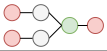
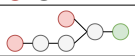
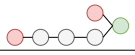
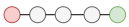
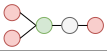
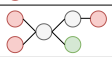
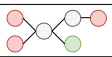
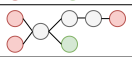
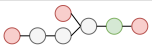
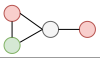
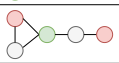
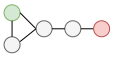
Iso-Code	Pictoral Desc	GrailQA		GraphQ		WebQSP		CWQ	
Iso-0		30496 (68.8)	2809 (77.9)	1533 (64.4)	640 (52.2)	1750 (58.3)	288 (43.8)	0 (0.0)	0 (0.0)
Iso-1		8900 (20.1)	559 (15.5)	544 (22.8)	391 (31.9)	716 (23.8)	197 (30.0)	5869 (21.8)	360 (14.5)
Iso-2		2676 (6.0)	135 (3.7)	167 (7.0)	37 (3.0)	106 (3.5)	19 (2.9)	4046 (15.0)	311 (12.5)
Iso-3		1066 (2.4)	18 (0.5)	69 (2.9)	50 (4.1)	35 (1.2)	6 (0.9)	6496 (24.1)	755 (30.4)
Iso-4		523 (1.2)	61 (1.7)	5 (0.2)	68 (5.6)	352 (11.7)	136 (20.7)	0 (0.0)	0 (0.0)
Iso-5		540 (1.2)	22 (0.6)	48 (2.0)	33 (2.7)	1 (0.0)	0 (0.0)	3712 (13.8)	279 (11.2)
Iso-6		42 (0.1)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	2454 (9.1)	324 (13.0)
Iso-7		26 (0.1)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	188 (0.7)	6 (0.2)
Iso-8		18 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	1 (0.0)	0 (0.0)	644 (2.4)	76 (3.1)
Iso-9		12 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	7 (0.2)	2 (0.3)	217 (0.8)	19 (0.8)
Iso-10		15 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)
Iso-11		23 (0.1)	0 (0.0)	0 (0.0)	6 (0.5)	4 (0.1)	1 (0.2)	225 (0.8)	17 (0.7)
Iso-12		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	30 (1.0)	8 (1.2)	0 (0.0)	0 (0.0)
Iso-13		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	69 (0.3)	14 (0.6)
Iso-14		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	810 (3.0)	111 (4.5)
Iso-15		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	57 (0.2)	9 (0.4)
Iso-16		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	1102 (4.1)	117 (4.7)
Iso-17		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	501 (1.9)	49 (2.0)
Iso-18		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	319 (1.2)	27 (1.1)
Iso-19		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	67 (0.2)	3 (0.1)
Iso-20		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	5 (0.0)	0 (0.0)
Iso-21		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	72 (0.3)	10 (0.4)
Iso-22		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	12 (0.0)	0 (0.0)
Iso-23		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	13 (0.0)	0 (0.0)
Iso-24		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	9 (0.0)	0 (0.0)
Iso-25		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	1 (0.0)	0 (0.0)
Iso-26		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	2 (0.0)	0 (0.0)
CIso-0		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	1 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)
CIso-1		0 (0.0)	0 (0.0)	15 (0.6)	0 (0.0)	1 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)
CIso-2		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	13 (0.0)	0 (0.0)
CIso-3		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	39 (0.1)	0 (0.0)
CIso-4		0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	2 (0.0)	0 (0.0)

Table 3.6: Distribution of different isomorphisms across the training and test splits for KBQA datasets. We include only instances in the test split that conform with the zero-shot criteria of GrailQA.

Chapter 4

Linguistic Frameworks for relation extraction across domains and languages

In this chapter, we motivate the role of linguistic frameworks in the form of dependency parses and abstract meaning representations (AMRs) as formal scaffolds for few-shot relation extraction (RE). We formalize this task to inspect generalization in two settings, i.e. generalization across domains for procedural text, and generalization across multiple languages.

4.1 Relation Extraction over Procedural Text

We augment a popular transformer-based relation extraction baseline with features derived from AMR (Banarescu et al., 2013) and dependency parses and investigate their impact in a few-shot setting both in-domain and across domains. Experiments show that both AMR parses and dependencies significantly enhance model performance in few-shot settings but that the benefit disappears when models are trained on more data. We additionally find that while cross-domain transfer can degrade the performance of purely text-based models, models that incorporate linguistic graphs provide gains that are robust to those effects.

4.1.1 Methodology

We design our methodology to test whether the inclusion of AMRs and dependency parses can improve the few-shot RE performance across datasets, by incorporating features from linguistic representations. We show an overview of our architecture in Figure 4.1, and go into further detail in Section 4.1.1. Our three datasets have their goal of generating a complete graph representation of a specified procedure. This graph is constructed by first finding salient entities in the procedural text, and then correctly identifying the appropriate relations between them. While this joint task is both challenging and useful, we restrict ourselves to the RE task for two reasons. Firstly, entity recognition results, as measured by baselines proposed in each of the dataset papers, vary widely, and entity recognition accuracy imposes an upper bound on end-to-end relation classification. Secondly, RE presents a common way to frame the tasks in each of these datasets.

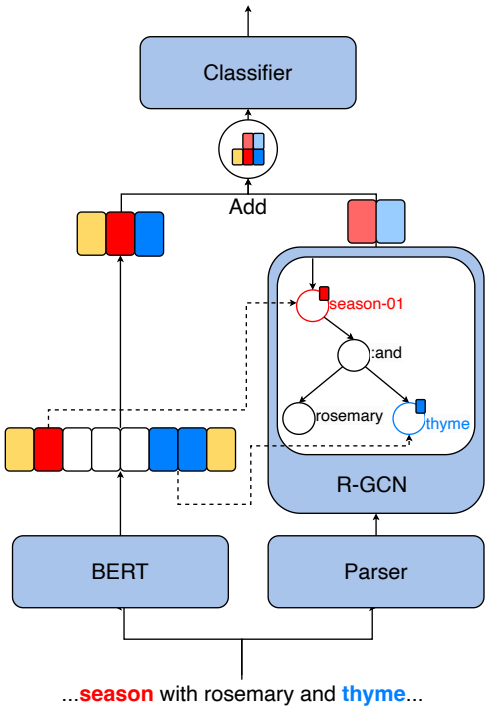


Figure 4.1: Model architecture. Yellow tokens denote BERT special tokens. Dotted lines indicate using BERT embeddings to seed the graph for the R-GCN.

Dataset Preprocessing

In order to simplify our dataset tasks into relation extraction, we begin by identifying tuples of (entity1, relation, entity2), where each entity refers to a span of text in the original document, and relation refers to the flow graph edge label from the dataset. We format each triple into an instance that contains the triple and its context. We consider the context to be the shortest set of contiguous sentences that span both entity text spans. To segment sentences, we use the `en_core_sci_md` model with default settings provided in SciSpacy (Neumann et al., 2019), to account for the scientific text in the MSCorpus dataset. So that our models do not learn shallow heuristics to predict relations based on entity type, as observed in Rosenman et al. (2020), we exclude the entity types from the original datasets.

Parsing

We then annotate each context entity with two linguistic representations: AMR (Banarescu et al., 2013) and dependency parses. We choose AMR primarily for the quality of parsers available relative to other semantic formalisms: AMR parsing is a relatively popular task, and state-of-the-art parsers are often exposed to scientific text in their training. However, despite the quality of parses, AMR as a formalism presents several challenges to its use in downstream applications. Foremost among these is the problem of *token alignment*: nodes and edges in AMR graphs do not have links back to the words in the text that they are generated from. As a contrast, we choose to use dependency parses as our syntactic framework, which are straightforward in their

correspondence to the original text: each node corresponds to a word.

For the dependency parses, we annotate each context span using the Stanza dependency parser (Qi et al., 2020b), which produces a dependency graph per sentence. We then create a "top" node for the graph to link the individual trees for relations that span sentences.

For the AMR parses, we use the SPRING model (Bevilacqua et al., 2021) as implemented in AMRLib [1] We additionally verified that the model did not perform significantly differently than the original implementation. In contrast to the dependency parser, we found SPRING to occasionally be brittle. Because of its sequence-to-sequence architecture which cannot enforce that the produced output is a valid parse, the model sometimes failed to produce a parse altogether. These errors were non-transient, and did not display a pattern we could discern. In the interest of evaluating the impact of off-the-shelf tools as they were, we chose to include instances without AMR parses in our datasets. Because of the brittleness of the SPRING model, we parsed sentences in the datasets individually. We then compose the graph representations of each context instance by joining the graphs of its constituent sentences. We follow the same procedure as with dependency parsing, joining all of the sentence-level AMR graphs with a top node.

AMR Alignment

Because AMR nodes are not required to point back to the tokens that generated them, extracting token-level features to incorporate into our RE model relied on the task of AMR aligment. AMR alignment is usually treated as a *post-hoc* task that relies on rule-based algorithms. We experimented with algorithms based on the common JAMR (Flanigan et al., 2014) and ISI (Pourdamghani et al., 2014) aligners. These were implemented in AMRLib as the `RBW` and `FAA` aligners, respectively. Both aligners perform poorly, especially on the scientific text in the MSCorpus dataset. Because alignments are necessary to producing token-level features from an AMR representation, we developed heuristics as a second pass of alignment after applying the FAA aligner to the original text/AMR pair. Our heuristics, developed on the training split of each of our datasets, iteratively seek out unaligned AMR triples, normalize the node labels, and compare them with words in the original sentence after lemmatization. The words are taken from SPRING's tokenization of the original sentence, and the lemmatization uses NLTK's (Bird et al., 2009) `WordNetLemmatizer` with the default parameters. We also normalize node labels to remove artifacts like Propbank sense indicators.

To measure the success of our alignment algorithm, we use a statistic that describes how many AMR triples in the graph that should be aligned (according to a combination of the AMR standard[2] and dataset-specific heuristics), are aligned to a token in the text. We also compute statistics based on how many triples contain at least one entity unaligned with the graph. With only the FAA aligner, over 59% of triples contain at least one entity without a corresponding aligned word across our three datasets. After realignment, we achieve a significantly higher rate of alignment, with just under 27% of triples having at least one entity unaligned to nodes in the graph.

[1] `https://github.com/bjascob/amrlib`
[2] https://amr.isi.edu/doc/amr-alignment-guidelines.html

Model Architectures

Baseline Model: We consider a common baseline architecture for relation extraction, based on BERT (Devlin et al., 2019b). We begin by embedding the context for each relation. We then extract the embeddings for all tokens that constitute each entity, and max-pool them into embeddings e_1 and e_2 . We concatenate e_1, e_2, and the embedding for the `[CLS]` token, which we consider a stand-in for the context, into one vector. We then pass that vector through a two-layer MLP with a $\tanh$ activation between layers, before finally applying a softmax for the classification.

Graph-aware models: To compute graph-based features, we first initialize the linguistic graph's nodes with feature vectors of the same size as the baseline BERT model's embeddings. For every aligned token, we initialize that feature vector with the max-pool of the embeddings of each of its aligned tokens, leaving the embeddings zeroed out for unaligned nodes. We then pass the graph through a relational graph convolution network (R-GCN, Schlichtkrull et al. (2018a)). We choose the R-GCN for its ability to model heterogeneous relations in graphs. After computing node embeddings, we employ a residual connection similar to the `Hier` setting shown in Figure 3a in (Bai et al., 2021), where the mean pool of node embeddings corresponding to e_1 and e_2 is added back to the BERT-based embeddings of the aligned entity tokens computed earlier. These updated embeddings are then passed to the same MLP relation classifier as in the baseline. We choose this type of residual connection for the bottleneck in representational capacity that it imposes on our models. Additionally, we measure the distribution of path lengths between entities in both frameworks in the train split of our datasets, and find that the mean path of each dataset lies between 3 and 4. We thus use an R-GCN of depth 4 for all experiments in order to capture most paths. Because of the residual connection architecture, we are restricted to using the baseline BERT model's word embedding size as the node embedding size as well. Combined with the GNN depth of 4, our model adds significantly more parameters — 203M parameters vs the plaintext model's 111M. However, we hypothesize that being forced to operate in the same embedding space as the baseline will discourage models from memorizing the original dataset and overfitting, especially in the few-shot setting.

We depict our architecture in figure 4.1. The baseline architecture omits the right-hand fork, using only BERT embeddings.

4.1.2 Experiments

In-Domain Experiments

We train both the baseline and graph-aware models on each dataset, using the train/dev/test splits where provided. If no dev split was provided, we randomly split the training dataset 80/20 into new train and dev splits. We use `bert-base-uncased` as available on the Huggingface Hub [3] as our base BERT model, for both the baseline and graph-aware variants. For our graph-aware variants, we use R-GCN as our graph network. We train each model with the Adam optimizer (Kingma and Ba, 2014) to minimize the cross-entropy loss between predicted and true labels. We use a learning rate of 2×10^{-5} and a batch size of 16. Each model is trained on 3 random seeds

[3] `https://huggingface.co/bert-base-uncased`

for 30 epochs, using early stopping criterion based on the macro-averaged F1 score on the dev split with a patience of 5 epochs. We keep the model that performs best on the dev split, and calculate its corresponding macro F1 score on the test set. We refer to the graph aware models that add dependencies and AMRs as +Dep and +AMR, respectively.

Few-shot Experiments

We formulate few-shot transfer learning as an N-way K-shot problem, where a model is trained on K instances of each of the N classes in the target domain. We experiment with $K \in \{1, 5, 10, 20, 50, 100\}$. Because of the label imbalance in our datasets, where K is greater than the number of labeled examples for a given class, we sample all of the labeled instances without replacement. This can result in fewer than K examples for a given class.

For the transfer process, we begin with the models trained in the in-domain experiments, and replace the MLP classification head with a freshly initialized head with a suitable number of outputs for the target domain's number of classes. We reuse the BERT and R-GCN components of the in-domain model, and allow their weights to be updated in the transfer finetuning.

We continue to train each model using the same settings as in-domain training using a batch size of 4, sampling each dataset three times with different seeds.

In addition, to control for the effects of the source and target dataset interactions and our sampling strategies, we train few-shot models in each domain from scratch, for each of the settings K described above, using the same settings.

All of our experiments were run on NVIDIA A4500 GPUs, and we used roughly 33 days of GPU time for all of the experiments in this project, including hyperparameter tuning.

4.1.3 Results and Discussion

We expect that more powerful linguistic representations than plain text will aid in few shot transfer between domains. In order for few shot transfer to be successful, the target data points used for transfer need to increase the relevant shared representation between the source and target datasets. Because of this, we expect that any effect of representation on test set performance will depend upon how much shared representation there was between the two domains to begin with and how much the few added examples closes the gap. A more efficient representation may lose its advantage once there are enough target domain examples to obviate the need for efficiency. In this section, we aim to answer a number of questions.

[RQ:1] Do linguistic representation aid in either in-domain or cross-domain transfer? We present our in-domain results on the complete datasets in table 4.1. Overall, we do not see significant differences between the baseline and +Dep and +AMR cases, even though they appear to overperform the baseline case on RISeC and MSCorpus. Notably, however, these models do not overfit more than the baseline: performance on the unseen test set remains similar.

We do, however, see differences in performance between the baseline and graph-aware cases in the few-shot transfer setting. In Figure 4.2, we visualize the difference between the macro-averaged F1 performance in each of our graph-aware cases and the baseline against the few-shot setting. We see that while in the 1-shot case, our results are highly variable, the 5-, 10-, and

Dataset	Case	Mean (std)
EFGC	+AMR	83.9 (0.3)
	+Dep	84.6 (1.3)
	Baseline	85.0 (0.8)
MSCorpus	+AMR	87.8 (1.0)
	+Dep	88.4 (0.5)
	Baseline	87.5 (0.5)
RISeC	+AMR	82.8 (1.6)
	+Dep	81.7 (2.1)
	Baseline	82.7 (1.3)

Table 4.1: Results from in-domain experiments. Each value represents the mean of runs with three random seeds, with standard deviation in parentheses.

20- shot cases yield noticeable improvement, peaking in the 5- and 10-shot settings. In our best-performing results, we see a 6-point absolute gain in F1 score.

We find that both dependency parse and AMR representations show a statistically significant positive effect on performance. In particular, we test the significance of the effect with an ANOVA model with multiple independent variables: namely, source and target dataset (EFGC, RISeC, MSCorpus), representation case (Baseline, +Dep, +AMR), few-shot setting (1, 5, 10, 20, 50, 100), and transfer setting (in-domain vs out-of-domain). The dependent variable is test set F1. The data table for the analysis includes 3 runs for each combination of variables each with a separate random seed. We train our models under a full-factorial experimental design, i.e. we ran trials for all combinations of variables. This design allows us to test the reliability of the effect of our variables under a variety of conditions while making the necessary statistical adjustments to avoid spurious significant effects that may occur when multiple statistical comparisons are made. We use this design rather than pairwise significance tests so that we can measure the effect of introducing linguistic formalisms as a whole, rather than arguing the statistical significance of individual, pairwise comparisons.

We expect that the similarity between source and target datasets, the variation in the target dataset, and the few-shot setting could all either dampen or magnify any effect of representation on the performance. We therefore include pairwise interaction terms in the ANOVA model for case by source dataset, case by target dataset, case by transfer setting, and case by few-shot setting. The examples added for the few shot setting in the transfer case are sampled from the training split of the target dataset. Thus, while we expect for the cross-domain case the few shot setting has an effect, we do not expect an effect in the in-domain case, since the target domain examples added to the training data simply replicate examples that were already part of the dataset. To account for this, we include a final interaction term between few shot setting and transfer setting in the ANOVA model.

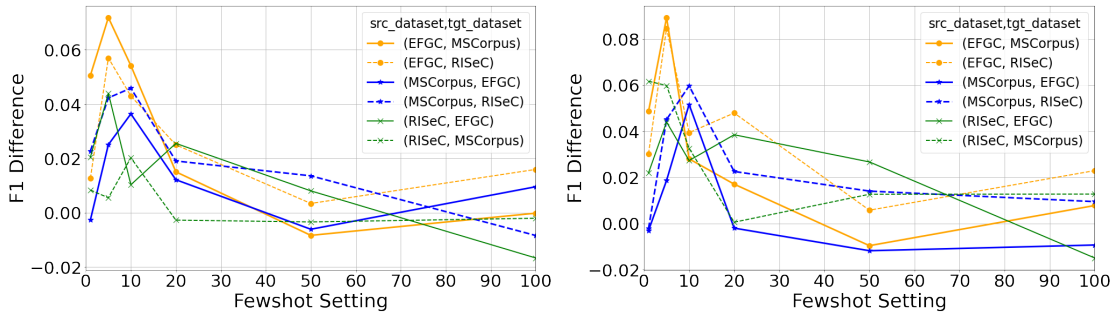
The ANOVA model explains 98% of the variation in F1 scores. The results align well with our intuitions. First, as expected we find a significant effect of transfer setting such that in-domain performance on the entire dataset is better than transfer performance in a few-shot setting: F(1,

Target	Source	Case	Fewshot Setting 1	5	10	20	50	100
RISeC	From Scratch	Baseline	18.6 (2.9)	36.5 (3.2)	48.3 (3.1)	60.2 (2.3)	**71.1 (1.1)**	76.9 (0.2)
		+Dep	19.3 (4.5)	**40.0 (2.8)**	51.5 (3.2)	**62.7 (4.5)**	71.1 (0.9)	**79.5 (1.5)**
		+AMR	**19.8 (7.1)**	39.3 (5.2)	**52.1 (2.6)**	60.9 (4.0)	70.6 (0.7)	78.4 (1.0)
	MSCorpus	Baseline	19.7 (5.5)	35.1 (5.4)	45.6 (0.8)	57.7 (0.9)	67.8 (1.3)	76.2 (1.6)
		+Dep	19.4 (2.1)	**39.7 (5.2)**	**51.6 (1.1)**	**60.0 (4.8)**	**69.2 (1.9)**	**77.2 (3.4)**
		+AMR	**21.9 (2.6)**	39.4 (4.2)	50.2 (0.9)	59.6 (0.9)	69.2 (2.2)	75.4 (1.3)
	EFGC	Baseline	25.8 (5.0)	42.0 (4.0)	53.7 (0.6)	61.8 (3.3)	71.1 (1.5)	75.2 (0.9)
		+Dep	**28.8 (7.7)**	**50.5 (3.9)**	57.6 (2.4)	**66.6 (0.9)**	**71.7 (1.2)**	**77.5 (0.8)**
		+AMR	27.0 (7.6)	47.7 (8.9)	**58.0 (2.6)**	64.3 (1.2)	71.5 (0.4)	76.8 (2.1)
MSCorpus	From Scratch	Baseline	25.0 (4.9)	46.9 (2.7)	63.4 (1.0)	**74.0 (1.1)**	82.7 (1.2)	82.6 (1.9)
		+Dep	**30.6 (2.8)**	**49.5 (1.0)**	**66.0 (3.2)**	72.7 (2.3)	**82.7 (0.9)**	**84.8 (0.3)**
		+AMR	26.7 (4.3)	45.3 (0.9)	62.4 (3.1)	72.6 (2.0)	82.2 (1.2)	84.3 (1.0)
	RISeC	Baseline	24.4 (2.2)	43.4 (2.5)	56.5 (3.3)	69.8 (1.3)	81.4 (0.9)	83.7 (0.6)
		+Dep	**30.6 (0.5)**	**49.4 (3.5)**	**59.8 (3.9)**	**69.9 (4.2)**	**82.6 (1.0)**	**85.0 (1.4)**
		+AMR	25.3 (3.1)	43.9 (3.4)	58.5 (4.9)	69.5 (2.2)	81.0 (1.0)	83.5 (1.5)
	EFGC	Baseline	26.9 (4.6)	46.6 (2.1)	63.8 (3.0)	72.5 (0.9)	**81.5 (0.9)**	83.6 (1.8)
		+Dep	31.7 (4.0)	**55.5 (5.6)**	66.6 (4.6)	**74.2 (2.5)**	80.5 (3.0)	**84.4 (1.1)**
		+AMR	**31.9 (3.8)**	53.8 (6.1)	**69.3 (0.8)**	74.0 (3.3)	80.7 (1.2)	83.6 (2.1)
EFGC	From Scratch	Baseline	16.2 (1.5)	29.3 (2.3)	38.9 (1.8)	47.6 (1.2)	61.0 (0.9)	63.8 (3.0)
		+Dep	**17.2 (4.1)**	30.3 (3.8)	**40.7 (2.5)**	48.6 (1.1)	60.2 (1.8)	**66.7 (2.4)**
		+AMR	14.2 (2.3)	**30.8 (2.2)**	39.9 (3.3)	**48.7 (1.1)**	**61.1 (2.1)**	64.1 (3.2)
	RISeC	Baseline	16.0 (1.7)	30.4 (3.0)	35.7 (0.4)	44.7 (1.5)	56.9 (1.2)	**65.8 (1.6)**
		+Dep	**18.2 (4.5)**	**34.8 (3.0)**	**38.4 (3.2)**	**48.6 (1.3)**	**59.5 (1.6)**	64.3 (2.9)
		+AMR	18.1 (1.5)	34.8 (1.4)	36.7 (1.5)	47.3 (2.4)	57.7 (2.7)	64.1 (2.9)
	MSCorpus	Baseline	**17.4 (4.4)**	29.5 (2.9)	39.7 (2.8)	49.2 (0.5)	**61.2 (1.0)**	64.4 (1.0)
		+Dep	17.0 (3.8)	31.4 (2.2)	**44.9 (1.9)**	49.0 (1.2)	60.0 (0.6)	63.5 (3.4)
		+AMR	17.1 (2.4)	**32.0 (0.2)**	43.4 (2.4)	**50.4 (2.6)**	60.6 (0.6)	**65.4 (3.7)**

Table 4.2: Few-shot learning results. "From Scratch" in the source column represents the case where we train a few-shot model from scratch, without transfer. Each cell represents the mean macro-F1 across three random seeds, with the standard deviation of those runs in parentheses. We group our results by the target dataset first to allow easier comparison of the impact of source datasets. Bold results represent the best case for a source-target pair.

679) = 10356.25, $p \leq .0001$. In these cases, the original dataset for in-domain training is between 5 and 15 times the size of the target training dataset. We also find a significant effect of the few-shot setting, such that larger numbers of target domain examples are associated with higher performance, $F(5, 679) = 716.79$, $p \leq .0001$. A post-hoc student-t analysis reveals that all pairwise comparisons are significant. Notably, there is a significant interaction between transfer setting and few shot setting: $F(5, 679) = 733.83$, $p \leq .0001$, such that the effect of the few-shot setting is restricted to the transfer setting, as expected.

Our hypothesis is primarily related to the importance of the representation of the data for efficiently enabling transfer between domains. We find a significant effect of representation case: $F(2, 679) = 5.26$, $p \leq .01$. **A student-t post-hoc analysis reveals that both +Dep and +AMR cases are better than plain text, but there is no significant difference between the two.** There is also a significant interaction between representation case and transfer setting: $F(2, 679) = 8.19$,



(a) +AMR vs baseline

(b) +Dep vs baseline

Figure 4.2: Differences in F1 over baseline from incorporating linguistic graphs in models.

$p \leq .0005$. In particular, the effect of case is only significant in the transfer setting. There is also a significant interaction between few shot setting and case: $F(2, 679) = 8.19$, $p \leq .0005$. A student-t post-hoc analysis reveals that the effect is only significant for the 5- and 10-shot settings. Thus, **1 target example is too small to yield a significant effect whereas 20 or more is too many such that the representational advantage disappears**. We also find a significant interaction between representation case and target dataset, but not with source dataset: $F(4, 679) = 2.61$, $p \leq .05$, such that the effect of representation is significant for RISeC and MSCorpus but not for EFGC.

We present all of our few-shot results in Table 4.2. Significance testing was performed on the difference in results between the baseline and lingusitic representation cases in the transfer setting. Additionally, we investigate the impact of source domain on the utility of linguistic representations. We therefore compare results between models trained in a few-shot setting from scratch, seeing only one dataset, with the transfer model that we train on a source dataset first. We show both of these cases in table 4.2, with few-shot models trained from scratch denoted in the source dataset column as "From Scratch" results.

[RQ:2]How important is the choice of the source domain on the transfer performance? We see several interesting patterns in our 5- and 10-shot results when we take our few-shot models trained from scratch into account. We visualizes differences in performance between the from-scratch models and models trained with a different source domain in table 4.3. We find that the transfer between datasets for our text-only models is of limited utility, if not outright harmful. While we see one instance (the EFGC to RISeC transfer) in which introducing a transfer source dataset improves the baseline model's performance on the target dataset consistently, we see more commonly that adding a transfer source dataset makes only a small difference, or even hurts the performance of the baseline model. In the cases of transfer between MSCorpus and RISeC in either direction, for instance, the baseline model in the transfer setting consistently underperforms the model trained from scratch by up to 7 F1 points, and does not close that gap even in the 50- and 100-shot settings. However, incorporating linguistic formalisms proves to be far more robust to the choice of source domain: the linguistic representations, regardless of source domain are never worse than the baseline trained on that source domain, and still frequently outperform the baseline trained from scratch, even when the choice of source domain imposes a performance penalty.

Interestingly, an intuitive notion of "domain distance" fails to explain when transfer will be

			Fewshot Setting	
Target	**Source**	**Case**	5	10
RISeC	MSCorpus	Baseline	-1.37	-2.66
		+Dep	-0.29	0.04
		+AMR	0.10	-1.92
	EFGC	Baseline	5.50	5.42
		+Dep	10.53	6.09
		+AMR	8.44	5.87
MSCorpus	RISeC	Baseline	-3.55	-6.92
		+Dep	-0.10	-6.24
		+AMR	-1.40	-3.91
	EFGC	Baseline	-0.33	0.41
		+Dep	6.06	0.63
		+AMR	8.45	6.82
EFGC	RISeC	Baseline	1.12	-3.23
		+Dep	4.51	-2.34
		+AMR	4.02	-3.23
	MSCorpus	Baseline	0.29	0.85
		+Dep	1.14	4.18
		+AMR	1.29	3.46

Table 4.3: Differences from baseline model trained from scratch in the 5- and 10-shot cases gained in using a different source domain. Linguistic representations are more robust to choice of source domain.

helpful. EFGC and RISeC both come from the cooking domain, but though RISeC and MSCorpus negatively influence each other in transfer, MSCorpus and EFGC in the baseline case have very little difference from the transfer case. Transfer between the abstract categories of ”cooking” dataset and ”materials science” dataset is highly variable.

Notably, we observe that the benefits we derive from transfer seem asymmetrical: even datasets that transfer well in one direction might not in the other direction. We see markedly better results transferring from EFGC to RISeC, for instance, than we see in the reverse direction, and we see a similar result (though less consistent) for transfer from EFGC to MSCorpus as compared to the reverse.

[RQ:3]What is the impact of linguistic structure on the performance of few-shot RE in-domain? When factoring in the effect of our graph-aware models, we see that they help models generalize, both in the few-shot in-domain setting, as well as the transfer setting. **Where transfer itself causes the performance of the baseline model to degrade, however, we see that the addition of linguistic representations sometimes makes up for that gap almost entirely**. In the case of the 10-shot MSCorpus to RISeC transfer, we see that the baseline transfer model performs an average 2.7 points worse than the baseline from-scratch model (48.5 vs. 45.3), but that the dependency models perform very similarly (51.5 vs. 51.6). In cases where the transfer pairs are well-matched, however, we see that while the baseline results remain similar, the benefit that the models derive from the linguistic representations is much more pronounced in the transfer setting. In the 10-shot transfer in both directions between EFGC and MSCorpus, as well as the

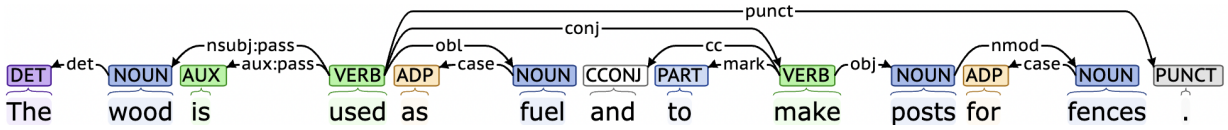


Figure 4.3: Example depicting the supplemental information provided by the *dependency tree*. The entities of interest are wood and fences, having the relationship material_used. The path *wood* ← *used* → *make* → *posts* → *fences* elicits this relationship.

EFCG to RISeC case, transfer models that incorporate dependencies and AMRs overperform their in-domain counterparts by between 3 and 7 points.

4.2 Multilingual Relation Extraction

We deal with the task of multi-lingual relation extraction, wherein we identify the nature of relationship between two annotated entities in a document. We show in Figure 4.3 how we can connect the entities wood and fences by traversing the dependency graph that connects these two entities, highlighting the potential utility of linguistic frameworks for this task. We explore the role of dependency parses for cross-lingual relation extraction in both a fine-tuned supervised setting and a prompting/ in-context learning setup.

Our work is motivated on the past work of Sachan et al. (2021) which showed the utility of adding syntactic information for different information extraction tasks in English. However, the observed benefits hold true only when the **gold parses** are available, with no improvements over the baseline in presence of off-the-shelf parses. In this study we expand upon this idea and investigate whether off-the-shelf dependency parses can assist language models in multi-lingual information extraction for both indomain and zero-shot transfer settings. Moreover, as we usher into an era of large language models, the question which looms over our head like the proverbial sword of Damocles "Are dependency parses helpful for information extraction in in-context learning scenarios?"

We propose a framework, DEPGEN, built on top of a pretrained multi-lingual language model that uses dependency parse information to perform relation extraction for both in-domain and zero-shot cross-lingual transfer settings. Through a comprehensive set of 2440 experiments spanning 10 languages over 2 datasets, we observe that incorporating dependency information brings about modest improvements for in-domain and cross-lingual fine-tuning setups by 0.9% and 1.5% respectively.

We also carry out extensive statistical analysis to identify which factors significantly impact performance. Our observations highlight that performance improvements is mostly predicated by the choice of the target language, and the choice of the pre-trained language model rather than the choice of the dependency parser for all cases. However, for the in-context learning setup, we demonstrate that the performance is determined by the choice of the prompting strategy, with our proposed approach boasting the highest gains, i.e. an absolute improvement of 1.67 F1 score over the baseline.

4.2.1 Methodology

We investigate the role of dependency parses for zero-shot cross-lingual relation extraction in two setups, namely (i) a fine-tuned setup where a model is first trained on a given source language and then evaluated on a target language, and (ii) an in-context-learning setup where we prompt an LLM to predict the relation between two specified entities in a zero-shot setting to test the innate capabilities of the LLM for RE.

Fine-Tuning Setting

We present a detailed description of our proposed framework, DEPGEN here. Our framework leverages the internal structure of a document text to aid relation classification. We define internal structure as the linguistic information encoded within the document based on syntactic rules in the form of dependency parses. This section describes the individual components that constitute our framework DEPGEN, namely the multilingual encoder, dependency parser, graph neural network, and the fusion layer. We dive deep into the methodology for representing the textual content, and elaborate on the approach employed for incorporating dependency parses for a given input sentence. Finally, we end the section with how the different modes of information are fused, and the classification setup. A pictorial representation of our framework can be seen in Figure 4.4 Our architecture involves the following components.

Multilingual Encoder: We experiment with mBERT (Devlin et al., 2018) and XLMR (Conneau et al., 2020b) as our multilingual text encoder to obtain representations of the input sentence(s). Past work has shown the efficacy of such contextual multilingual encoders in capturing long-range semantic dependency in text (Litschko et al., 2021). Similar to these works, we consider the final encoder layer representation of [CLS] token as the text representation. The sentence(s) are fed as input to the MULTILINGUAL ENCODER (Figure 4.4) and the [CLS] token representation from the final layer is fed into the FUSION LAYER. The individual token representations from the final layer are used to initialize the node embeddings in the dependency graph of the INTERNAL STRUCTURE module, which we describe below.

Internal Structure: We incorporate the internal structure information by learning the syntactic dependency information between the tokens in the input sentence. We first pass the input tokens through a DEPENDENCY PARSER to obtain the dependency tree for each sentence. We then construct a dependency graph from the constituent dependency trees, which is then fed as input to a Graph Neural Network (GNN) (Scarselli et al., 2008). The various components of this module are as follows.

- **Dependency Parser:** To generate the dependency tree, we use off-the-shelf multilingual dependency parsing modules, i.e. Stanza (Qi et al., 2020b) and Trankit (Nguyen et al., 2021b). The resulting dependency tree represents the syntactic dependency relations between the words in a sentence; the dependencies follow the Universal Dependencies formalism (Nivre et al., 2016; Zeman et al., 2019), resulting in 76 types of dependencies across the different languages for our experiments.
- **Dependency Graph:** Since the dependency tree is defined for a sentence, the output from DEPENDENCY PARSER will be in the form of a forest of disconnected dependency trees; for example 4 trees for 4 sentences in Figure 4.4. We add a pseudo node [CENTRAL] and

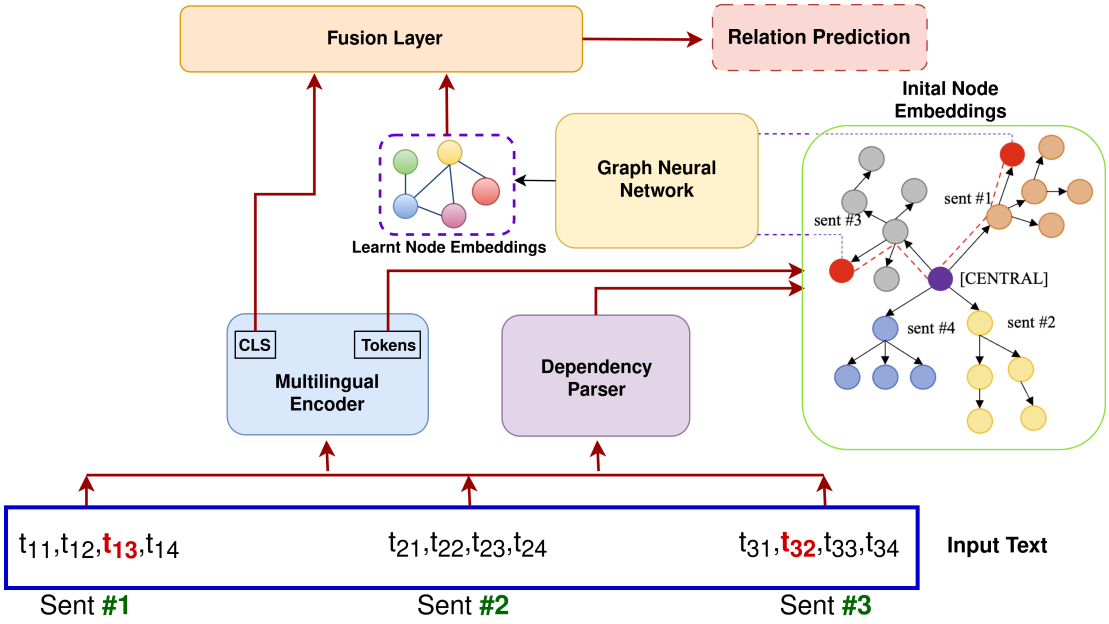


Figure 4.4: An overview of our proposed framework DEPGEN. The architecture takes as input a document, which comprises a sequence of sentences, with the entities highlighted in red. This document passes through a multilingual encoder to obtain the token embeddings, and a dependency parser that generates dependency parses for each sentence. The individual sentences in the dependency parser are connected using a central [CENTRAL] node to obtain a connected graph. The nodes are initialized using the embeddings obtained from the multilingual encoder and updated using a Graph Neural Network. The final representations of the entities obtained from the GNN are fused with the entity embeddings and concatenated with the [CLS] token of the document to predict the relation.

add a new type of dependency relation [SENT] between the [CENTRAL] and all the [ROOT] nodes of the sentences. The proposed design has two benefits - (1) The [CENTRAL] node allows for information exchange between the sentences, which otherwise would probably lead to different clusters of representations (represented by colors in Figure 4.4) for nodes in different sentences, (2) The distance between the two entities is reduced (dotted red line in Figure 4.4) when the entities are present across two different sentences, resulting in an efficient information flow between them.

- **Graph Neural Network:** We represent each word as a node in the dependency graph and the dependency relations as the edges between the nodes. Each node in the graph is initialized with the representations obtained from the final layer of the MULTILINGUAL ENCODER. We aggregate the sub-token representations via max-pooling and obtain the final representation of a word. This initialization helps incorporate the semantic relationship between the nodes and facilitates end-to-end joint training of the MULTILINGUAL ENCODER and the INTERNAL STRUCTURE modules. The relation embeddings for the all the relation types are initialized at random and learnt jointly along with the node embeddings. The representations of the two entities from the multi-layer GNN are then fed to the FUSION LAYER along with

the sentence representation for relation prediction.

Relation Prediction: We concatenate the representations obtained from the MULTILINGUAL ENCODER and the INTERNAL STRUCTURE modules in the FUSION LAYER and perform a multi-class classification for predicting the relation. During training, we compute the standard Cross Entropy loss, and back-propagate it jointly through all the components of the network.

In-context Learning Setting

In addition to the DEPGEN framework that encapsulates the fine-tuned setting, we also explore the role of dependency parses when provided as additional inputs to LLMs in a zero-shot prompting setup. We experiment with three different types of prompt formats that encodes the dependency information which we describe below.

Tuple Format: In the tuple-based prompt format, we simply provide the dependency parse as a list of tuples or dictionary keys. Each tuple comprises three elements, i.e. a node in the dependency graph or a word, the corresponding head node of that word, and the relation that connects the head node to the word. For example, the phrase "Porsche Panamera", would have the following information in the form of a tuple.

```
{
    word: Porsche
    head: Panamera,
    rel: compound
}
```

Text Format: Instead of providing the dependency parse information in the form of tuples, we verbalize the dependency relations between the words in the sentence in natural language format. In the above example of "Porsche Panamera", we re-write the tuple information as ``Porsche is Compound noun modifier of Panamera''. We do this for all the tuples in the dependency graph.

Filtered Text Format: As opposed to verbalizing all the tuples in the dependency graph, we filter out only the tuples that connect the two entities in the sentence via the dependency relations. Not only does this reduce the number of input tokens to the LLM, it also helps filter out redundant information.

As a control, we also prompt the models with only the text, without any dependency information, which serves as a baseline. The details of the prompts are in the Appendix.

4.2.2 Experimental Setup

Fine-tuned Experimental Setup: We experiment with the following settings:

1. **Baseline:** We experiment with mBERT (Devlin et al., 2019c) and XLMR (Conneau et al., 2020a) as our choices to encode the document text and the entity spans. We concatenate the pooled representation of the entities and the [CLS] embedding and use it for relation classification.

		RedFM					IndoRE		
					mBERT				
DEP	GNN	en	es	fr	it	de	en	hi	te
-	-	84.3±0.7	80.0±0.6	78.6±0.3	76.3±0.8	78.7±0.3	94.3±0.6	89.6±0.4	84.9±0.4
stanza	rgcn	**85.7±0.8**	80.5±1.0	79.7±1.0	78.2±0.5	80.0±0.9	94.4±0.2	**90.9±0.3**	**86.1±0.9**
stanza	rgat	85.2±1.4	**82.2±0.6**	79.9±0.4	77.9±1.2	**80.5±0.6**	**94.9±0.3**	89.5±1.4	85.9±1.1
trankit	rgcn	84.3±0.4	81.8±0.8	**80.7±0.8**	**78.9±0.7**	79.7±0.9	94.0±0.2	89.7±0.1	85.9±1.9
trankit	rgat	85.5±1.3	80.9±0.3	80.2±0.2	77.3±0.8	78.9±0.7	94.1±0.5	88.9±0.5	84.6±0.8
					XLMR				
-	-	**84.0±1.1**	77.2±2.0	76.2±1.0	74.8±1.2	75.2±0.6	92.1±0.8	88.7±0.9	**86.3±1.1**
stanza	rgcn	83.7±0.6	76.8±0.8	**76.7±0.9**	73.3±0.7	75.7±1.5	91.8±0.8	89.6±1.1	85.6±0.7
stanza	rgat	**84.0±0.8**	**77.5±1.4**	74.4±0.9	**75.6±1.2**	**76.2±1.1**	92.2±0.4	**89.9±0.9**	85.7±0.6
trankit	rgcn	83.8±0.5	76.4±1.1	74.7±1.0	72.6±2.3	73.9±2.6	91.9±0.9	**89.9±0.8**	85.2±0.5
trankit	rgat	82.6±0.8	77.3±0.2	75.0±0.3	74.0±1.7	75.9±0.1	**92.6±0.7**	89.2±1.0	85.9±1.6

Table 4.4: In-domain RE performance of mBERT and XLMR on RedFM and IndoRE, with dependency information (i.e. choice of the parser or DEP, and the choice of the GNN used to encode the information, i.e. GNN). Results are averaged across the top 3 seeds, with the highest values in each column bolded.

2. **Graph Neural Network:** We experiment with RGCN (Schlichtkrull et al., 2018b) and RGAT (Busbridge et al., 2019) as the backbone GNN architecture to encode the dependency information between words in the document. We use a GNN with 2 hidden layers for all our experiments.

In-context Learning Experimental Setup: We employ three different instruction-tuned LLMs for our in-context learning experiments, i.e. LLaMA (Meta-Llama-3-8B-Instruct) (Grattafiori et al., 2024), Mistral (Mistral-7B-Instruct-v0.3) (Jiang et al., 2023) and Qwen (Qwen2-7B-Instruct) (Yang et al., 2024). We use instruction-tuned LLMs since we wish to employ these LLMs in a zero-shot setup for relation extraction without fine-tuning or additional training. Similar to the fine-tuned experimental setup, the dependency parse information are obtained from two sources, i.e. Stanza and Trankit.

4.2.3 Results and Insights

In this section, we pose the following research questions (RQs) and attempt to answer the same.

RQ1. Impact of dependency parses on RE for indomain and cross-lingual transfer ?

We report the in-domain and cross-lingual relation extraction performance with mBERT and XLMR as the multilingual encoders, stanza and trankit as the choice of the off-shelf-parsers, and RGCN and RGAT being the backbone GNN for both the IndoRE and RedFM datasets, in Tables 4.4 and 4.5 respectively.

At the outset, we observe that across both datasets, adding dependency information generally improves performance over the baseline in the in-domain setting; we see higher gains when we have mBERT as the MLM as opposed to XLMR. We also observe that the gains are higher for

		RedFM							IndoRE		
						mBERT					
DEP	GNN	en	es	fr	it	de	ar	zh	en	hi	te
-	-	77.5±1.1	81.0±1.1	78.8±1.1	76.7±1.1	75.6±1.1	72.5±1.1	70.0±1.1	57.5±1.8	57.6±2.7	42.4±2.4
stanza	rgcn	78.2±0.8	81.0±0.8	**79.5±0.8**	**76.8±0.8**	77.1±0.8	72.6±0.8	70.0±0.8	57.0±1.0	57.1±0.8	44.6±1.2
stanza	rgat	78.0±1.0	81.1±1.0	78.8±1.0	76.5±1.0	77.2±1.0	**73.2±1.0**	**70.4±1.0**	56.4±1.2	57.7±1.2	**45.2±1.4**
trankit	rgcn	**78.7±0.8**	**81.3±0.8**	79.3±0.8	75.4±0.8	77.8±0.8	72.8±0.8	70.0±0.8	**57.9±0.8**	**59.1±0.6**	44.9±1.6
trankit	rgat	77.9±0.8	80.6±0.8	79.1±0.8	76.3±0.8	**77.9±0.8**	73.1±0.8	**70.4±0.8**	57.1±1.4	57.9±1.8	45.1±1.7
						XLMR					
-	-	72.7±1.4	74.2±1.4	72.2±1.4	66.8±1.4	70.7±1.4	61.8±1.4	63.1±1.4	50.0±2.2	55.1±1.5	45.9±1.6
stanza	rgcn	**73.4±1.4**	74.5±1.4	73.2±1.4	67.7±1.4	70.3±1.4	61.2±1.4	**63.9±1.4**	49.3±1.8	55.4±1.4	46.1±1.7
stanza	rgat	73.3±1.5	74.3±1.5	**73.4±1.5**	**67.9±1.5**	68.4±1.5	61.1±1.5	63.2±1.5	50.0±1.6	53.8±2.8	46.3±2.0
trankit	rgcn	73.1±1.3	74.7±1.3	73.1±1.3	66.8±1.3	69.5±1.3	**62.7±1.3**	63.8±1.3	50.7±0.7	**56.3±1.1**	45.5±2.9
trankit	rgat	73.1±1.1	**75.7±1.1**	**73.4±1.1**	65.9±1.1	**70.9±1.1**	62.1±1.1	63.6±1.1	**50.8±1.4**	56.0±2.2	**46.9±2.6**

Table 4.5: Zero-shot Cross-lingual RE performance on RedFM and IndoRE with mBERT and XLMR as the multilingual encoders with different combinations of dependency information. For a given target language, we average the performance across the different source languages. The highest values in each column are highlighted in bold. Detailed individual cross-lingual performance metrics are given in the Appendix.

the REDFM dataset than IndoRE, possibly due to the poorer quality of dependency parses on low-resource languages like Hindi and Telugu, as opposed to standard high-resource cases like English, Spanish, and Italian. In fact, for all languages other than English, we see a consistent improvement in F1-score of approximately 2.0% and 1.0% with the mBERT model on the REDFM and IndoRE dataset respectively, for the best combination of dependency parser and GNN.

In the zero-shot cross-lingual transfer scenario from Table 4.5 we observe trends that are markedly different from the in-domain setting. Each entry in this Table is computed by averaging the macro-F1 score over the other source languages, apart from itself, for the top 3 seeds. We notice only slight improvements in RE performance for mBERT but higher gains for XLMR. We hypothesize that since XLMR has a worse performance than mBERT, it benefits more from the dependency information in the zero-shot setting. In a similar vein, we observe much higher gains for Hindi and Telugu (around 2.6% and 6.6% relative performance improvements respectively) in the zero-shot setting for mBERT. The markedly lower scores in IndoRE in the zero-shot transfer setup as compared to REDFM can be attributed to the higher number of relations in the dataset (32 for IndoRE vs 51 for REDFM).

RQ2. Which scenarios benefit the most with additional information in the fine-tuned setup?

In the fine-tuned setup, we analyze which scenarios or inputs benefit the most from including dependency information. We thus group the test instances according to three different dimensions, i.e. (1) input sentence length (2) lexical distance between two entities in the sentence and (3) dependency path length. Figures 4.5 and 4.6 show the effect of these components for the in-domain and zero-shot cross-lingual transfer settings for the IndoRE and RedFM datasets respectively. The blue, orange and green plots reflect the bottom quartile, inter-quartile range and the top quartile respectively for each of these three dimensions.

- **Sentence Length:** We quantify sentence length based on the total number of tokens in

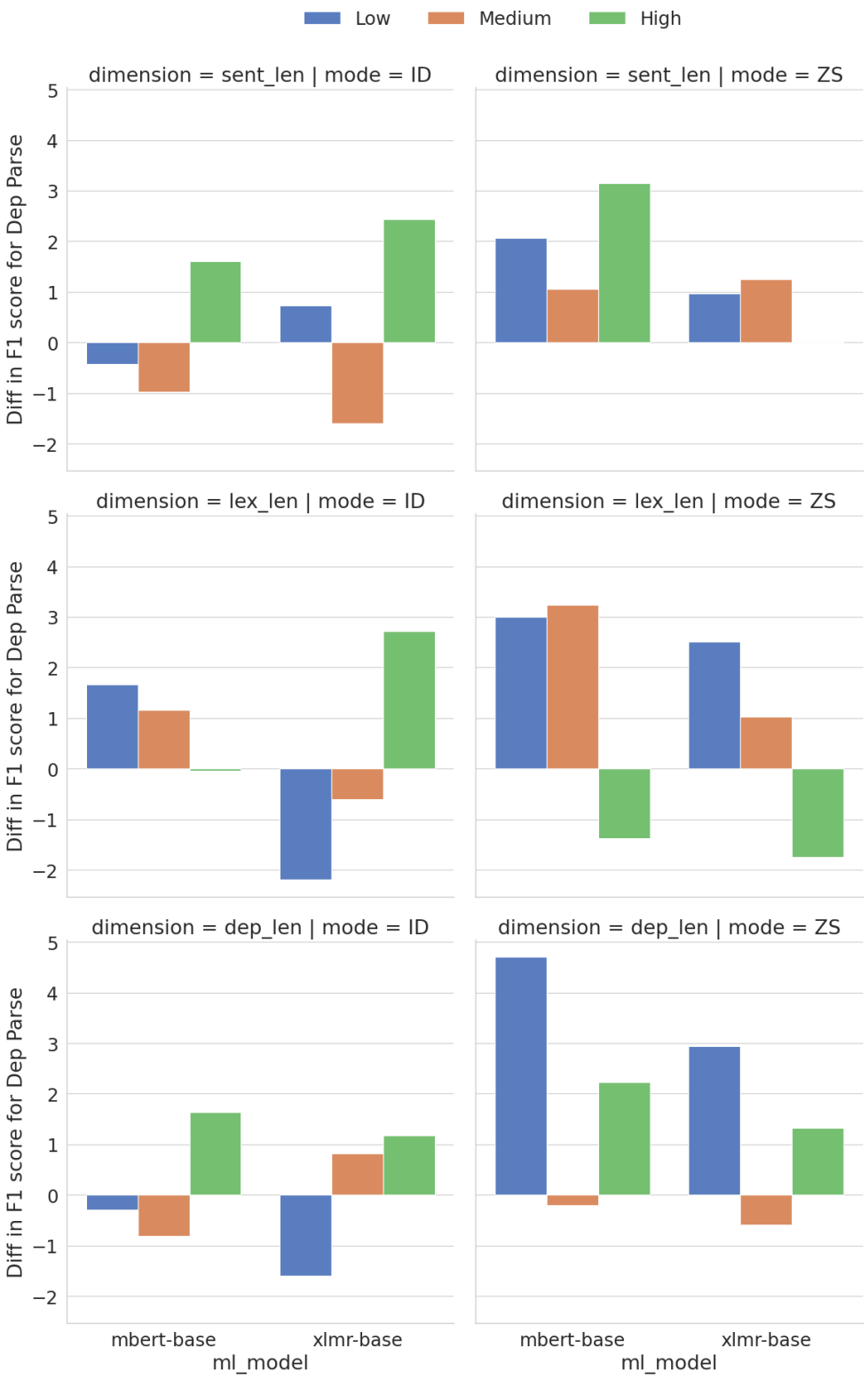


Figure 4.5: Performance of DEPGEN for in-domain and zero-shot cross-lingual transfer settings on the IndoRE dataset analyzed across variations in sentence, lexical and dependency length

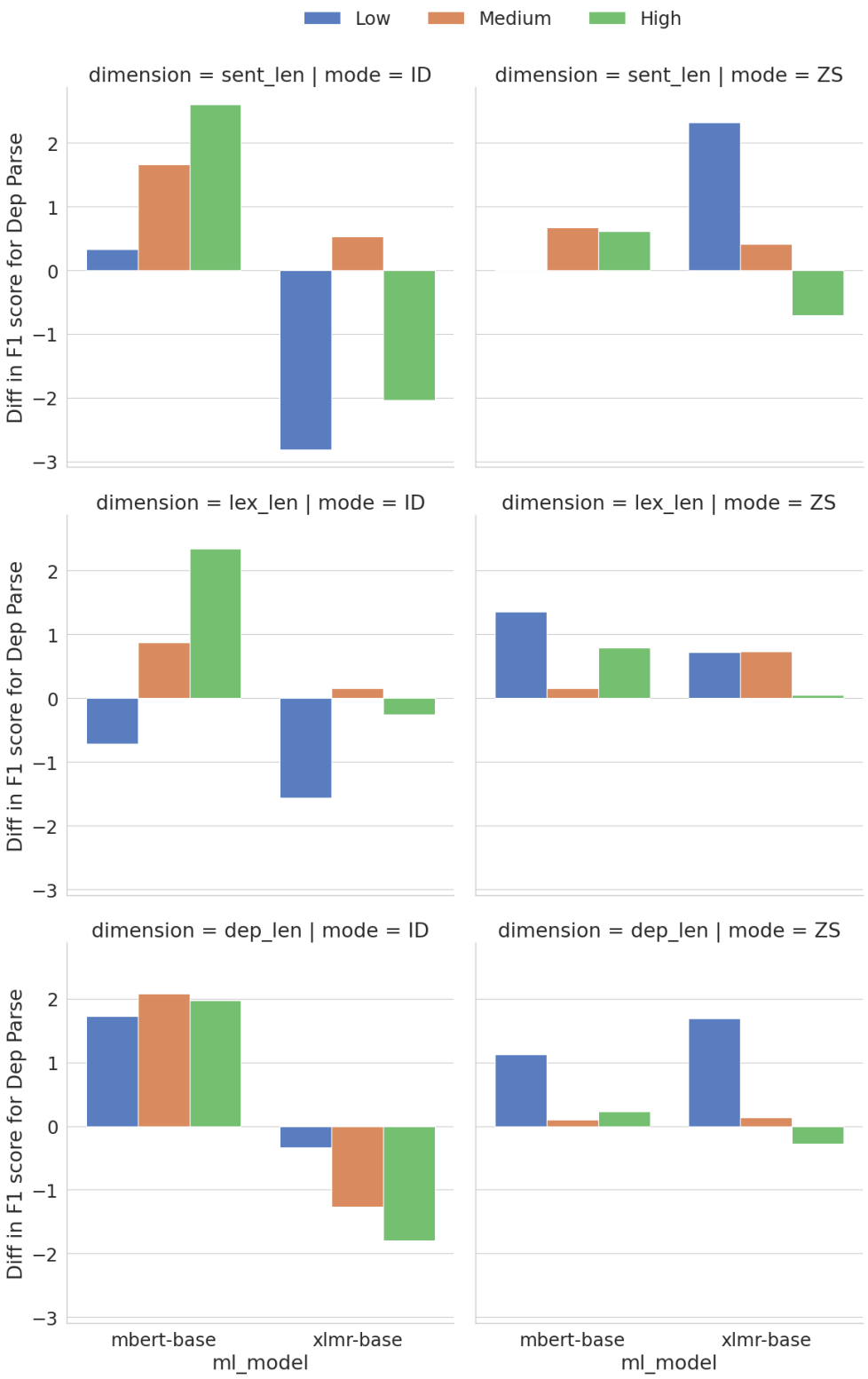


Figure 4.6: Performance of DEPGEN for in-domain and zero-shot cross-lingual transfer settings on the RedFM dataset analyzed across variations in sentence, lexical and dependency length

Model	Parser	Prompting	RedFM							IndoRE			Average
			ar	de	en	es	fr	it	zh	en	hi	te	
Llama	None	-	25.6	25.7	27.0	27.0	16.7	**36.7**	**37.1**	47.6	39.0	21.9	30.4
	Stanza	Tuple	24.3	19.3	23.9	17.0	18.4	19.3	29.6	30.2	28.3	10.9	22.1
	Stanza	Text	25.1	24.5	22.6	23.1	23.5	24.0	30.6	44.4	37.4	22.9	27.8
	Stanza	Filtered Text	**33.5**	35.0	32.3	**31.6**	**30.5**	34.0	36.1	**48.0**	**44.5**	**29.8**	**35.5** (↑5.1%)
	Trankit	Tuple	30.3	17.1	**37.3**	17.2	18.3	22.1	32.4	27.5	30.6	10.5	24.3
	Trankit	Text	23.4	25.4	22.7	22.6	23.8	25.6	30.5	44.8	38.4	24.0	28.1
	Trankit	Filtered Text	33.1	**35.2**	35.6	31.4	28.7	30.3	35.3	46.2	42.8	29.5	34.8
Mistral	None	-	36.7	38.2	39.0	35.8	36.0	38.3	35.6	**51.3**	**38.5**	10.6	36.0
	Stanza	Tuple	27.2	35.9	30.9	31.9	28.1	35.1	30.9	48.4	30.6	9.8	30.9
	Stanza	Text	29.2	32.0	34.4	32.6	30.4	33.4	33.2	47.5	37.1	8.7	31.9
	Stanza	Filtered Text	39.1	39.5	**40.9**	**37.1**	36.6	**40.2**	36.7	50.8	**38.5**	10.3	**37.0** (↑1.0%)
	Trankit	Tuple	27.4	35.3	32.5	31.5	26.9	30.6	31.3	48.0	30.5	10.8	30.5
	Trankit	Text	27.9	32.0	34.7	30.7	31.0	32.7	34.1	46.8	36.4	**11.2**	31.7
	Trankit	Filtered Text	**39.3**	**39.7**	39.3	36.3	**36.9**	37.8	**38.1**	50.9	38.3	**11.2**	36.8
Qwen	None	-	**44.3**	**39.6**	**40.3**	38.0	**36.8**	**43.0**	**40.8**	42.7	39.2	29.1	**39.4**
	Stanza	Tuple	35.4	32.0	34.6	31.8	31.9	37.8	31.4	38.3	38.2	26.1	33.8
	Stanza	Text	33.8	34.8	36.0	33.3	33.3	33.3	29.9	39.5	41.1	**30.6**	34.6
	Stanza	Filtered Text	42.1	32.8	39.8	37.3	33.6	38.4	40.4	**44.7**	45.4	28.6	38.3(↓1.1%)
	Trankit	Tuple	34.3	30.7	35.2	34.1	28.1	35.1	33.6	39.6	37.4	21.8	33.0
	Trankit	Text	35.4	35.2	34.2	33.1	34.0	33.4	30.2	40.5	40.5	27.3	34.4
	Trankit	Filtered Text	39.9	36.0	35.4	**39.4**	34.7	38.6	34.5	44.0	**45.9**	26.6	37.5

Table 4.6: Effect of dependency parses and prompting techniques for LLM-based relation extraction for the REDFM and IndoRE datasets. Performance reported in terms of F1-Score. Best performing methods are shown in bold.

the document. For both zero-shot and in-domain settings across the two datasets, adding linguistic information in the form of dependency graphs improves relation extraction for longer sentences. We posit that including dependency information helps to capture long range dependencies across words and thus the observed gains for longer sentences.

- **Lexical Distance:** We quantify the lexical distance as the number of tokens between the two entities. Here, we observe that dependency information is more helpful for cases where the distance between the entities is not high, i.e. Low and Medium categories.
- **Dependency Path Length:** We quantify the dependency path length as the number of dependency relations that separate the two entities in the dependency graph. We see prominent gains for both short and long range dependency paths, especially for the ZS case for IndoRE. However, similar to lexical distance, the gains are more prominent when the dependency path between the entities is small. Since our chosen GNN has only two layers, we hypothesize that it is unable to capture signals across long dependency paths effectively.

RQ3. Can dependency parses help improve relation extraction performance for LLMs?
Table 4.6 summarizes the performance of three LLMs - LLaMA (Grattafiori et al., 2024), Mistral (Jiang et al., 2023) and Qwen (Yang et al., 2024) for zero-shot relation extraction on the IndoRE and RedFM datasets. To account for the skew in distribution of relations, we employ the macro-F1 score as the primary evaluation metric. We observe that for the LLama-3 and Mistral models,

Source	sum_sq	df	F	P(>F)
C(src)	1.844	2.000	6.265	**0.020**
C(GNN)	0.185	1.000	1.258	0.291
C(DEP)	1.226	1.000	8.330	**0.018**
C(ENC)	0.308	1.000	2.094	0.182
C(src):C(DEP)	0.165	2.000	0.56	0.590
C(src):C(ENC)	7.124	2.000	24.20	**0.000**
C(src):C(GNN)	1.335	2.000	4.534	**0.043**
C(DEP):C(GNN)	0.055	1.000	0.371	0.557
C(ENC):C(GNN)	1.045	1.000	7.098	**0.026**
C(DEP):C(ENC)	1.005	1.000	6.827	**0.028**
Residual	1.325	9.000	NaN	NaN

Table 4.7: Indore In-Domain ANOVA Results. src denotes the corresponding source language. GNN denotes the graph neural network that serves as the backbone, i.e. RGCN or RGAT. DEP denotes the kind of the dependency parser that is used, i.e. Trankit or Stanza. ENC denotes the multilingual encoder, i.e. mBERT or XLMR. Significant results are in bold.

incorporating dependency parses improves performance across several cases. The gains are most prominent when the dependency information is presented in the form of natural language text; we see consistent improvements for the Text Prompt Format over the Tuple Prompt Format, where the information is presented as a list of tuples. We see that the filtered prompt that removes information not pertaining to the two entities, improves performance further.

The improvement can be as significant as 1% to 5% in some cases in terms of absolute F1-score for Mistral and LLama-3 respectively. For the Qwen model, dependency parses do not afford much benefits. Thus the choice of the LLM and the description of the prompt, play a significant role in zero-shot relation extraction performance. It should be noted, however, that the zero-shot performance for the in-context learning setup is significantly worse than the zero-shot cross-lingual performance in the fine-tuned setup. With LLMs, we see an average absolute improvement of 1.67% across all models and languages with the Filtered Text Prompt.

RQ4. Which factors influence generalization?

We now inspect the factors that characterize performance improvements over the baseline for the two datasets in the fine-tuned learning and in-context learning setup. We perform a multivariate ANOVA analysis with the relative performance difference (expressed as a percentage over the baseline), from including the dependency parses, as the dependent variable.

The independent variables chosen are the choice of the multilingual encoder, (mBERT or XLMR), dependency parser (Stanza or Trankit), GNN employed (RGCN or RGAT), and the source and target language [4]. We also consider the pair-wise interaction effects of each of these variables, and note the F-statistic and their corresponding p-value for the indomain (Tables 4.7 and 4.9) and zero-shot cross-lingual (Tables 4.8 and 4.10) respectively.

For the indomain setting in IndoRE, we observe that the relative performance change hinges most on the choice of the dependency parser followed by source language. Although the choice

[4]For the indomain setting we consider only the target language

Source	sum_sq	df	F	P(>F)
C(src)	48.606	2.000	2.449	0.108
C(GNN)	4.009	1.000	0.404	0.531
C(DEP)	23.301	1.000	2.348	0.139
C(ENC)	20.426	1.000	2.058	0.164
C(tgt)	199.051	2.000	10.030	**0.001**
C(tgt):C(DEP)	13.604	2.000	0.686	0.513
C(tgt):C(ENC)	85.332	2.000	4.300	**0.025**
C(tgt):C(GNN)	19.710	2.000	0.993	0.385
C(tgt):C(src)	12.388	4.000	0.312	0.735
C(src):C(DEP)	6.487	2.000	0.327	0.724
C(src):C(ENC)	73.878	2.000	3.723	**0.039**
C(src):C(GNN)	7.459	2.000	0.376	0.691
C(DEP):C(GNN)	0.845	1.000	0.085	0.773
C(ENC):C(GNN)	0.923	1.000	0.093	0.763
C(DEP):C(ENC)	1.561	1.000	0.157	0.695
Residual	238.143	24.000	NaN	NaN

Table 4.8: Indore Cross-Domain ANOVA Results. src and tgt denotes the corresponding source and target language. GNN denotes the graph neural network that serves as the backbone, i.e. RGCN or RGAT. DEP denotes the kind of the dependency parser that is used, i.e. Trankit or Stanza. ENC denotes the multilingual encoder, i.e. mBERT or XLMR. Significant results are in bold.

Source	sum_sq	df	F	P(>F)
C(src)	1.862	4.000	0.408	0.800
C(GNN)	0.719	1.000	0.630	0.438
C(DEP)	3.613	1.000	3.167	0.093
C(ENC)	51.586	1.000	45.228	**0.000**
C(src):C(DEP)	2.027	4.000	0.444	0.775
C(src):C(ENC)	9.053	4.000	1.984	0.143
C(src):C(GNN)	3.373	4.000	0.739	0.578
C(DEP):C(GNN)	0.221	1.000	0.194	0.665
C(ENC):C(GNN)	1.773	1.000	1.555	0.229
C(DEP):C(ENC)	1.601	1.000	1.403	0.252
Residual	19.390	17.000	NaN	NaN

Table 4.9: RedFM In-domain ANOVA Results. src and tgt denotes the corresponding source and target language. GNN denotes the graph neural network that serves as the backbone, i.e. RGCN or RGAT. DEP denotes the kind of the dependency parser that is used, i.e. Trankit or Stanza. ENC denotes the multilingual encoder, i.e. mBERT or XLMR. Significant results are in bold.

Source	sum_sq	df	F	P(>F)
C(src)	14.700	4.000	0.988	0.322
C(GNN)	0.109	1.000	0.029	0.864
C(DEP)	1.111	1.000	0.299	0.585
C(ENC)	4.923	1.000	1.323	0.252
C(tgt)	10.040	6.000	0.450	0.718
C(tgt):C(DEP)	25.753	6.000	1.154	0.334
C(tgt):C(ENC)	106.197	6.000	4.757	**0.000**
C(tgt):C(GNN)	1.642	6.000	0.074	0.998
C(tgt):C(src)	314.185	24.000	3.518	**0.000**
C(src):C(DEP)	23.724	4.000	1.594	0.178
C(src):C(ENC)	323.737	4.000	21.752	**0.000**
C(src):C(GNN)	49.322	4.000	3.314	**0.012**
C(DEP):C(GNN)	0.615	1.000	0.165	0.685
C(ENC):C(GNN)	2.771	1.000	0.745	0.389
C(DEP):C(ENC)	0.389	1.000	0.105	0.747
Residual	647.408	174.000	NaN	NaN

Table 4.10: RedFM Cross-Domain ANOVA Results. src and tgt denotes the corresponding source and target language. GNN denotes the graph neural network that serves as the backbone, i.e. RGCN or RGAT. DEP denotes the kind of the dependency parser that is used, i.e. Trankit or Stanza. ENC denotes the multilingual encoder, i.e. mBERT or XLMR. Significant results are in bold.

Source	sum_sq	df	F	P(>F)
C(src)	58.4	2	0.657	5.26E-01
C(DEP)	2.2	1	0.048	8.28E-01
C(LLM)	1260.3	2	14.17	**6.18E-05**
C(PRM)	3042.5	2	34.22	**3.94E-08**
C(src):C(DEP)	16.7	2	0.187	8.30E-01
C(src):C(LLM)	543.7	4	3.058	**3.36E-02**
C(src):C(PRM)	426.9	4	2.401	7.46E-02
C(DEP):C(LLM)	62.3	2	0.708	5.05E-01
C(DEP):C(PRM)	48.0	2	0.54	5.87E-01
C(LLM):C(PRM)	2205.3	4	12.40	**7.47E-06**
Residual	1200.1	27	NaN	NaN

Table 4.11: Indore Zero-shot ICL ANOVA Results. src denotes the corresponding source language. LLM denotes the large language model that is used for prompting, i.e. Llama. Mistral, or Qwen. DEP denotes the kind of the dependency parser that is used, i.e. Trankit or Stanza. PRM denotes the kind of prompt used, i.e. standard, tuple, text, or filtered text. Significant results are in bold.

Source	sum_sq	df	F	P(>F)
C(src)	6123.02	6	13.34	**2.91E-10**
C(DEP)	5.09	1	0.07	7.97E-01
C(LLM)	4945.81	2	32.32	**6.97E-11**
C(PRM)	12473.39	2	81.51	**1.23E-19**
C(src):C(DEP)	178.97	6	0.39	8.83E-01
C(src):C(LLM)	13819.12	12	15.05	**1.46E-15**
C(src):C(PRM)	1727.37	12	1.88	5.01E-02
C(DEP):C(LLM)	131.03	2	0.86	4.29E-01
C(DEP):C(PRM)	101.88	2	0.67	5.17E-01
C(LLM):C(PRM)	3130.31	4	10.23	**1.12E-06**
Residual	5815.44	76	NaN	NaN

Table 4.12: RedFM Zero-shot ICL ANOVA Results. src denotes the corresponding source language. LLM denotes the large language model that is used for prompting, i.e. Llama. Mistral, or Qwen. DEP denotes the kind of the dependency parser that is used, i.e. Trankit or Stanza. PRM denotes the kind of prompt used, i.e. standard, tuple, text, or filtered text. Significant results are in bold.

of the encoder and the GNN do not have any significant effect on relative performance, their pair-wise interactions is indeed significant. The story is remarkably different for REDFM where only the choice of the encoder has any significant effect on RE.

In the zero-shot cross-lingual setting for IndoRE, we see significant effects arising from the choice of the target language and the pairwise interaction between the choice of the source/target language with that of the encoder. A similar story also holds for REDFM, wherein we notice the only significant interactions are between the choice of the source/target language and the encoder, and also between the choice of the source/target language pairs themselves. **Simply put in the zero-shot setting the role of the dependency information is insignificant for both datasets.**

We carry out a similar statistical analysis for the zero-shot ICL setup, with the relative performance change over the baseline as the dependent variable, and the choice of the LLM (i.e. LLama-3, Qwen, and Mistral), the prompt (i.e. Tuple Format, Text Format, and Filtered Text Format), the language (7 for RedFM and 3 for IndoRE), and the choice of the dependency parser (i.e. Trankit and Stanza) as the independent variables. We also consider the pair-wise interaction effects of each of these variables, and note the F-statistic and their corresponding p-value for the IndoRE and REDFM dataset respectively in Tables 4.11 and 4.12 respectively. We observe, over both datasets, significant effects arising from the choice of the LLM, and the choice of the prompt, as well as the pairwise interaction between the choice of the prompt and LLM, and the choice of the source language and LLM.**Once again, we see that the choice of the dependency parser, i.e. the Stanza or Trankit, does not play a significant role.**

4.3 Conclusion and Takeaways

This chapter demonstrates the utility of linguistic frameworks as formal scaffolds for relation extraction in procedural text, especially when the target setting is data-scarce or distributionally-shifted. We observe that augmenting a transformer based RE baseline with features derived from AMRs and dependency parses significantly improve performance in few-shot regimes and are **robust to cross-domain transfer** even when the corresponding text-based systems degrade under domain shift. Nevertheless, our experiments also surfaces an important boundary condition; the gains diminish as the training data increases, i.e. linguistic scaffolds matter the most under limited supervision.

Our cross-domain experiments on RE over procedural text serve as a control for MLRE where dependency parses normally do not yield significant improvements over the text baseline during cross-lingual transfer. The only exception is for low-resource languages like telugu. We hypothesize that scaffolds are most beneficial when they encode information that are complementary to what can be learned from the text representations alone. Nevertheless we observe how effective organization of the dependency parse information can facilitate in-context learning in LLMs by focusing on only the relevant details. Thus being aware of the task particulars can help us integrate linguistic frameworks as scaffolds more effectively.

We continue with this line of thought in Chapter 5 where we move from internal linguistic structure to an external structured substrate: personalized knowledge graphs, where the generalization bottleneck is not domain/language shift but new graphs/entities per user and the need to reason using structure and paths rather than memorizing node embeddings.

Chapter 5

PERKGQA: Question Answering over Personalized Knowledge Graphs

We outline the methodology for solving the task of question answering over knowledge graphs in a personalized setting, or PERKGQA. In PERKGQA the user has access to their specific knowledge graphs or KG that contains information only relevant to the user. We are only restricted to the user's KG to answer their queries both during training and inference.

5.1 Methodology

We propose two approaches to handle PERKGQA. The former is a non-parametric case-based reasoning approach that does not require training. The latter is a deep neural architecture that employs graph convolutions and encodes structural and path information for reasoning.

5.1.1 PATHCBR

PATHCBR is a non-parametric approach that employs case-based reasoning to retrieve queries without any training . Given a question q, the corresponding knowledge graph $\mathcal{K}_q$ and the source entities, $s_1, s_2, \cdots, s_k$, PATHCBR (Figure 5.1) performs the following steps:
(i) **Query Retrieval:** For a query, q, we first retrieve similar questions from the available training set. We consider a question to be similar if they share similar answer types with the query rather than the entities (Das et al., 2020). We perform Named Entity Recognition (NER) to identify text-spans that correspond to source entities $s_1, s_2, \cdots, s_k$ in K_q (Sun et al., 2019; Wang et al., 2020c). We substitute the extracted text spans with a special [MASK] token, yielding the masked query template q_{MASK}. We hypothesize that masking entities can help us learn the association of the entity with the template and could generalize to unseen entities. We employ a pretrained language model, such as RoBERTa, to create a contextualized embedding of q_{MASK} and call it v_q. We then retrieve the top n questions ($q_1, \cdots, q_n$) and their respective KGs, ($\mathcal{K}_{q_1}, \cdots, \mathcal{K}_{q_n}$) ranked by decreasing cosine similarity between v_q and v_{q_i}. The v_{q_i} are created in the same manner as v_q. We represent the steps of masking and retrieving below:

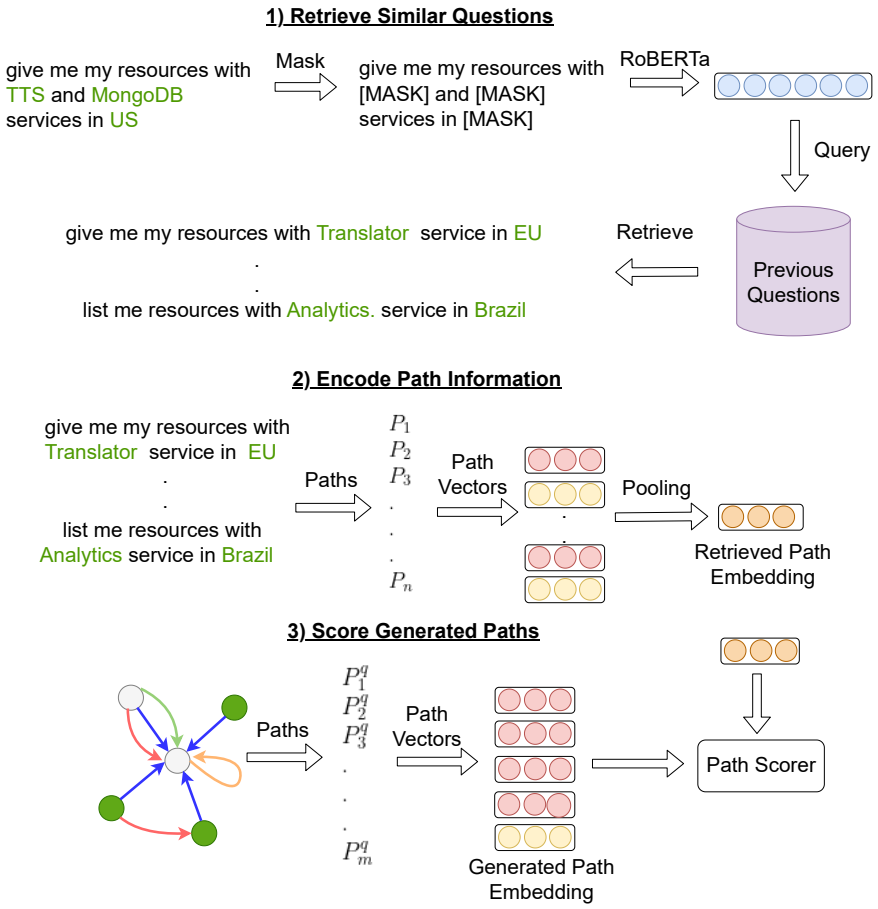


Figure 5.1: PATHCBR Overview: (1) Retrieve questions similar to a given query template from set of questions; (2) Encode path information as a path embedding; (3) Score generated paths using the retrieved path embedding.

$$
\begin{aligned}
q_{\text{MASK}} &\leftarrow \textsc{Mask}(q) \\
v_q &\leftarrow \textsc{RoBERTa}(q_{\text{MASK}}) \\
(q_1, \mathcal{K}_{q_1}), \cdots, (q_n, \mathcal{K}_{q_n}) &\leftarrow \textsc{Retrieve}(v_q)
\end{aligned}
$$

(ii) **Encoding path information:** We now construct the answer paths for the retrieved KGs $\mathcal{K}_{q_i}$. An answer path $p_{s_{ij},a_{ik}}$ comprises a sequence of relations, starting from a source s_{ij} entity to the answer entity a_{ik} in $\mathcal{K}_{q_i}$. There can be multiple answer paths between the source and the answer, but for simplicity we consider only the shortest paths, similar to Srivastava et al. (2021). We represent an answer path, either explicitly as a sequence of relations $(r_{i1}, r_{i2}, ..r_{im})$ leading from s_{ij} to a_{ik}, or by pooling over its constituent relation embeddings $(v_{r_{i1}}, v_{r_{i2}}, ..v_{r_{im}})$. We describe different approaches to obtain the relation embedding v_{r_i} in Section 5.2. Once we have embeddings for individual paths, we pool across all possible answer paths over the retrieved KGs, $\mathcal{K}_q$ to obtain the retrieved path embedding, v_P^q for q. We describe the steps to encode the path

information below:

$$
\begin{aligned}
p_{s_{ij},a_{ik}} &\leftarrow [r_{i1}, r_{i2}, \cdots, r_{im}] \\
v_{p_{s_{ij},a_{ik}}} &\leftarrow \text{MAX-POOL}([v_{r_{i1}}, v_{r_{i2}}, \cdots, v_{r_{im}}]) \\
v_P^q &\leftarrow \text{MAX-POOL}([\forall v_{p_{s_{ij},a_{ik}}}])
\end{aligned}
$$

(iii) **Scoring generated paths:** For the given query q, we generate all possible paths of a certain length, arising from $s_1, s_2, \cdots, s_k$. The length of the path is determined by the maximum length of the answer path encountered during retrieval. These generated paths (say p_j) constitute a sequence of relations arising from the source node (say $r_1, r_2, \cdots, r_m$), similar to the retrieved paths. We encode them by pooling over the constituent relation embeddings to obtain v_{p_j}, the generated path embedding. We finally score the generated path embedding against the retrieved path embedding v_P^q; a higher similarity implies that the generated path is more likely to lead to an answer. However, if we store the path information explicitly as a sequence of relations, then the nodes we reach by traversing the retrieved sequences are answers for q. The equations follow:

$$
\begin{aligned}
p_j &\leftarrow [r_1, \cdots, r_{im}] \\
v_{p_j} &\leftarrow \text{MAX-POOL}([v_{r_1}, ..., v_{r_m}]) \\
score(v_{p_j}) &\leftarrow \text{SIM}(v_{p_j}, v_P^q).
\end{aligned}
$$

5.1.2 PATHRGCN

We now propose our parametric PATHRGCN model that can encode and fine-tune path embeddings for KGQA. Given a question q, the corresponding knowledge graph $\mathcal{K}_q$ and the source entities, $s_1, s_2, \cdots, s_k$, PATHRGCN (Figure 5.2), encompass the following steps during training:
(i) **Initialization:** We encode q using a pretrained language model (PTLM) such as RoBERTa (Liu et al., 2019a), to obtain the corresponding representation, v_q. We use unsupervised graph representation learning techniques like Node2Vec (Grover and Leskovec, 2016) and Walklet (Perozzi et al., 2017), that leverage the neighbourhood information of nodes in $\mathcal{K}_q$ to obtain the corresponding embeddings: $v_{e_1}, v_{e_2}, \cdots, v_{e_N}$ for the N nodes $e_1, e_2, \cdots, e_N$ in $\mathcal{K}_q$. Unlike Wang et al. (2020b,c), we do not use pretrained word embeddings since user-provided names can be arbitrary.

$$
\begin{aligned}
v_q &\leftarrow \text{ROBERTA}(q) \\
v_{e_1}, v_{e_2}, ..., v_{e_N} &\leftarrow \text{WALKLET}(\mathcal{K}_q).
\end{aligned}
$$

(ii) **Information propagation using GNN:** We employ graph neural networks (GNN) to update the node representations of $\mathcal{K}_q$. We modify $\mathcal{K}_q$ by adding the inverse-relations between nodes and self-loops to facilitate information propagation across both directions similar to Wang et al. (2020b,c). We concatenate v_{e_i} with v_q and a binary value of b_i. b_i has a value of 1 or 0, corresponding to whether e_i is a source entity. The resultant representation, $h_{e_i}^0 = [v_q, v_{e_i}, b_i]$, is then passed as input to the first GNN layer, and the representations of all nodes are updated. We perform such updates L times, where L denotes the number of GNN layers, resulting in

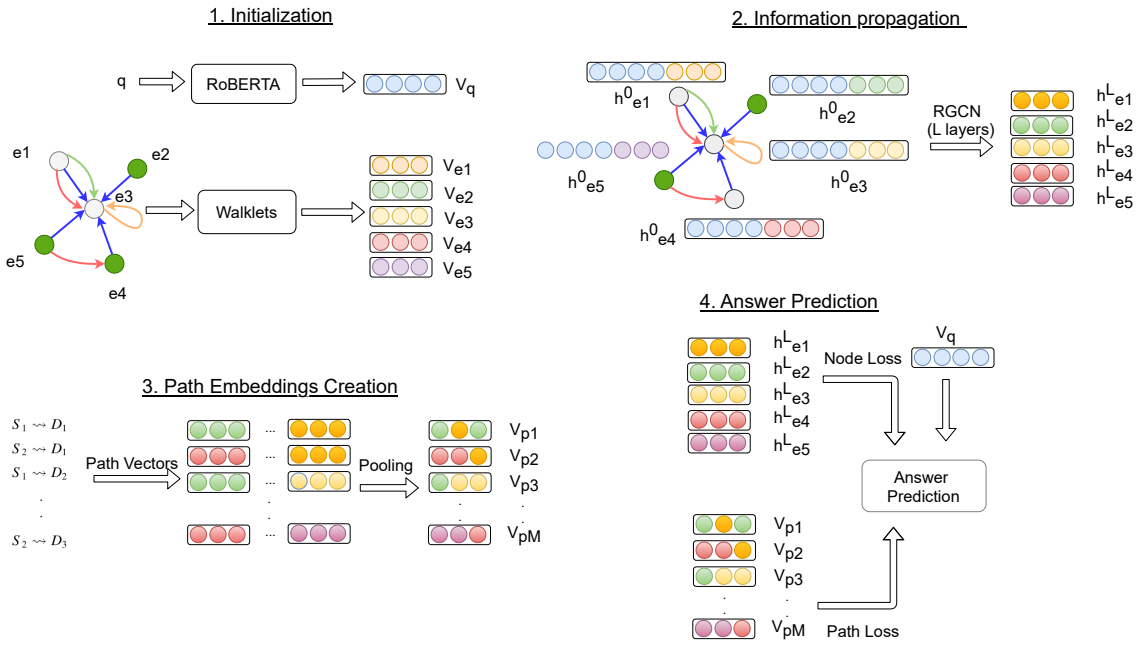


Figure 5.2: PATHRGCN Overview: (1) Initialize the question using a pretrained language model (PTLM) and the nodes in the corresponding KG; (2) Perform information propagation using RGCN to update node embeddings; (3) Encode path information from the source entities (shown in green) to all possible target nodes by pooling over the constituent node embeddings; (4) Perform answer prediction at both the path and node level.

the final representation of $h^L_{e_i}$. We use softmax as the non-linear activation and add dropout for regularization between updates. We use the RGCN model (Schlichtkrull et al., 2018a) to account for different relationships between nodes.

$$h^0_{e_i} \leftarrow v_q \oplus v_{e_i} \oplus b_i$$
$$h^{j+1}_{e_i} \leftarrow \text{RGCN}(h^j_{e_i})$$

(iii) **Path embedding generation:** We construct all possible paths $p_1, p_2, ...p_m$ upto a fixed distance from the source entities, and generate their corresponding path embeddings. The embeddings for path p_j or v_{p_j} is obtained by pooling over the updated representations of the nodes that constitutes p_j. We hypothesize that learning the path structure can provide intermediate supervision (Srivastava et al., 2021) and can help prune-out nodes that are unlikely to be reached from the source.

$$v_{p_j} \leftarrow \text{MAX-POOL}(h^L_{e_i}) \; \forall e_i \in p_j$$

(iv) **Answer prediction:** We perform answer prediction both at the node and path level. We concatenate the updated representation for node e_i as $h^L_{e_i}$, with the question-embedding v_q, and pass it through a linear layer with sigmoid activation. to obtain $\hat{y}_{e_i}$). This represents the probability of e_i being an answer and is trained against the ground truth value of y_{e_i}. We perform the same procedure at the path level to obtain the probability of path p_j that leads to e_i as ($\hat{y_{p_j,e_i}}$). We use

binary cross-entropy loss for answer prediction at the node level (NL) and path level (PL) and minimize these losses jointly during training. Specifically :

$$\hat{y}_{e_i} \leftarrow \sigma(\text{FFN}(h^L_{e_i} \oplus v_q))$$
$$\hat{y}_{p_j,e_i} \leftarrow \sigma(\text{FFN}(v_{p_j} \oplus v_q))$$
$$\text{NL} = -\sum_{e_i \in \mathcal{K}_q} y_{e_i} \cdot log(\hat{y}_{e_i})$$
$$\text{PL} = -\sum_{e_i} \sum_{\forall p_j \rightsquigarrow e_i} y_{e_i} \cdot log(\hat{y}_{p_j,e_i})$$

Inference: During inference, given a question q^* and its corresponding sub-graph $\mathcal{K}_{q^*}$, the learnt PATHRGCN models outputs (i) probability that the node $e_1, e_2, \cdots, e_N$ is an answer and (ii) probability that the paths $p_1, p_2, \cdots, p_m$ leads to an answer. Thus for a given entity, e_i, we compute the maximum probability amongst all paths that end in e_i. We compute the mean of this probability alongside the probability of e_i being an answer.

5.2 Experiments

5.2.1 Baselines

EmbedKGQA: The EmbedKGQA model (Saxena et al., 2020) performs Knowledge Graph Completion (KGC) on an existing knowledge graph, to learn node representations. They use ComplEx (Trouillon et al., 2016) to generate node embeddings, to account for the anti-symmetric nature of the relations between nodes. Furthermore, they use RoBERTa (Liu et al., 2019a) as the Pre-Trained Language Model (PTLM) to encode the question. They learn an objective function to select answers based on the similarity between question and node embeddings and further perform pruning based on the relation type to prevent over-generation of candidates. EmbedKGQA can perform arbitrary multi-hop reasoning, is not restricted to a specific neighbourhood, and can effectively handle incomplete links/edges. To ensure EmbedKGQA can be applied in our setting, we carried out KGC on the KG associated with the question instead of the entire Freebase KG. This ensures that the entity representations are distinct for each individual KG.
Rel-GCN: The Rel-GCN approach of Wang et al. (2020b) first constructs a smaller sub-graph $\mathcal{K}_q$ for a given question, using PPR (Haveliwala, 2003) from the large base knowledge-graph, $\mathcal{K}$. They encode the question q using PTLM as v_q, and use TransE (Bordes et al., 2013) on $\mathcal{K}$ to obtain the node representations v_{e_i} for node e_i in $\mathcal{K}$. They concatenate the node embedding with the question-embedding e_q, and then perform RGCN on $\mathcal{K}_q$ to obtain their updated representations. These updated representations are used to score whether a given node is an answer or not. For PERKGQAsetting we perform TransE not on the original graph, $\mathcal{K}$, but on each sub-graph $\mathcal{K}_q$.
GlobalGraph: The GlobalGraph technique of Wang et al. (2020c) is similar in conception to Rel-GCN, having the same steps, (i) sub-graph construction, (ii) encoding representations of question and nodes, (iii) running RGCN to update the node representations. Moreover, to capture long-dependencies between nodes, the model leverages the set of incoming and outgoing relations to assign a global type for each node. They also identify nodes that are correlated with the question

and construct a dynamic graph connecting such similar nodes. GCN over this dynamic graph yields updated representations for such nodes. Once again, for PERKGQA, we perform TransE on the individual KG associated with the question $\mathcal{K}_q$.

5.2.2 Experimental Details

PATHCBR: We experiment with how masking entities impact QA performance. For CloudKGQA, we identify entities by performing string-match over text spans in the question to their corresponding nodes in the KG. For Mod-WebQSP, we use the publicly available SpaCy NER[2]. [1] We also experiment with SpaCy's POS-Tagger to mask proper nouns. The masked query is encoded using the [CLS] token of RoBERTa-BASE (Liu et al., 2019a). We experiment with different ways to encode relations, either as a one-hot vector or using RoBERTa-BASE to encode the text. We perform max-pooling over the constituent relation embedding to obtain the resultant path-embedding. Likewise, max-pooling over the resultant path-embeddings yields the retrieved path-embedding. We also experimented with mean-pooling, but max-pooling fared consistently better. The generated paths are similarly encoded during inference. We compute cosine-similarity between a generated and retrieved path embedding. We retrieve the top 5 questions in descending order of their similarity for a given query.

PATHRGCN: For PATHRGCN, we use RoBERTa-BASE to encode the question text, and Walklet (Perozzi et al., 2017) during initialization to generate the unsupervised node-representations for each KG. We use Walklet instead of Node2Vec since it exhibits the highest performance over several node classification tasks (Rozemberczki and Sarkar, 2020). Moreover, it does not require any additional features to generate the embeddings and is computationally fast; Walklet was ≈ 20 times faster than Node2Vec. The embedding sizes for the question, nodes, and GNN layers was set to 768, 128, and 200, respectively. We fix L, the number of GNN layers to 1. For Path-RGCN, the length of an answer-path is chosen based on the maximum distance between a source entity and an answer entity encountered during training. This corresponds to a distance of 3 for CloudKGQA and a distance of 2 for Mod-WebQSP. We used Adam optimizer with a low learning rate of 2e-5, a decay of 5e-4, and patience of 30, and trained for 100 epochs. Each model took around 3 hours to complete on a p3.8x large EC2 instance.

Baselines: For Rel-GCN (Wang et al., 2020b), and GlobalGraph (Wang et al., 2020c), we use RoBERTa-BASE (Liu et al., 2019a) to encode the question, and TransE embeddings to initialize the nodes (Bordes et al., 2013). We use the publicly available PyTorch-Geometric library (Fey and Lenssen, 2019) to implement RGCN (Schlichtkrull et al., 2018a) for these two baselines. The embedding dimensions for our question, node, and GNN layers are 768, 128, and 200 respectively. The number of GNN layers, was set to 2 and 1 for Rel-GCN and GlobalGraph respectively, as specified in their papers. For EmbedKGQA, we use the publicly available code of Saxena et al. (2020) [2] along with the default hyper-parameters for training. We use the publicly-available, LibKGE (Broscheit et al., 2020) library to generate Complex embeddings for each KG.

[1][2] `https://spacy.io/usage/spacy-101#annotations-ner`
[2] https://github.com/malllabiisc/EmbedKGQA

5.2.3 Evaluation Metrics

We evaluate the performance of the baselines and our proposed approaches across two metrics commonly used in KGQA, namely, Hits@1 and Accuracy. For a given question, Hits@1 has a value of 100 if the highest-scoring candidate is a correct answer; else, it is 0. Accuracy denotes the fraction of answers predicted correctly amongst the top K candidates (as a percentage). We also measure Hits@K for a question, for which the value is 100 if the answer is present amongst the top K candidates; else it is 0. For both Accuracy and Hits@K, K is the number of correct answers. We carry out experiment for five random seeds and report the mean and standard deviation. We perform statistical significance using the paired bootstrapped test of Berg-Kirkpatrick et al. (2012) in Dror et al. (2018).

5.3 Results

	CloudKGQA			**Mod-WebQSP**		
Method	Hits@1	Hits@K	Accuracy	Hits@1	Hits@K	Accuracy
EmbedKGQA	31.6 ± 3.3	31.6 ± 3.3	31.6 ± 3.3	29.1 ± 1.9	32.6 ± 2.2	25.1 ± 1.8
Rel-GCN + TransE	44.9 ± 8.7	52.5 ± 6.1	41.4 ± 6.3	49.4 ± 2.3	59.6 ± 1.2	48.5 ± 1.8
GlobalGraph + TransE	46.6 ± 3.6	56.1 ± 1.9	43.6 ± 2.5	48.4 ± 0.6	59.1 ± 0.7	48.3 ± 0.9
PATHRGCN + Walklet (Ours)	$\mathbf{90.4 \pm 2.1}$	$\mathbf{91.3 \pm 1.5}$	$\mathbf{90.7 \pm 1.5}$	$\mathbf{68.6 \pm 0.2}$	$\mathbf{75.2 \pm 0.4}$	$\mathbf{68.5 \pm 0.3}$

Table 5.1: Performance of the baselines and our approaches on CloudKGQA, and Mod-WebQSP. *K* is the number of correct answers. We report the mean and standard deviation across 5 runs. The best performance is highlighted.

In this section, we pose the following research questions (RQs) and attempt to answer the same. We present instances of preprocessed questions that serve as input to the model.

RQ1. How do our proposed approaches fare on PERKGQA compared to KGQA baselines?

We observe that both PATHCBR and PATHRGCN, yield the highest performance on CloudKGQA, outperforming the existing baselines by over 100% for Hits@1 and Accuracy in Table 5.1. We attribute the poor performance of prior KGQA techniques to their inability to (i) learn global node embeddings over the large base KG or (ii) update the embeddings during training.

For Mod-WebQSP, PATHRGCN achieves the highest performance outperforming preexisting baselines significantly (p-value ≤ 0.001). However, PATHCBR achieves performance comparable to the baselines, and can answer questions corresponding to templates encountered during training, for instance, *"who plays ken barlow in coronation"*. We attribute the low performance of PATHCBR to:

(i) The underlying global KG for Mod-WebQSP is more complex and dense. There are 572 possible relations as opposed to 8 for CloudKGQA . Moreover, there can be multiple relations between two entities, (e.g. *'location.country.capital'* and *'location.contained_by'* are both valid relations between Tokyo and Japan), a characteristic absent in CloudKGQA. The possible paths

increase exponentially with hops, and additional supervision afforded by GNNs helps answer these questions with long-range dependencies (Wang et al., 2020c).

(ii) Not all possible relations encountered during inference were available during training. E.g., the most relevant question retrieved for *"what was wayne gretzky 's first team"* was *"what team does plaxico burress play for"*, because the relation corresponding to *"first team"* was absent during training. At times, the pretrained language model could not infer the query's semantic meaning. E.g, the most relevant question for *"what town was martin luther king assassinated in"* was *"what town was abe lincoln born in"*, despite the occurrence of questions like *"where was huey newton killed"*. Thus if the templates are widely different, it is not sufficient to encode the question using a PTLM; rather, we need to fine-tune the questions to learn meaningful representation.

We further inspect the capabilities of our techniques to address the individual characteristics of PERKGQA, namely multiple answers, variable hop distance, multiple constraints, and variable KG size. Our approaches outperform baselines consistently and significantly on all such fronts.

RQ2. What is the impact of entity masking and encoding different path-information strategies on PATHCBR's performance?

	No Masking			Masking Entities			Masking Proper Nouns		
CloudKGQA	Hits@1	Hits@K	Acc	Hits@1	Hits@K	Acc	Hits@1	Hits@K	Acc
Path Sequence	67.9	67.9	67.9	67.9	67.9	67.9	66.4	66.4	66.4
One-Hot Vector	88.8	89.4	88.8	<u>95.4</u>	<u>96.7</u>	<u>95.8</u>	82.4	84.9	83.6
Text Embedding	83.6	86.1	84.8	**95.7**	**96.9**	**96.0**	78.4	80.9	79.5
Mod-WebQSP	Hits@1	Hits@K	Acc	Hits@1	Hits@K	Acc	Hits@1	Hits@K	Acc
Path Sequence	33.0	37.9	32.8	41.6	46.5	41.1	<u>47.4</u>	<u>52.2</u>	<u>46.2</u>
One-Hot Vector	32.5	41.1	32.3	44.6	52.1	43.7	**49.3**	**56.0**	**48.0**
Text Embedding	13.7	21.1	16.1	22.4	28.7	23.5	25.2	32.1	26.7

Table 5.2: Mean performance of PATHCBR across different settings for entity masking and encoding path information, as a sequence of relations (Path Sequence), as a One-Hot Vector, or as a Text Embedding using a PTLM. The best performance is highlighted in bold and the second best is underlined.

We investigate the impact of different strategies for masking entities and encoding path information on the performance of the PERKGQAtask for the two datasets and report them in Table 5.2.

(i) **Entity-masking:** For Mod-WebQSP, entity masking using either a publicly-available NER or a POS Tagger, shows a huge boost in performance as seen in Table 5.2. Masking entities facilitates retrieving relevant questions which share similar answer types rather than similar entity names in the query. For example, for "What county is *greeley colorado* in ?", the most relevant question retrieved after masking is "What county is *novato california* in?", as opposed to "What college is in *greeley colorado*?". We observe a similar trend for CloudKGQA when we mask entities linked to nodes in the KG. However, the performance drops substantially when we use a POS-Tagger. Since the naming convention for nodes is arbitrary, like *"abc123"*, they are not

detected as proper nouns; this creates inconsistent templates, and irrelevant questions appear higher in the ranked list.

(ii) **Encoding path information:** We observe that encoding relations as one-hot vectors fare just as well, if not better than encoding the relation-text using a PTLM. This is especially true for Mod-WebQSP where relation-names have high lexical overlap and thus exhibit high similarity. For example, for *"where is jamarcus russell from"*, the correct relation is **"people.person.place_of_birth"**, but the relation predicted, was **"people.person.date_of_birth"**. Encoding relations as one-hot-vectors circumvents this issue. Encoding the path-information, as a sequence of relations works well for Mod-WebQSP but not for our CloudKGQA, since the questions encountered during inference have different templates.

RQ3. What role does graph structure and path-information play on PERKGQA?

	CloudKGQA			Mod-WebQSP		
Method	Hits@1	Hits@K	Accuracy	Hits@1	Hits@K	Accuracy
Rel-GCN + TransE	44.9 ± 8.7	52.5 ± 6.1	41.4 ± 6.3	49.4 ± 2.3	59.6 ± 1.2	48.5 ± 1.8
GlobalGraph + TransE	46.6 ± 3.6	56.1 ± 1.9	43.6 ± 2.5	48.4 ± 0.6	59.1 ± 0.7	48.3 ± 0.9
PATHRGCN + TransE	51.4 ± 4.8	68.4 ± 2.6	57.0 ± 4.4	53.1 ± 0.9	62.6 ± 0.7	52.0 ± 0.8
Rel-GCN + Walklet	79.1 ± 3.9	79.8 ± 4.2	79.3 ± 4.0	63.0 ± 1.1	71.3 ± 0.8	63.0 ± 1.2
GlobalGraph + Walklet	86.3 ± 3.8	87.2 ± 4.0	86.5 ± 3.9	64.4 ± 0.9	72.6 ± 0.9	64.6 ± 0.8
PATHRGCN + Walklet	$\mathbf{90.4 \pm 2.1}$	$\mathbf{91.3 \pm 1.5}$	$\mathbf{90.7 \pm 1.5}$	$\mathbf{68.6 \pm 0.2}$	$\mathbf{75.2 \pm 0.4}$	$\mathbf{68.5 \pm 0.3}$
PATHRGCN + Walklet - NL	90.3 ± 7.1	91.1 ± 6.9	90.6 ± 6.8	65.7 ± 1.0	73.0 ± 1.1	65.8 ± 1.0

Table 5.3: Performance of the baselines and PATHRGCN when initialized with different node embeddings. We report the mean and standard deviation across 5 runs. The best performance is highlighted. NL stands for Node Loss.

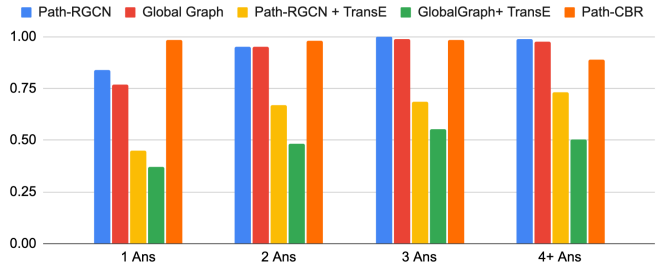
We investigate the benefits of unsupervised graph representation learning techniques to initialize node embeddings. In particular, we compare the efficacy of Walklet and TransE embeddings, when applied to Rel-GCN, GlobalGraph, and PATHRGCN. We see significant improvements for all models when TransE embeddings are substituted with Walklet in Table 5.3.

Since we operate for individual KGs, TransE does not have sufficient information to generate meaningful node representations. Walklet leverages the neighbourhood information and thus can capture the structural representation for each KG. PATHRGCN significantly outperforms the baselines on both fronts, when all three models are initialized with Walklet or when all three models are initialized with TransE embeddings.

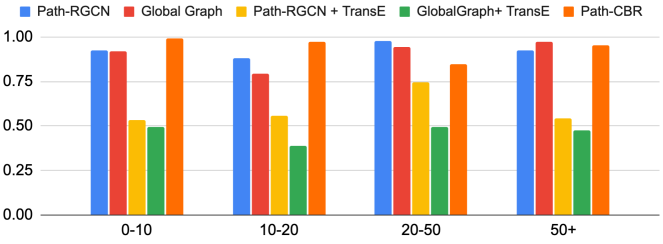
We also investigate the importance of incorporating node loss (NL in Table 5.1) for additional supervision. This aids Mod-WebQSP, where multiple relations between entities give rise to several possible paths between source and answer, most of which are spurious. Since multiple paths do not exist for CloudKGQA, removing the node loss does not deteriorate performance.

RQ4. How does our proposed approaches fare against the baselines for different KGQA properties?

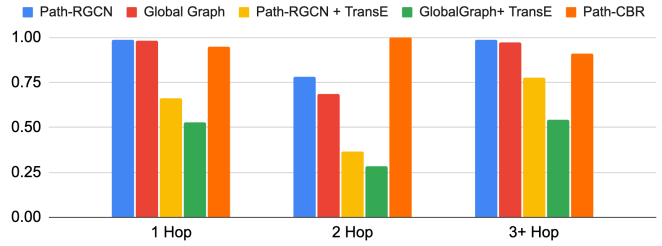
We investigate the performance of the different methods (accuracy) on the PERKGQA task for different properties of the dataset. The methods we investigated were (i) PATHRGCN (ii)



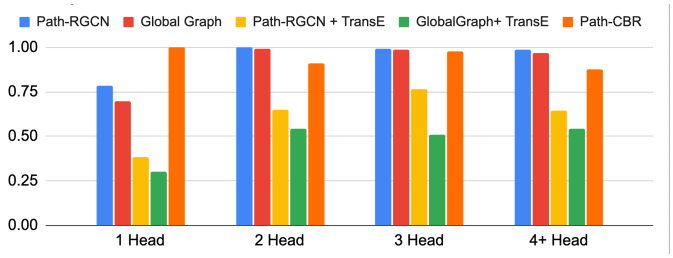
(a) CloudKGQA: Accuracy vs # Answers



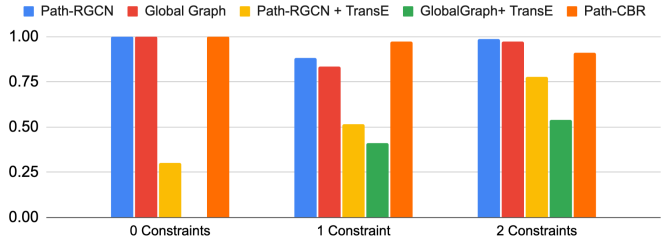
(b) CloudKGQA: Accuracy vs Subgraph size



(c) CloudKGQA: Accuracy vs # Hops

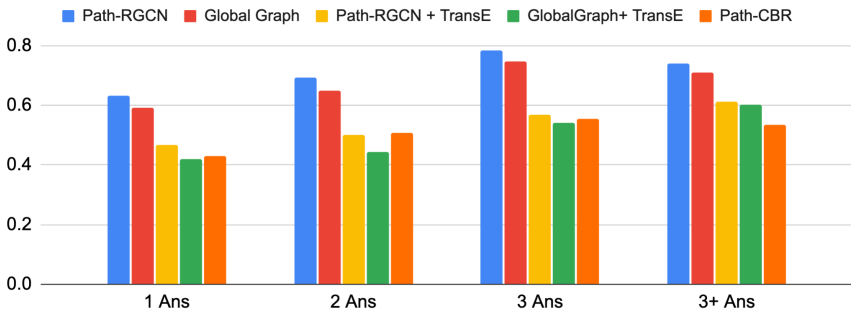


(d) CloudKGQA: Accuracy vs # source entities

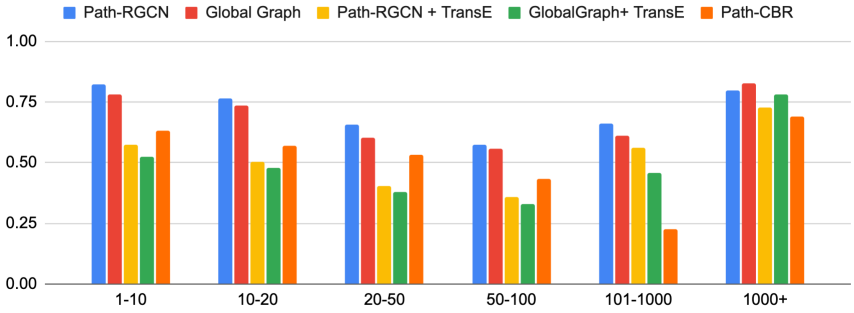


(e) CloudKGQA: Accuracy vs # Constraints

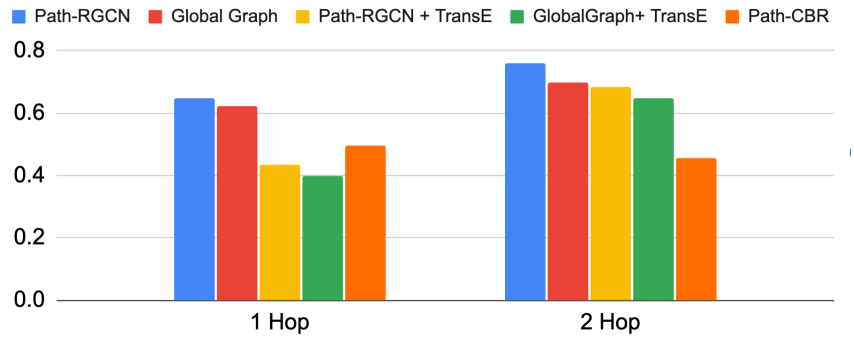
Figure 5.3: Performance of the models on the CloudKGQA dataset across different parameters such as size of the subgraph, number of answers, hops, source entities, and constraints.



(a) Mod-WebQSP: Accuracy vs # Answers



(b) Mod-WebQSP: Accuracy vs # Subgraph Size



(c) Mod-WebQSP : Accuracy vs # Hops

Figure 5.4: Performance of the different techniques on the CloudKGQA dataset based on the number of hops, head-nodes, logical constraints

PATHCBR (iii) GlobalGraph initialized with Walklet (iv) PATHRGCN initialized with TransE, and (v) GlobalGraph initialized with TransE, the best baseline without any modifications. We investigate the following dataset properties.
(i) Variable number of answers: We observe the performance for variable number of answers, for CloudKGQA in Figure 5.3a and for Mod-WebQSP in Figure 5.4a.
(ii) Variable size of the graph: We note the effect of for varying graph size on different methods for CloudKGQA in Figure 5.3b and for Mod-WebQSP in Figure 5.4b.
(iii) Variable Hop Distance: We investigate the performance for varying number of hops for the CloudKGQA in Figure 5.3c and for Mod-WebQSP in Figure 5.4c.
(iv) Complex Questions: We observe specifically for CloudKGQA how the accuracy across methods varies for complex questions, based on the varying number of head-nodes in Figure 5.3d and the number of logical constraints in Figure 5.3e. This information was available to us for our internal dataset but not for Mod-WebQSP.

For CloudKGQA, we observe that our non-parametric PATHCBR approach achieves the highest performance when the number of answers is few (≤ 3), the subgraph is comparatively smaller (# edges ≤ 50), the number of hops is few (≤ 2), and when there are fewer constraints, (number of logical constraints ≤ 2, and number of source entities ≤ 3). PATHRGCN boasts a comparative higher performance for the converse scenarios, i.e., greater answers, a larger size of the KG, more hops, and additional constraints. This observation highlights the trade-off between model complexity and the complexity of the question itself. The only exception lies for the 2-hop cases wherein PATHCBR achieves a score of 1.0 because the questions seen during training had a similar template, and answers were found within two hops. Nevertheless, across all sub-cases, we see that our proposed architectures, PATHRGCN or PATHCBR, boasts the highest performance, while the GlobalGraph + TransE, the best performing baseline, achieve the lowest performance. The baseline fares are consistently poorer than the PATHRGCN + TransE, which shows that incorporating the path information was beneficial across all stages.

For Mod-WebQSP, we see that our PATHRGCN model consistently boasts the highest accuracy across all sub-cases. The trend is similar to CloudKGQA, where the PATHRGCN model can handle a larger KG size and more considerable hop distance. The only difference is the higher performance of PATHCBR when there are more answers, which is justifiable since the mean number of answers for Mod-WebQSP is five instead of two.

5.4 Conclusion and Takeaways

In this chapter, we formalize PERKGQA as a setting where each user has their own KG and the system must answer questions using only that user-specific graph, thereby contrasting with prior work, since the model cannot rely on a single global KG with stable entities across train/test. Our main takeaway is that strong performance in PERKGQA hinges on scaffolds derived from the internal structure of the corresponding KG of the user and path information from sources to answers, precisely to avoid brittle dependence on learned node representations that would not transfer to new users and graphs corresponding to those users.

Methodologically, we operationalize this idea with two complementary approaches: (i) a non-parametric case-based reasoning method (PATHCBR) that retrieves similar question templates

and reuses reasoning patterns without training, and (ii) a neural architecture (PATHRGCN) that uses graph convolutions while explicitly encoding structural and path cues for reasoning.

We build upon the observation on how leveraging the graph structure as a scaffold enables generalization to unseen graphs and entities (a "new-user" shift) in the following chapter. Chapter 6 extends the generalization challenge from "new entities and new graphs" to "new schemas and reasoning templates" in a semantic parsing based KBQA approach by leveraging isomorphisms that characterizes the reasoning structure itself. We highlight how these isomorphisms can serve as both diagnostic tools as well as scaffolds for facilitating zero-shot generalization in KBQA.

Chapter 6

IsoKBQA: Isomorphism for KBQA

In this chapter, we investigate the dual role of isomorphisms for the task of question answering over knowledge bases. Firstly, we showcase how isomorphisms act as diagnostic tools to analyze the strengths and pitfalls of different KBQA systems; for example whether a model is better equipped to handle questions with longer hops or more constraints. Secondly, we illustrate how isomorphisms can serve as scaffolds to facilitate zero-shot generalization in LLMs without additional training.

6.1 Isomorphisms as diagnostic tools

In this section we outline our approach for using isomorphisms as diagnostic tools for investigating the zero-shot generalization capabilities of KBQA systems on our proposed GrailQA++ dataset. We begin with an overview of our experimental setup that describes pre-existing baseline KBQA systems and our evaluation criteria. We then put forward our research questions and carry out a detailed analysis of the same.

6.1.1 Experimental Setup

Baselines: We experiment with two semantic-parsing baselines for KBQA namely RNG-KBQA (Ye et al., 2021b) and ArcaneQA (Gu and Su, 2022). We chose these models because they encapsulate two different strategies of carrying out semantic parsing in the context of KBQA (Gu et al., 2022a). Furthermore, they achieve impressive performance on the GrailQA leaderboard and also have publicly available checkpoints which can be used for evaluation. We follow the inference setting mentioned in their Github repositories, with the single exception that for RNG-KBQA we do not restrict ourselves to the subset of Freebase domains for GrailQA.

RNG-KBQA (Ye et al., 2021b) follow a ranking-based approach wherein they first enumerate all possible candidates and then perform semantic matching to rank the enumerated candidates in decreasing order of relevance. They then use a pre-trained LM (T5-large) to generate an executable query from the top-ranked candidates.

ArcaneQA (Gu and Su, 2022) employ a seq2seq generative LM to obtain the final logical form from the natural language query. They leverage a constrained decoding paradigm that leverages

the information in the KB during query generation to ensure executability.

	RNG-KBQA		ArcaneQA	
Dataset	EM	F1	EM	F1
GrailQA (dev)	83.5	86.0	77.9	81.7
GrailQA++	28.5	38.6	18.6	32.5
- EAD	56.1	70.2	31.5	49.9
- GraphQ	53.2	61.7	30.2	44.8
- WebQSP	19.9	25.9	17.6	28.7
- CWQ	12.6	23.0	10.2	23.2

Table 6.1: EM and F1 scores for RNG-KBQA and the ArcaneQA model on the GrailQA and GrailQA++ datasets (with gold entities). EAD stands for the Expert Annotated Dataset that we had created.

Evaluation Criteria: We evaluate the performance of the two baselines in terms of EM (exact match) and F1 scores (between the predicted and gold answers). We decouple the impact of entity recognition and entity linking from the main task of KBQA by providing gold entities during inference. All experiments are carried out on a RTX-1080Ti GPU with 12GB RAM, using the author-provided model-checkpoints on the public GrailQA dev set.

6.1.2 Results and Analysis

In this section we put forward the following research questions and attempt to answer the same.
RQ1. How well do the baselines generalize to our proposed GrailQA++ dataset?

We present the zero-shot performance of RNG-KBQA and ArcaneQA on GrailQA and GrailQA++ in Table 6.1. We observe that models show impressive performance on GrailQA with RNG-KBQA achieving a very high F1 score of 86.0 overall. We also note that these models suffer a drop of at least 10 points in Gu and Su (2022) in absence of gold entities, emphasizing the importance of NER and entity-linking (EL) for KBQA.

Nevertheless, even while controlling for perfect EL, the performance drops sharply on GrailQA++, resulting in an F1 score of 38.6 and 32.5 for RNG-KBQA and ArcaneQA respectively. We attribute this to the skewed distribution of ismorphisms in the original GrailQA dev split, where the simpler isomorphisms (Iso-0,1,2) accounts for 97% of the dataset. RNG-KBQA achieves an F1 score of 86.5 and 30.1 on the simple and complex isomorphisms in GrailQA respectively (see Table 6.2).

We also investigate the models' performance on questions with additional functions. These functions are (i) comparatives (ex. greater than, less than), (ii) superlatives (argmax, argmin), (iii) counting or aggregation, and (iv) none (absence of any specific operation). The results in Table 6.3 highlights that ArcaneQA scores higher on superlatives and comparative functions (in terms of F1 score) as opposed to RNG-KBQA for GrailQA++.

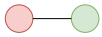
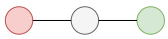
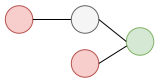
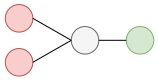
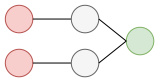
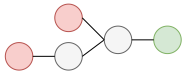
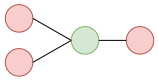
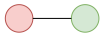
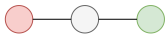
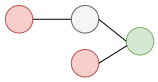
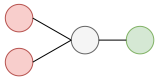
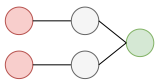
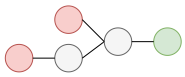
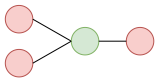
		GrailQA (Dev)		GrailQA++		EAD	
	Iso-Codes	RNG	Arc	RNG	Arc	RNG	Arc
0		87.1/ 88.0	83.8/ 86.4	53.2/ 59.7	36.9/ 47.6	89.2/ 91.2	71.1/ 77.1
1		81.9/ 85.1	66.7/ 70.5	47.9/ 53.4	36.7/ 47.0	81.5/ 83.7	52.6/ 59.3
2		74.8/ 86.2	53.3/ 75.8	39.2/ 51.9	15.8/ 34.7	96.9/ 97.9	50.0/ 67.8
3		5.6/ 44.8	0.0/ 20.2	13.2/ 28.3	2.1/ 24.3	75.3/ 88.8	14.8/ 44.1
4		9.8/ 47.6	11.5/ 27.5	25.0/ 32.9	1.8 / 16.8	35.6/ 49.0	4.9/ 25.6
5		0.0/ 1.5	0.0/ 0.0	0.8/ 10.1	16.7/ 22.1	3.1/ 19.2	15.3/ 25.9
6		-	-	0.0/ 3.4	3.0/ 8.8	-	-
8		-	-	0.0/ 4.4	0.0/ 1.6	-	-
11		-	-	0.0/ 61.2	0.0/ 47.5	0.0/ 61.2	0.0/ 47.5
		GraphQ		WebQSP		CWQ	
	Iso-Codes	RNG	Arc	RNG	Arc	RNG	Arc
0		63.4/ 69.9	31.8/ 42.7	29.0/ 36.7	31.4/ 43.4	-	-
1		53.2/ 57.6	32.1/ 44.8	9.0/ 13.3	13.0/ 26.2	49.7/ 58.1	45.4/ 53.9
2		63.6/ 87.9	6.1/ 6.1	0.0/ 16.9	0.0/ 16.7	18.0/ 33.2	5.9/ 27.3
3		48.4/ 98.9	0.0/ 72.1	0.0/ 13.3	0.0/ 13.3	4.5/ 18.2	0.7/ 19.9
4		17.9/ 24.7	0.0/ 30.9	19.1/ 23.4	0.0/ 6.3	-	-
5		0.0/ 0.0	90.9/ 92.1	-	-	0.0/ 7.9	7.5/ 11.5
6		-	-	-	-	0.0/ 3.4	3.0/ 8.8
8		-	-	-	-	0.0/ 4.4	0.0/ 1.6
11		-	-	-	-	-	-

Table 6.2: EM / F1 scores for RNG-KBQA (RNG) and ArcaneQA (Arc), across the different Isomorphisms (Iso) in GrailQA (zero-shot subset) and GrailQA++. EAD stands for the expert annotated dataset that was created.

	RNG-KBQA			
Dataset	None	Count	Comparative	Superlative
GrailQA (Dev)	90.1/ 90.9	91.1/ 95.3	38.6/ 73.8	0.0/ 7.8
GrailQA++	29.9/ 40.9	84.3/ 84.3	0.0/ 1.9	0.0/ 6.6
	ArcaneQA			
Dataset	None	Count	Comparative	Superlative
GrailQA (Dev)	80.2/ 83.2	68.1/ 71.7	41.2/ 65.5	72.1/ 76.5
GrailQA++	20.0/ 34.7	20.6/ 21.6	0.0/ 15.5	8.7/ 15.5

Table 6.3: EM/ F1 scores for RNG-KBQA and the ArcaneQA model on the GrailQA and GrailQA++ datasets with different functional forms. None means no special function was present.

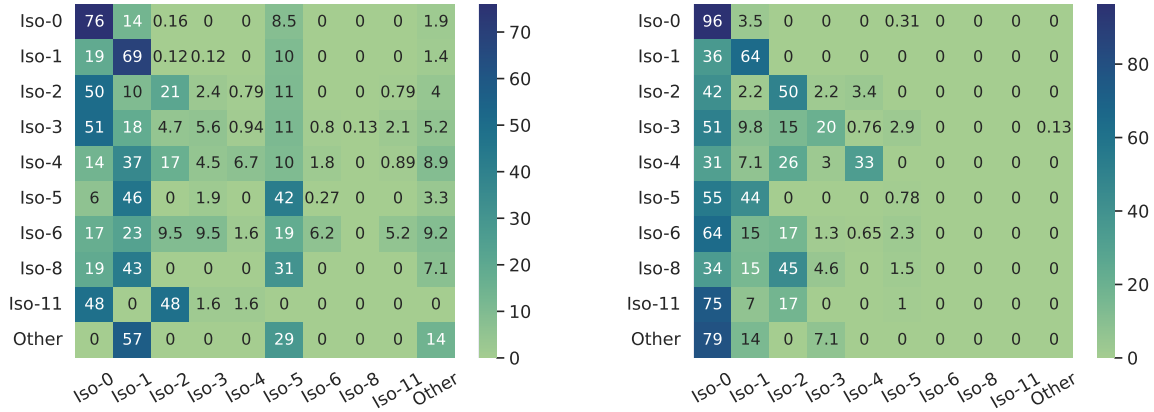


Figure 6.1: Confusion matrices for gold Isomorphisms vs predicted Isomorphisms on the GrailQA++ dataset for ArcaneQA (left) and RNG-KBQA (right).

RQ2. Do models exhibit similar performance on different isomorphism types?

We present a breakdown of the model performance according to the isomorphism type for GrailQA and GrailQA++ in Table 6.2.

The enumeration strategy of RNG-KBQA generates candidates corresponding to the first 5 isomorphisms (Iso-0,1,2,3 and 4). Consquently, we obtain high scores for those specific isomorphisms and low (or zero) EM for the others. This suggests that a ranking-based approach, such as RNG-KBQA, requires prior knowledge of all possible isomorphisms to facilitate meaningful generalization. Nevertheless, RNG-KBQA achieves a comparatively higher F1 score for most isomorphisms in GrailQA++. ArcaneQA, on the other hand, has a higher score on Iso-5, and Iso-6 for GrailQA++.

We hypothesize that different KBQA models are biased towards generating/retrieving logical forms that conform to specific isomorphisms. To delve deeper, we categorize the models mispredictions into different isomorphism types. We obtain confusion matrices for correct isomorphisms against the predicted isomorphism type for ArcaneQA and RNG-KBQA in Figure 6.1.

Dimension	GrailQA	EAD	GraphQ	WebQSP	CWQ	All
Complexity Score	-0.282^{***}	$+0.001$	$+0.00$	$+0.00$	-0.124	-0.093^{*}
Grammaticality	$+0.013$	$+0.011$	-0.063	$+0.037$	$+0.027$	-0.023
Readability	$+0.000$	$+0.001$	-0.001	-0.001	-0.001	-0.002^{***}
Coherence	-0.069^{***}	-0.075^{***}	-0.085^{***}	-0.031^{**}	-0.024^{***}	-0.068^{***}
Sentence Length (#W)	$+0.010^{***}$	-0.006	-0.015^{*}	$+0.028^{*}$	$+0.006$	$+0.0021$
Common Nouns (#N)	$+0.037^{***}$	$+0.000$	$+0.031$	-0.022	$+0.027^{***}$	$+0.026^{***}$
Zero-shot Items (#Z)	-0.065^{***}	$+0.011$	-0.005	-0.114^{***}	-0.100^{***}	-0.035^{***}

Table 6.4: Coefficients of the different dimensions on the F1 score obtained through linear regression and their corresponding p-values. A positive coefficient indicates a positive correlation and vice versa. *, **, *** indicate that the coefficient is statistically significant with a p-value $\leq$ 0.05, 0.01, and 0.001 respectively.

We observe that ArcaneQA is biased towards generating logical forms with longer hops (See the column corresponding to Iso-5 and Iso-1 in Figure 6.1) which explains the higher EM of ArcaneQA on GrailQA++ for Iso-5. Furthermore, since RNG-KBQA outputs logical forms corresponding to the first 5 isomorphisms (Iso-0,1,2,3,4), the mispredictions are mostly confined to those specific forms.

Our experiments demonstrates the complementary strengths of these models such that RNG-KBQA fares better in presence of multiple constraints (Iso-3,4) whereas ArcaneQA is better for multiple hops (Iso-5).

RQ3. What linguistic characteristics of a dataset enable zero-shot generalization?

We observe from Table 6.1 that the constituent datasets of GrailQA++ exhibit wide variation in performance for both models. While complex isomorphisms usually have lower scores than the simpler ones, there are a few exceptions . For example, on the GraphQ split in Table6.2, RNG-KBQA has a very high F1 score of 98.9 on Iso-3 as opposed to 69.9 for Iso-0. This motivates us to delve deeper and investigate whether certain dataset characteristics can explain this variation.

We inspect the following dataset characteristics namely the sentence length (#W), number of common nouns (#N), number of zero-shot items (#Z), readability, grammaticality, complexity, and coherence. The number of common nouns (#N) serves as a proxy for explicitness, i.e how thorough were the annotators in framing the question. The metrics corresponding to readability, complexity, and grammaticality helps to gauge the naturalness of a question, whereas coherence is used to quantify fluency. We adopt the following dimensions of Khosla et al. (2023) on our proposed dataset.

- Sentence Length (#W): We simply count the number of words for each natural language questions across all datasets.
- Common Nouns (#N): We use NLTK's POS-tagger to identify common nouns that corresponding to "NN" and "NNS" tags.
- Grammaticality & Complexity: We use the BLIMP (Warstadt et al., 2020) and COLA corpora (Warstadt et al., 2019) to fine-tune BERT-based text classification model to detect whether a given question is grammatical or not. We follow the same to determine whether a given question is complex or not, i.e. has several clauses.

Dimension	GrailQA(Dev)	EAD	GraphQ	WebQSP	CWQ
Complexity Score	0.0 (0.1)	0.2 (0.4)	0.0 (0.0)	0.0 (0.0)	0.0 (0.1)
Grammaticality	0.7 (0.5)	0.6 (0.5)	0.8 (0.4)	0.7 (0.4)	0.8 (0.4)
Readability	60.5 (26.9)	58.4 (24.5)	71.8 (25.7)	77.0 (25.3)	69.9 (22.1)
Coherence	-9.8 (1.2)	-9.7 (1.2)	-9.4 (1.2)	-9.9 (1.3)	-9.3 (1.2)
Sentence Length (#W)	12.6 (3.7)	17.3 (5.2)	11.1 (3.0)	6.7 (1.6)	14.4 (3.3)
Common Nouns (#N)	4.7 (1.8)	6.6 (2.4)	3.4 (1.3)	2.2 (1.0)	5.3 (1.6)
Zero-shot items (#Z)	2.1 (0.9)	2.6 (1.3)	2.4 (1.2)	1.6 (0.7)	2.1 (0.9)

Table 6.5: We present the mean (std) on different linguistic dimensions on the zero-shot split of GrailQA development set (Dev), and GrailQA++.

- Readability:We use the Flesch-reading score to characterize the readability of each question in the dataset, using the readability library in python. [1]
- Coherency: We quantify fluency or naturalness of a question using coherency. We measure coherency using a reference free metric called CTRLEval (Ke et al., 2022).

We perform a multivariate regression analysis over the combined dataset or "All" with F1 score as the dependent variable and the aforementioned linguistic factors and number of zero-shot items as the independent variables to identify which dimensions are statistically significant. We carry out the same analysis for each individual dataset. We present the results in Table 6.4.

For the combined dataset, All, we observe that all factors except grammaticality and sentence length, are significant . We also note that complexity, readability, coherence, and the number of zero-shot items are negatively correlated with F1, while the number of common nouns (#N) is positively correlated.

While, there are fluctuations in trends, we note that for all the datasets, "coherence" is significantly and negatively correlated with performance. This observation aligns with prior findings of Linjordet and Balog (2022) where the fluency and naturalness of questions degrades KBQA performance. Moreover, the negative correlation with #Z implies that questions with a greater proportion of unseen classes and relations are harder for models to answer. Furthermore, a positive correlation with #N signifies that being more explicit in framing questions is beneficial for model performance. We see similar trends in #N and #Z across most datasets.

An interesting observation is that our constructed dataset, EAD, is most similar to GraphQuestions both in terms of the EM/F1 scores (Table 6.1) as well as coefficients of different linguistic dimensions (Table 6.4). One hypothesis is that these datasets were created in a similar fashion.

We observe that GrailQA mostly follows a similar trend to All since it accounts for 50% of the entire dataset. However, the readability metric for All is influenced by the pre-existing datasets (CWQ, GraphQ, and WebQSP) which have comparatively higher scores in Table 6.5. All in all, we note that KBQA systems struggle with fluent and natural questions (high coherence and readability scores).

[1] https://pypi.org/project/py-readability-metrics/

6.2 Isomorphism Prediction Task

6.2.1 Task Formulation

GrailQA

Question: what is the role of opera designer gig who designed the telephone / the medium?

Structured Input: m.0pm2fgf: opera.opera_production: The Telephone / The Medium | opera.opera_role opera.opera_role.opera opera.opera_production | opera.opera_production opera.opera_role.opera opera.opera_role |
...
opera.opera_character opera.opera_role.role opera.opera_role | opera.opera_production opera.opera_production.cast opera.opera_role | opera.opera_production opera.opera_production.designers opera.opera_designer_gig

Isomorphism: Iso-1

Figure 6.2: An example of the isomorphism prediction task for the GrailQA dataset.

As discussed in Chapter 3, we formulate the isomorphism prediction task as multi-class classification with the natural language question and the structured knowledge as the input and the isomorphism category as the output. The structured knowledge, a snapshot of the KB schema, contains all the necessary information to carry out prediction. We explore two distinct ways of organizing this structured information for our task.

Text Mode: The first mode involves simply representing the structured information as a linearized sequence of tuples (e_h, r, e_t) which is concatenated with the natural language question. This resultant **text sequence** is then finetuned on the task using a pre-trained language model.

Graph Mode: Alternately, we can also arrange the structured information in the form of a knowledge graph or KG, where the nodes represent the head and tail entities in the tuple (i.e. e_h and e_t) while the edge represents the relation i.e. r . Such a topological view ensures the information is organized consistently and avoids ordering effects. We emphasize that this approach simply provides an alternate way to organize the data and does not add any additional information. We can thus encode the graph information using a Graph Neural Network (GNN) and subsequently finetune the resultant representation for our task.

Given the complementary ways of representing the same information, we propose a unified approach that leverages both the text and graph representations.

6.2.2 Methodology

We propose a unified, task-agnostic framework, henceforth called IsoCoD (**Figure 6.3**), to understand how the text and graph representations evolve during training. The proposed framework is generalizable and enables us to observe how information from text and graph are represented and integrated.

Task Loss: The text-based and graph-based representations are encoded using modality-specific encoders: $h_{\text{t}} = f_{\text{t}}(T)$ and $h_{\text{g}} = f_{\text{g}}(G)$. We then create a hybrid representation h_{hybrid} through concatenation or residual connection (addition) to perform isomorphism prediction and compute the task loss as follows:

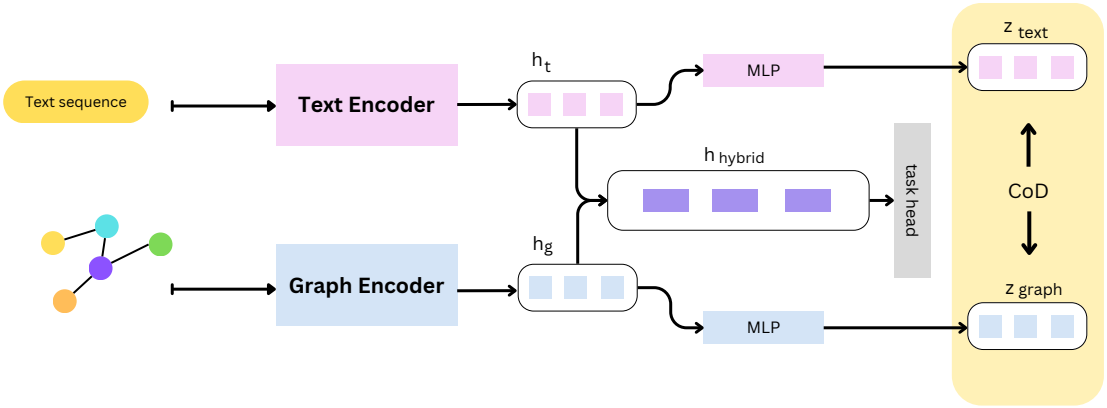


Figure 6.3: Our unified framework for analyzing how text and graph representations complement each other. A text sequence and its corresponding graph are processed by separate encoders. Their outputs are used in two ways: (1) combined as hybrid inputs for task prediction, and (2) projected into a shared space where a contrastive co-distillation (CoD) objective encourages mutual learning and enables representation-level analysis.

$$h_{\text{hybrid}} = f_{\text{fuse}}(h_{\text{t}}, h_{\text{g}}) \tag{6.1}$$

$$\mathcal{L}_{\text{task}} = CE(h_{\text{hybrid}}, y) \tag{6.2}$$

where y denotes the gold isomorphism category and CE is the cross-entropy loss function (since isomorphism prediction is a multiclass classification task).

To analyze text and graph representations, we require a shared space where they can be directly compared. Thus, we apply modality-specific MLP projection heads that learn to map each representation into a shared latent space during training:

$$z_{\text{text}} = \text{MLP}_{\text{t}}(h_{\text{t}}), z_{\text{graph}} = \text{MLP}_{\text{g}}(h_{\text{g}}). \tag{6.3}$$

Contrastive codistillation loss: While learning a shared space enables comparison, it cannot solely influence how text and graph will complement one another. We thus apply a contrastive knowledge co-distillation (CoD) objective (Yao et al., 2024) which combines a contrastive loss with a stop-gradient operation (Chen and He, 2021) to explicitly encourage bidirectional knowledge transfer. Such a formulation allows us to observe how the information encoded in one modality influences the other during mutual learning.

Formally, the contrastive loss l_{cl} between the teacher t and the student s representations is:

$$l_{\text{cl}}(t, s) = -\log \frac{e^{sim(t,s)/\tau}}{\sum_u 1_{[u \neq t]}\, e^{sim(t,u)/\tau}} \tag{6.4}$$

where u indicates representations from the training data other than t and s, $sim(.,.)$ is cosine similarity, τ is the temperature scaling parameter (Tian et al., 2022a). Note that the notions of

"teacher" and "student" are interchangeable and fully symmetric: one scenario treats the text projection behaves as the teacher supervising the graph projection, while in another the graph projection supervises the text projection. This bidirectional design ensures that either modality can act as teacher or student at each step, thus mutually distilling knowledge from each other. Hence, the full CoD loss is computed as

$$\mathcal{L}_{\text{CoD}} = \frac{1}{2}\sum_{i}[l_{\text{cl}}(z_i^{\text{text}}, \hat{z}_i^{\text{graph}}) + l_{\text{cl}}(z_i^{\text{graph}}, \hat{z}_i^{\text{text}})] \tag{6.5}$$

where $\hat{\cdot}$ is the stop gradient operator (Chen and He, 2021) that sets the input variable to a constant. Finally, we combine this with the task loss to enable end-to-end model optimization:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda\mathcal{L}_{\text{CoD}} \tag{6.6}$$

where λ controls the weight of the CoD signal.

ISOCOD thus serves as a task-agnostic framework to facilitates learning and analysis over dual modalities.

6.2.3 Experimental Particulars

We outline the experimental details for the isomorphism prediction task. Following Xie et al. (2022), we use the T5-base model as the text encoder. To accommodate the input length constraints, we simplify the way the entities in the KB (for WebQSP) are represented in the linearized graph input. Instead of using full entity identifiers (e.g., `m.02896`), we assign short, unique placeholder tokens (e.g., `<E1>`, `<E2>`) to each entity as a part of the tokenizer vocabulary. This helps reduce the input sequence length and avoids unwanted subword tokenization. In addition, we ensure that these placeholder tokens are assigned consistently across modalities: the same entity is represented as node v_i in the graph and as token `<Ei>` in the linearized text.

We explore different GNN models, i.e., GCN, GAT, RGCN, and RGAT to encode the structured information when represented as a graph. We initialize the nodes in the graph using Walklets (Perozzi et al., 2017) and adopt a 2-layer GNN model for all our experiments. We still encode the question using a separate encoder such as T5 and fuse the question representation with the nodes after initialization. We use cross-entropy loss as the loss function and macro F1 score as the evaluation metric. We run our experiments over five seeds to account for variations across runs.

6.2.4 Isomorphism as scaffolds for KBQA in LLMs

In a stand-alone or baseline setting, an LLM takes as input the query and the structured knowledge information (KB) and generates the corresponding logical form (LF) or s-expression. To ensure the LLM has sufficient context to do the task, we provide guidelines on how to construct a valid s-expression. Additionally, we can also provide few-shot examples to further ground the generation process in the system-prompt. Subsequently, we explore both zero-shot and few-shot examples of generating s-expressions using LLMs.

We improve upon this baseline setting by introducing a simple verification module that leverages the isomorphism information for KBQA. We present an overview of the same in

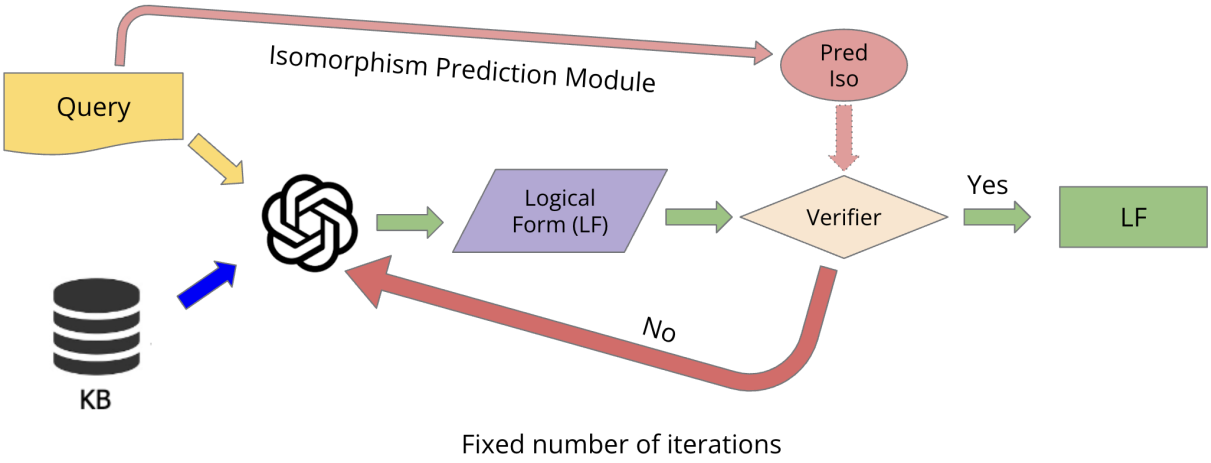


Figure 6.4: Our framework of incorporating the isomorphism information with LLMs for the task of question answering over knowledge bases (KBQA) with verification and feedback. For cases where the gold isomorphism is available for a given query, we can simply supply that instead of the Pred Iso to the verifier.

Figure 6.4. Specifically, the verifier deterministically compares the isomorphism category of the generated logical form with the one predicted by our framework, i.e. IsoCoD. If there is a mismatch, we simply prompt the LLM to regenerate the logical form, else we accept the current generated s-expression as the final logical form. We carry out this process for a maximum of three iterations.

We use the exact match (EM) metric to validate the correctness of the generated logical form with the gold s-expression. We use WebQSP, GrailQA, and GrailQA++ as the datasets of interests for this experiment. We randomly sample 1000 instances from GrailQA to ensure feasibility of our in-context learning experiments.

We experiment with both proprietary and open-weight LLMs. The proprietary LLMs in our experiments include GPT-3.5-turbo and GPT-4o, while the open-weight models include the instruction tuned versions of Llama-3-8B, Gemma-3-4B, and Gemma-3-27B. For each LLM, we evaluate KBQA performance in a stand-alone setting without any verification which serves a baseline. We compare this baseline performance against the case where the predicted isomorphism is used during verification to capture realistic settings. Finally, to benchmark the best-possible outcome performance, we use the gold isomorphism during verification.

6.2.5 Results and Analyses

In this section, we put forth the following research questions and attempt to answer the same.

RQ1. What is the impact of IsoCoD on isomorphism prediction?

We present the overall results of isomorphism prediction on the three datasets, i.e. WebQSP, GrailQA, and GrailQA++ in Table 6.6. We compare the performance of our proposed framework IsoCoD against the other baselines, i.e. the text mode and the graph mode separately. We observe unanimously across all datasets, IsoCoD achieves a significantly higher F1 score than either the

Mode	Dataset WebQSP	GrailQA	GrailQA++
Question	50.22	62.09	36.87
Text	62.83	66.40	46.42
Graph	55.49	63.47	36.32
IsoCoD	**67.93**	**70.32**	**47.91**

Table 6.6: Isomorphism prediction performance measured in terms of macro F1 score for the three datasets.

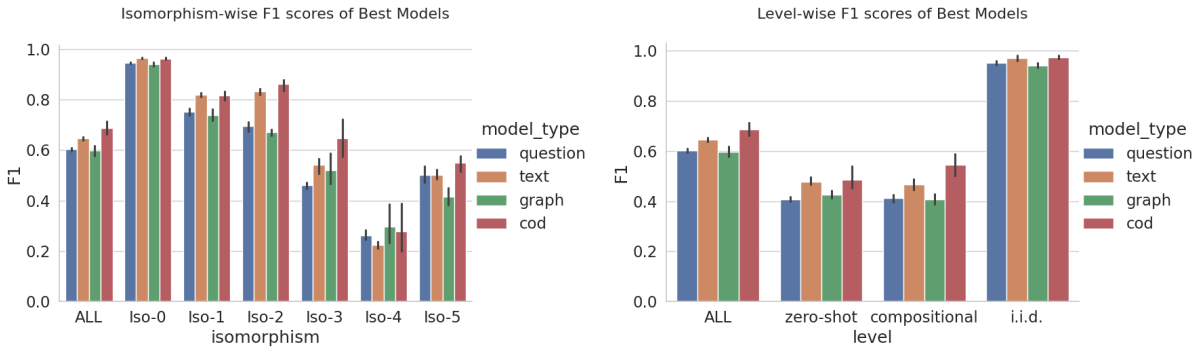


Figure 6.5: Isomorphism prediction performance on GrailQA . We compare performance across different isomorphism categories and generalization levels for different model settings.

text or the graph baseline. To account for the impact of structured knowledge on isomorphism prediction, we ablate the text baseline to include only the natural language question as part of the input. As we see from Table 6.6, the Question mode achieves the lowest performance for both WebQSP and GrailQA and compares similarly to the graph mode on GrailQA++.

Impact of generalization level and isomorphism category on isomorphism prediction: Isomorphism prediction performance varies significantly across datasets; IsoCoD achieves an F1 score of 47.9 on GrailQA++ compared to 70.3 on GrailQA. As opposed to GrailQA, GrailQA++ has an equal distribution of complex and simple isomorphisms and comprises solely of zero-shot instances. We thus investigate the isomorphism prediction performance of IsoCoD across different generalization levels and isomorphism categories. We present our findings pictorially in Figures 6.5 and 6.7 for GrailQA and GrailQA++ respectively. Our proposed framework IsoCoD outperforms other baselines on GrailQA for the zero-shot and compositional generalization splits. A similar story holds for the more complex isomorphism forms, i.e. Iso-3 and Iso-5 on GrailQA.

Nevetheless, there is a considerable drop in F1 score on the zero-shot split of GrailQA compared to the entire dataset, i.e. ALL (48.6 vs 70.3). In fact, the IsoCoD performance on the zero-shot split of GrailQA is similar to that of GrailQA++. Unsurprisingly, the greatest performance degradation happens for the complex isomorphisms categories, especially, Iso-4 and Iso-5, possibly due to the poor distribution of these categories in the training data. We see similar results for WebQSP in Figures 6.6.

Impact of GNN architecture on isomorphism prediction performance: We also explore the

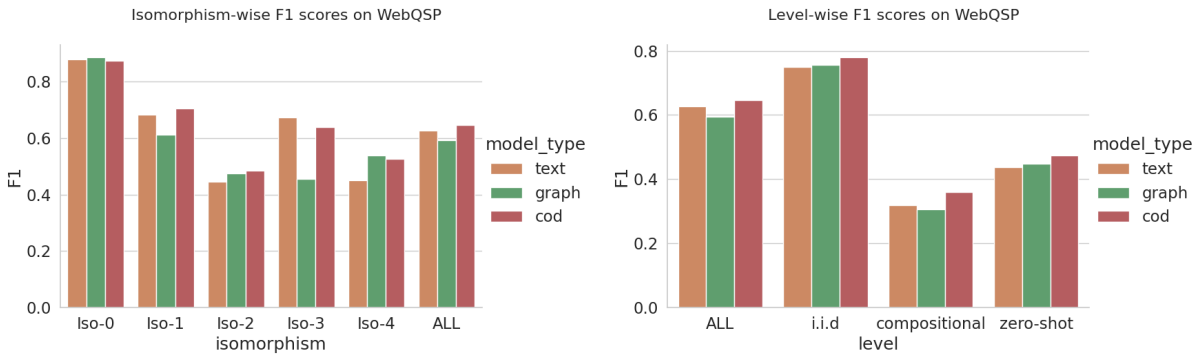


Figure 6.6: Isomorphism prediction performance on WebQSP. We compare performance across different isomorphism categories and generalization levels for different model settings.

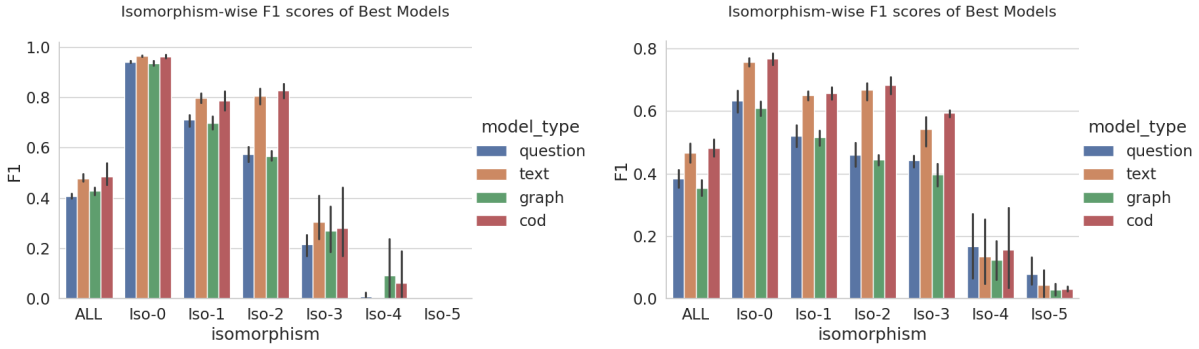


Figure 6.7: Performance on the isomorphism prediction task on the zero-shot instances of GrailQA (left) and GrailQA++ (right) across isomorphism categories for different model settings (i.e. Question, Text, Graph, and IsoCoD).

choice of the GNN architecture on the isomorphism prediction task. We present the results for 4 different GNN models on the Graph only and IsoCoD setting in Table 6.7. Relational GNNs or GNNs that explicitly include the edge relation such as RGCN and RGAT fare worse than their non-relation counterpart i.e. GCN and GAT respectively on the Graph only view for both GrailQA and GrailQA++. We attribute this to increased memorization as a result of over-parameterization as evidenced by the higher F1 scores for relational GNNs on the i.i.d. validation set. On the other hand, we note higher performance for the relational GNNs on WebQSP, possibly due to significantly fewer relations in the WebQSP ontology ($\approx$ 1K) as opposed to GraiQA ($\approx$ 19K). Our proposed IsoCoD framework encourages regularization by aligning the over-parameterized graph representation with the corresponding text representation. Subsequently, we see higher or comparable scores for relation GNNs than their non-relational counterparts in the CoD setting.

Error analysis for isomorphism prediction: We inspect the misclassifications errors of IsoCoD on the isomorphism prediction task. We use confusion matrices to visualize the fraction of cases a given isomorphism category (true label) is classified as another (predicted label) in Figures 6.8. We use the best performing model (IsoCoD with RGAT as the GNN module) for our analyses on GrailQA and GrailQA++. We make the following observations.

Firstly, simple isomorphisms (Iso-0,1,2) have significantly lower errors than the complex

GNN Model	Graph WebQSP	Graph GrailQA	Graph GrailQA++	CoD WebQSP	CoD GrailQA	CoD GrailQA++
GAT	52.66	**63.47**	34.97	66.12	67.08	46.86
RGAT	54.06	48.88	24.19	63.34	**70.32**	**47.91**
GCN	50.77	58.76	**36.32**	**67.93**	67.80	45.76
RGCN	**55.49**	46.48	23.20	67.41	68.46	46.87

Table 6.7: Isomorphism prediction performance in terms of macro F1 score for different datasets with different GNN architectures as the backbone for the Graph and CoD mode. The best performance is highlighted in bold.

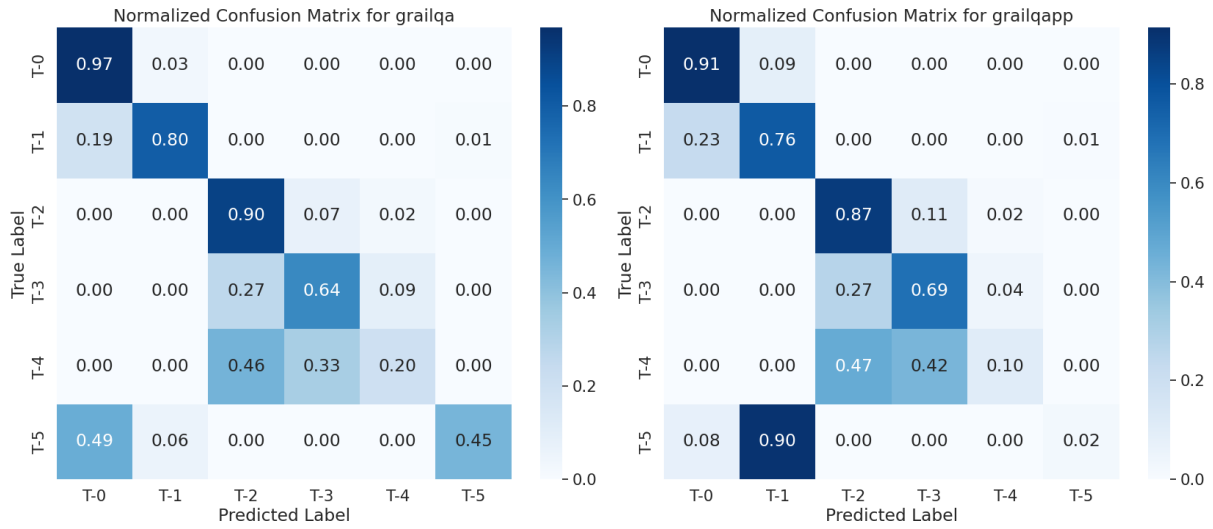


Figure 6.8: Confusion matrices for our proposed CoD model (with RGAT as the backbone) on the isomorphism prediction task on the GrailQA and GrailQA++ dataset.

isomorphisms, as evidenced by the strength of the main diagonal. Secondly, the predicted isomorphism categories are clustered based on the number of constraints. For example, Iso-0, Iso-1, and Iso-5 corresponds to the isomorphism category where the answer originates from a single constraint (or entity) and requires traversing multiple hops (1, 2, and 3 hops for Iso-0, Iso-1, and Iso-5 respectively). Subsequently, an instance corresponding to Iso-0, Iso-1, or Iso-5 will be predicted to belong to one of these categories only. The same holds for Iso-2, Iso-3 and Iso-4 which involves exactly 2 constraints that converge to an answer after varying number of hops. This observation highlights that even if we are unable to predict the exact label, we can confidently filter out the isomorphism categories that do not have the same number of constraints as the predicted category.

RQ2. Do IsoCoD improve KBQA performance in LLMs?

We note the impact of incorporating isomorphism information on KBQA performance for proprietary LLMs and open-weight LLMs in Tables 6.8 and 6.9 respectively. Unsurprisingly, adding in the gold isomorphism resulted in consistent and significant performance improvements

	GPT 3.5-turbo						GPT-4o					
	WebQSP		GrailQA		GrailQA++		WebQSP		GrailQA		GrailQA++	
	zshot	fshot	zshot	fshot	zshot	fshot	zshot	fshot	zshot	fshot	zshot	fshot
Baseline	6.2	52.1	2.2	38.2	1.3	26.2	8.3	65.8	34.5	69.4	32.1	63.8
Ours + Pred Iso	14.8	59.7	7.5	41.1	4.5	27.2	17.9	71.8	49.3	71.9	41.7	63.5
Ours + Gold Iso	15.0	62.9	6.4	42.6	3.7	31.3	22.9	81.0	55.3	74.0	51.4	73.0

Table 6.8: KBQA performance in terms of EM score (exact match accuracy) of proprietary LLMs (GPT-3.5-turbo and GPT-4o) on WebQSP, GrailQA, and GrailQA++ under different settings. The baseline setting involves simply prompting the LLM with zero-shot (zshot) and few-shot (fshot) examples. We compare this performance against our verification setting with the predicted isomorphism / gold isomorphism as the input.

	LLama-3-8B			Gemma-3-4B-it			Gemma-3-27B-it		
	WQSP	GQA	GQA++	WQSP	GQA	GQA++	WQSP	GQA	GQA++
Baseline	18.1	23.2	11.4	39.5	18.2	10.9	59.8	54.1	45.5
Ours + Pred Iso	20.7	30.0	16.0	47.5	21.4	11.3	64.8	59.7	45.8
Ours + Gold Iso	21.5	31.1	17.5	48.6	21.7	12.1	67.4	62.6	56.0

Table 6.9: KBQA performance in terms of EM score (exact match accuracy) of open-weight LLMs (LLama-3-8B, Gemma-3-4B-it, and Gemma-3-27B-it) on WebQSP (WQSP), GrailQA (GQA), and GrailQA++ (GQA++) under different settings . The baseline setting involves simply prompting LLMs in a few-shot (fshot) examples. We compare this performance against our verification setting with the predicted / gold isomorphism as the input.

over the baseline across all datasets, LLMs, and k-shot scenarios. We observe a mean relative gain of 20.5% in the EM score over few-shot experiments when we use the gold isomorphism during verification. Likewise, the mean relative gain for the predicted isomorphism category during verification is 12.2%, thereby cementing the effectiveness of our approach. The cases where using the predicted isomorphism yields comparable performance to the baseline is for the more challenging GrailQA++ dataset and using relatively powerful LLMs such as Gemma-3-27B-it and GPT-4o. We conclude that gains on the isomorphism prediction task directly correlates with improved performance on the KBQA task.

6.3 Conclusion and Takeaways

Chapter 6 establishes the dual-role of isomorphisms in KBQA. Firstly, we highlight how popular KBQA datasets predominately contains questions corresponding to simple isomorphism categories and thus the need to construct challenging benchmarks with equitable distribution of simple and complex isomorphism forms. We construct GrailQA++ to address this particular limitation and we subsequently showcase how state-of-the-art KBQA systems exhibit significant performance degradation on our constructed dataset, most of which is attributable to the skew in isomorphism distribution.

Secondly, we showcase the utility of isomorphisms to improve KBQA performance. We propose ISOCOD a framework for predicting the isomorphism category for a given KBQA query, by unifying text and graph representations of the constituent knowledge store. We can incorporate this isomorphism prediction module into a light-weight verification-and-regeneration loop using LLMs. Using isomorphisms as scaffolds in this manner yields consistent improvements over the no-verification baseline, with gold isomorphisms providing an upper bound and predicted isomorphisms delivering realistic gains.

We thus conclude the first part of our thesis where formal scaffolds provide realistic improvements across several information extraction or tasks dealing with factual knowledge. Formal scaffolds are most valuable when the problem involves grounding the information into a structured representation or when they can serve as constraints to guide exploration or generation during inference. We now transition from formal scaffolds to the next part of the thesis where the scaffolds are no longer expressed as a formal structure but in a free-text format and captures information not explicitly stated in the text. We thus introduce informal scaffolds in the form of rationales that can make the latent cues explicit and facilitate generalization for several social meaning tasks.

Part II

Informal Scaffolds

Chapter 7

Operationalizing Informal Scaffolds for Generalization in Dialogue

7.1 Informal Scaffolds: Definition

As opposed to explicitly grounding the textual information into a structured representation, informal scaffolds augment a system's capabilities using free-form natural language. We can thus define informal scaffolds as systems that enriches or clarifies the base input text by either (i) injecting relevant world knowledge absent from the text or (ii) implicit cues not explicitly articulated in the text's surface. In effect, informal scaffolds verbalize knowledge that may be latent in a model's parameters (or an external tool's behavior) into usable textual guidance such as rationales, explanations, and chain-of-thought style reasoning (Majumder et al., 2022; Wiegreffe et al., 2021; Wei et al., 2022c). We contextualize the role of informal scaffolds in the role of social meaning understanding tasks in this thesis.

7.2 Social Meaning

Beyond content focused areas of Natural Language Processing, the past two decades have witnessed a surge of interest in modeling language from a social perspective (Nguyen et al., 2016), such as predicting emotions and mental states of users to identifying what strategies are being employed to solve a particular task like persuasion or negotiation. We specifically deal with social scenarios where human communication plays a pivotal role.

According to sociologist Erving Goffman (Goffman, 2002) language conveys two forms of "social meaning", i.e. one that is *given* or intentional, and one that is *given off* or unintentional, often thought of as "reading between the lines". The former embodies the idea of linguistic agency, the deliberate choices people make to protect their identity (Gee, 2014) or to accomplish social goals (Martin and Rose, 2003). The latter encompasses involuntary cues which signals their disposition, like mental illness (Kayi et al., 2017; Alqahtani et al., 2022), personality (Mairesse et al., 2006; Moreno et al., 2021), attitude (Martin and White, 2003), or emotion (Hazarika et al., 2018).

Social meaning is thus defined as the signaling people do during interactions to maintain

positioning in terms of identity and relationship. While originally defined in the context of socio-linguistics, the term "social meaning" has been heavily used in the computational linguistics community. It can refer to different ways or styles people interact (Jurafsky et al., 2009), or the social background and identity of a user that can be predicated from linguistic variation (Nguyen et al., 2021a), or the meaning that emerges through human interaction on social media in the form of emotion, sarcasm, irony and the like (Zhang and Abdul-Mageed, 2022).

Given the myraid definitions, we use "social meaning detection tasks" as an umbrella term to refer to tasks that involve understanding the users behavior or their characteristics in a social setting. Since social meaning is subtly encoded, traditional classification models often over-fit to context-specific linguistic elements that correlate with these subtle cues only within context. Consequently, this makes transfer to unseen domains and subsequently to unseen tasks especially challenging. We thus hone in on social meaning detection tasks as a good test bed to investigate generalization, both in a cross-domain setting as well as in a cross-task setting.

7.3 Cross Domain Generalization

We first investigate the capabilities of models to transfer across different domains for the same social meaning detection task. For example, the same strategy to resist persuasion attempts would manifest in different ways, depending on whether one is negotiating the price of a commodity, or one is hesitating donating to charity (Dutt et al., 2021). We thus realize our investigation in two tasks namely resisting strategies detection and emotion recognition.

7.3.1 Resisting Strategies Detection (RES)

Motivation

The use of persuasion strategies to change a person's view or achieve a desired outcome finds several real-world applications, such as in election campaigns (Knobloch-Westerwick and Meng, 2009; Bartels, 2006), advertisements (Speck and Elliott, 1997), and mediation (Cooley, 1993). Consequently, several seminal NLP research have focused on operationalizing and automatically identifying persuasion strategies (Wang et al., 2019b), propaganda techniques (Da San Martino et al., 2019), and negotiation tactics (Zhou et al., 2019), as well as the impact of such strategies on the outcome of a task (Yang et al., 2019; He et al., 2018c; Joshi et al., 2021b). We view the task from a different perspective i.e. investigating resisting strategies to foil persuasion.

Resisting strategies have been widely applicable in marketing (Heath et al., 2017), cognitive psychology (Zuwerink Jacks and Cameron, 2003), and political communication (Fransen et al., 2015b) . Some notable works include the identification and motivation of commonly-used resisting strategies (Fransen et al., 2015a; Zuwerink Jacks and Cameron, 2003), the use of psychological metrics to predict resistance (San José, 2019; Ahluwalia, 2000), and the design of a framework to measure the impact of resistance (Tormala, 2008). As opposed to these qualitative methods, we adopt a data-driven approach, and propose a generalised framework to characterise resisting strategies and employ state-of-the-art neural models to infer them automatically.

Datasets

We choose persuasion-oriented conversations, rather than essays or advertisements (Yang et al., 2019), since we can observe how the participants respond to the persuasion attempts in real-time. To that end, we leverage two publicly available corpora on persuasion (Wang et al., 2019b) and negotiation (He et al., 2018c). We refer to these datasets as "Persuasion4Good" or P4G and "Craigslist Bargain" or CB henceforth.

P4G comprises conversational exchanges between two anonymous Amazon Mechanical Turk workers with designated roles of the persuader, ER and persuadee, EE. ER had to convince EE to donate a part of their task earnings to the charity *Save the Children*. We investigate the resisting strategies employed only by EE in response to the donation efforts. We emphasise that the conversational exchanges are not scripted, and the task is set up so that a part of EE's earnings is deducted if they agree to donate. Since there is a monetary loss at stake for EE, we expect them to resist.

CB consists of simulated conversations between a buyer (BU) and a seller (SE) over an online exchange platform. Both are given their respective target prices and employ resisting strategies to negotiate the offer.

We choose these datasets since they involve non-collaborative goal-oriented dialogues. As a result, we can definitively assess the impact of different resisting strategies on the goal.

Framework Description

We briefly describe the resisting strategies commonly referenced in social and cognitive psychology literature. This enables us to design a unified framework for the two datasets, built upon common underlying semantic themes. Fransen et al. (2015a) identified 4 major clusters of resisting strategies, namely **contesting** (Wright, 1975; Zuwerink Jacks and Cameron, 2003; Abelson and Miller, 1967), **empowerment** (Zuwerink Jacks and Cameron, 2003; Sherman and Gorkin, 1980), **biased processing** (Ahluwalia, 2000), and **avoidance** (Speck and Elliott, 1997). Each individual category can be subdivided into finer categories showcased in italics henceforth.

Contesting refers to attacking either the source of the message (*Source Derogation*) or its content (*Counter Argumentation*). A milder form of contesting involves seeking clarification or information termed *Information Inquiry*. Prior work has shown a positive association between working knowledge and one's ability to resist persuasion (Wood and Kallgren, 1988; Luttrell and Sawicki, 2020). Therefore, *Information Inquiry* can be interpreted as a form of resistance where the resistor seeks to satisfy their doubts because they are sceptical of the persuader's intents or messages. This is prominent in certain conversations in P4G where a sceptical EE questions the charity's legitimacy.

Empowerment strategies encompass reinforcing one's personal preference to refute a claim (*Attitude Bolstering*) (Sherman and Gorkin, 1980), attempting to arouse guilt in the opposing party (*Self Pity*) (Vangelisti et al., 1991; O'Keefe, 2002), stating one's wants outright (*Self Assertion*) (Zuwerink Jacks and Cameron, 2003), or seeking validation from like-minded people (*Social Validation*) (Fransen et al., 2015a). Overall, empowerment strategies drive the discussion towards the resistor's self as opposed to attacking the persuader.

Biased processing mitigates external persuasion by selectively processing information that

conforms with one's opinion or beliefs (Fransen et al., 2015a). For simplicity, we subsume strategies that denote personal preference, namely *Attitude Bolstering* and *Biased Processing*, into a unified category *Personal Choice*. We refrain from incorporating *Self Assertion* into the *Personal Choice* category since it deals with bolstering one's confidence and not one's opinions or attitudes. The subtle difference is highlighted in Table 7.1.

Avoidance strategies distance the resistor from persuasion, either physically or mechanically, or refuse to engage in topics that induce cognitive dissonance (Fransen et al., 2015a). However, in the context of task-oriented conversations, wherein participants are expected to further a goal, avoidance often manifests as *Hesitance* to commit to the current situation.

We identify seven major resisting strategies across the datasets, namely *Source Derogation, Counter Argumentation, Information Inquiry, Personal Choice, Self Pity, Hesitance,* and *Self Assertion*. Since the datasets comprise two-party conversations between strangers, *Social Validation*, which requires garnering the support of others, was absent. We now describe how these resisting strategies were instantiated in the following section.

Instantiating the Resisting Strategies Framework

We emphasise that although the description and meaning of a strategy remain the same across the two datasets, their semantic interpretation depends on the context, **making them prime candidates for investigating transfer**. For example, scepticism towards the charity in P4G and criticism of the product in CB are instances of *Source Derogation*. This is because ER represents the charity, whereas the seller is being accused of selling an inferior product. Likewise, we instantiate the predicates for the remaining six resisting strategies for the two datasets, with examples in Table 7.1.

We label the utterances of persuadee (EE) in P4G and the buyers (BU) and sellers (SE) in CB with at least one of the seven corresponding resisting strategies, or 'Not-A-Strategy' if none applies. The 'Not-A-Strategy' label includes greetings, off-task discussions, agreement, compliments, or other tokens of approval. We acknowledge that an utterance can have more than one resisting strategy embedded in it. For example, the utterance "The price is slightly high for used couches, would you come down to 240 if I also picked them up?", is an instance of both *Personal Choice* and *Counter Argumentation*.

We also note that ***Information-Inquiry* is not a resisting strategy for CB** since asking additional information/clarification is an expected behaviour before finalising a deal. We keep the label nevertheless to show comparison with P4G.

Annotation Procedure and Validation

We describe the annotation procedure for both the CB and P4G dataset here and its subsequent validation. For CB, three authors independently annotated five random conversations adhering to the flowchart. If the conversations chosen were simple or had few labels, a new set of 5 conversations were taken up. This constitutes one round. After each round, the Fleiss Kappa score was computed, and the authors discussed to resolve the disagreements and revise the flowchart. Then began the next round on a new set of 5 random conversations. For CB, 5 rounds of revision were carried out over 24 conversations, until a high Fleiss kappa (0.790) (Fleiss, 1971) was

Table 7.1: Framework describing the resisting strategies for persuasion (P4G) and negotiation (CB) datasets, as specified in Dutt et al. (2021). Examples of each strategy are italicised. The examples for each of P4G and CB were borrowed from the original datasets of the same name from Wang et al. (2019a) and He et al. (2018a) respectively.

Resisting Strategy	Persuasion (P4G)	Negotiation (CB)
Source Derogation	Attacks/doubts the organisation's credibility. *My money probably won't go to the right place*	Attacks the other party or questions the item. *Was it new denim, or were they someone's funky old worn out jeans?*
Counter Argument	Argues that the responsibility of donation is not on them or refutes a previous statement. *There are other people who are richer*	Provides a non-personal argument/factual response to refute a previous claim or to justify a new claim. *It may be old, but it runs great. Has lower mileage and a clean title.*
Personal Choice	Attempts to saves face by asserting their personal preference such as their choice of charity and their choice of donation. *I prefer to volunteer my time*	Provides a personal reason for disagreeing with the current situation or chooses to agree with the situation provided some specific condition is met. *I will take it for $300 if you throw in that printer too.*
Information Inquiry	Ask for factual information about the organisation for clarification or as an attempt to stall. *What percentage of the money goes to the children?*	Requests for clarification or asks additional information about the item or situation. *Can you still fit it in your pocket with the case on?*
Self Pity	Provides a self-centred reason for not being able/willing to donate at the moment. *I have my own children*	Provides a reason (meant to elicit sympathy) for disagreeing with the current terms. *$130 please I only have $130 in my budget this month.*
Hesitance	Attempts to stall the conversation by either stating they would donate later or is currently unsure about donating. *Yes, I might have to wait until my check arrives.*	Stalls for time and is hesitant to commit; specifically, they seek to further the conversation and provide a chance for the other party to make a better offer. *Ok, would you be willing to take $50 for it?*
Self-assertion	Explicitly refuses to donate without even providing a factual/personal reason *Not today*	Asserts a new claim or refutes a previous claim with an air of finality/ confidence. *That is way too little.*

obtained. Finally, the three authors independently went ahead and annotated approximately 250 distinct conversations, yielding a corpus of 800 CB conversations. Our annotation procedure requires a rigorous reliable refinement phase but a comparatively faster annotation phase by dividing the annotation between the authors. Thus the conversations annotated by each author were mutually exclusive. Similarly, for P4G dataset, four authors annotated 3 conversations per round, since a conversation in P4G was comparatively longer. 4 rounds of revision across 12 conversations was done to achieve the final kappa-score of 0.787.

7.3.2 Emotion Recognition in Conversations (ERC)

Motivation

While emotion understanding has been a long-standing goal in NLP and affective computing, emotion recognition in conversations (ERC) adds an extra layer of difficulty since the unit of prediction is an utterance embedded in an evolving dialogue. Early multimodal resources such as IEMOCAP highlighted that even in controlled dyadic interactions, emotion is expressed through a combination of verbal and non-verbal signals and involves reliance on surrounding context rather than isolated sentences (Busso et al., 2008). More recently, ERC has grown into an active research area with standardized benchmarks such as EmotionLines and the EmotionX shared task, as well as large-scale multi-party datasets such as MELD, which explicitly frame the problem as identifying the emotion of each dialogue turn in context (Poria et al., 2019; Hsu and Ku, 2018).

A core motivation for ERC is that the same surface form can realize different emotions depending on conversational history, speaker identity, and discourse dynamics. For example, a short utterance like "fine" may signal neutrality, resignation, irritation, or sarcasm depending on what preceded it and who said it; similarly, emotions often shift across turns as new information arrives or interpersonal tension escalates. Consequently, ERC systems should be designed to take into account context propagation across turns, inter-speaker relationships, and crucially implicit commonsense or pragmatic cues (Majumder et al., 2019; Raheja and Tetreault, 2019; Ghosal et al., 2020). The need for deeper contextual modeling in ERC makes it a perfect test-bed for investigating the social meaning detection capabilities of different NLP systems.

Datasets

ERC is realized via two representative datasets namely "IEMOCAP" (Busso et al., 2008) and the "Friends" dataset of (Hsu et al., 2018).

The IEMOCAP dataset of Busso et al. (2008) comprises dyadic (two-speaker) interactions performed by trained actors in both scripted and improvised scenarios designed to elicit emotion. Beyond transcripts, it is explicitly multimodal, pairing speech and video with motion-capture signals (e.g., facial/head/hand markers), making it a common testbed for ERC settings where emotion is expressed not only through words but also through prosody and nonverbal behavior.

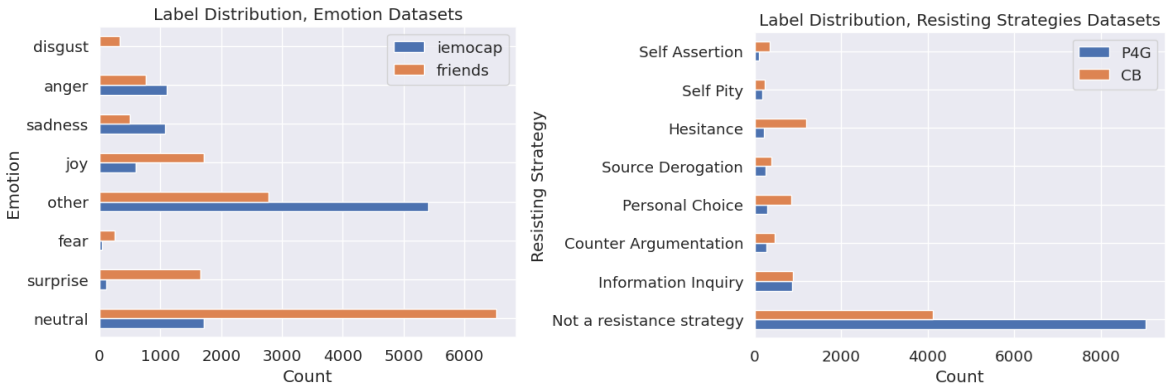
The Friends datset of (Hsu et al., 2018) is a text-only, multi-party ERC dataset built from TV scripts of Friends, where the creators treat each scene as a dialogue and each character line as an utterance within that dialogue context. Every utterance is labeled with one of the eight emotions including Ekman's six basic emotions, neutral, and other by Amazon Mechanical Turk annotators,

Table 7.2: Framework describing the emotion labels in the emotion recognition datasets (IEMOCAP and Friends) (Busso et al., 2008; Poria et al., 2019). Examples of each label are italicised.

Emotion	IEMOCAP	Friends
Neutral	Neutral emotion is characterized by the absence of strong feelings or emotions. *I'll go to basketball games.*	Neutral emotion is characterized by the absence of strong feelings or emotions. *Yeah, apparently they're turning it into some kinda coffee place.*
Joy	Joy is a feeling of extreme gladness, delight, or exultation of the spirit arising from a sense of well-being or satisfaction. *I don't know it seemed like a pretty good spot to me. Look at the moon - view the moon view I got from here.*	Joy is a feeling of extreme gladness, delight, or exultation of the spirit arising from a sense of well-being or satisfaction. *I'm so proud of you.*
Sadness	Sadness is an emotional state of unhappiness, ranging in intensity from mild to extreme and usually aroused by the loss of something that is highly valued *Augie, I'm sorry.*	Sadness is an emotional state of unhappiness, ranging in intensity from mild to extreme and usually aroused by the loss of something that is highly valued *Uh, well... Joey and I broke up.*
Surprise	Surprise is an emotion typically resulting from the violation of an expectation or the detection of novelty in the environment. *Shut up. No- in Vegas?*	Surprsie is an emotion typically resulting from the violation of an expectation or the detection of novelty in the environment. *Oh my God, wh-what happened?*
Fear	Fear is a basic, intense emotion aroused by the detection of imminent threat, involving an immediate alarm reaction that mobilizes the organism by triggering a set of physiological changes. *Good God.*	Fear is a basic, intense emotion aroused by the detection of imminent threat, involving an immediate alarm reaction that mobilizes the organism by triggering a set of physiological changes. *Oh boy, I just can't watch. It's too scary!*
Disgust	Disgust is characterized by strong aversion to something deemed revolting, or toward a person or behavior deemed morally repugnant. *It was a terrible thing. I hated it.*	Disgust is characterized by strong aversion to something deemed revolting, or toward a person or behavior deemed morally repugnant. *Ew! What is that? Something exploded!*
Other	An emotion or feeling which does not include anger, surprise, sadness, joy, fear, or disgust. *How long did that row last?*	An emotion or feeling which does not include anger, surprise, sadness, joy, fear, or disgust. *Oh well, okay, good luck.*

	ERC		RES	
	Friends	**IEMOCAP**	**P4G**	**CB**
Dialogues	1000	151	473	713
Total datapoints	14503	10039	11260	8511
Labels	8	8	8	8
Avg. Turns/Dialogue	14.50	66.49	36.05	11.94
Avg. Words/Turn	7.83	11.57	9.22	12.38

Table 7.3: We present here the statistics of the datasets for cross domain generalization for both ERC and RES .



(a) Label Distribution in the emotion datasets(b) Label Distribution for the resisting strategies datasets

Figure 7.1: We present here the label distribution for the emotion recognition and the resisting strategies datasets.

who were explicitly instructed to consider the full dialogue context when assigning labels, with majority vote used to determine the gold emotion.

Similar to the RES, both ERC datasets operate over the same eight labels, but are instantiated in different contexts, and thus provides a valid test bed to investigate cross-domain generalization.

7.4 Cross Task Generalization

We also inspect the generalization capabilities of models across different kinds of social meaning detection tasks, rather than across domains. We thus realize them for six dialogue understanding tasks, each of which is instantiated with a distinct dataset, such that each task operates over a distinct domain. Moreover, these datasets have unique labels or categories to prevent any overlap between them. Such a setting would enable us to inspect the cross-task generalizability of models where a given model is trained for one task and then evaluated on another.

7.4.1 Datasets and Tasks

We describe in detail the six different datasets (or tasks) that we explore for cross-task generalization below.

1. Persuasion - The task involves identifying persuasive strategies between two AMT workers where one adopts the role of the persuader and is expected to convince the other party (the persuadee) to donate to charity. We use the Persuasion for Good (P4G) dataset of Wang et al. (2019a).
2. Negotiation tactic - The negotiation task is grounded in the CaSiNo corpus of (Chawla et al., 2021), which consists of bargaining for campsite resources between crowd workers in a simulated camping setting. Dialogs contain various aspects of a realistic negotiation, such as building relationships, discussing preferences, exchanging offers, emotional expression, and persuasion with personal and logical arguments.
3. Resisting Strategies - Complementary to task of identifying persuasive attempts, the task proposed by Dutt et al. (2021) involves detecting resisting strategies, i.e. strategies employed to resist being persuaded by others. We focus on the Craigslist Bargain dataset (henceforth res_CB) which consists of simulated conversations between a buyer (BU) and a seller (SE) over an online exchange platform. Both are given their respective target prices and employ resisting strategies to negotiate the offer.
4. Empathy in mental health - We use the framework and dataset of Sharma et al. (2020) that characterizes the communication of empathy in text-based conversations. The task involves detecting different dimensions of empathy in text-based mental health support, i.e., empathy expressed or communicated by peer supporters in their textual interactions with seekers.
5. Argumentation - We formalize the task of argumentation into identifying different kinds of proposition in rhetorical debates. We use the data set of Jo et al. (2020) which consists of four categories of propositions: normative statements, desires statements, statements about future possibilities, and reported speech.
6. Implicit Hate Speech Detection - The task involves identifying different categories of covert or indirect language that disparages a particular individual or group based on certain protected attributes (ElSherief et al., 2021). Some instances include irony, inferiority language, and incitement to violence, among others.

We also provide descriptions of the label categories for each dataset along with an example of each for res_CB, Casino, EMH, PROP, IMP_HATE, and P4G in the Tables 7.1, 7.4,7.5, 7.6, 7.7, and 7.8 respectively.

7.4.2 Statistics

We present an overview of the statistics of the six datasets/tasks in Table 7.9 in terms of number of utterances, the mean utterance length (or words per turn), the mean context length (or turns per dialogue), and number of labels. For example, IMP_HATE consists of individual stand-alone utterances and hence has no preceding dialogue history and thus has an average turn of 0. We further show the distribution of these statistics across the train, validation, and test splits for

Table 7.4: Description of the negotiation strategies used in our work for Casino (Chawla et al., 2021). Examples of each strategy are italicised.

Negotiation Label	**Description**
self-need	Participant argues for creating a personal need for an item in the negotiation. *Yes. I'm actually taking a large group of people. Some friends and family are going and I kind of also wanted a bit of extra firewood.* :)
no-need	Participant points out that they do not need an item based on personal context. *I don't like food. my stomach is always full. I only drink water since im thirsty most of the time.*
promote-coordination	Participant promotes coordination between the two partners. *Alright so I think we can make a fair deal here where we both will be happy.* :)
small-talk	Participant engages in small talk while discussing topics apart from the negotiation in an attempt to build a rapport. *My mistake, hypothermia is messing with my brain.*
uv-part	Participant undermines the requirements of their opponent. *I understand that atleast you are going to be close to water, that will be our most important thing since we will be thirsty and you know kids and trying to tell them to ration the water...LOL*
elicit-pref	Participant provides an attempt to discover the preference order of the opponent *I get that and understand completely. I have a large number of mouths to feed making the food a necessity or all the firewood to cook whatever we hunt. How many you have?*
vouch-fair	Participant announces a callout to fairness for personal benefit, either when acknowledging a fair deal or when the opponent offers a deal that benefits them *hey buddy I hope we both end up with a good deal* :)
other-need	Participants discuss a need for someone else rather than themselves. *I would be willing to do that if I could have two of the waters? I didn't bring as much as I thought I would need because I forgot I would have my dog.*
showing-empathy	Participant positively acknowledges or displays empathetic behavior towards a personal context of the partner. *Are you sure that's enough firewood for you and the baby? I know that babies can easily get very sick from dropping temperatures.*
non-strategic	Utterance does not have any strategic element *oh well that's fantastic, so let me know what your deal is*

Table 7.5: Description of the different dimensions of empathy used in our work for EMH (Sharma et al., 2020). Examples of each strategy are italicised.

Empathy Dimension	**Description**
emotion	Responder specifies the experienced emotions explicitly or communicates emotional reactions without the emotions being explicitly labeled. *I love you all. Genuinely the replies here made me feel much less alone. your kindness is beautiful*
interpretation	Responder specifies the inferred feeling or experience, or communicates understanding through descriptions of similar experiences, or mentions that they understand the situation. *I struggle with the same thing, but with my step dad. I've basically given up on it.*
exploration	Responder seeks to explore the specific experiences and feelings of the supporter, which can be specific to the situation or generic. *What kind of connections do you want / need?*

Table 7.6: Description of the argumentation labels used in our work for PROP (Jo et al., 2020). Examples of each strategy are italicised.

Argumentation Label	**Description**
Normative	A normative proposition is defined as a proposition where the speaker or someone else proposes that a certain situation should be achieved or that an action should be carried out. *Americans have a right to know if there are any conflicts of interest*
Desire	A desire proposition is defined as a proposition that explicitly claims that the speaker or someone else desires to own something, do something, or desires for a certain situation to be achieved. *Bernie Sanders' soft stance on guns is not what Americans want to hear right now*
Future Possibility	A future possibility proposition claims a possibility or prediction that something may be the case in the future. *the next president of the US could very well be a reality TV star*
Reported Speech	A reported speech proposition must convey an explicit or implicit predicate borrowed from a source external to the speaker. *Clinton's view that sometimes capitalism needs controls in order to deal with inequality were a lot more realistic*

Table 7.7: Description of the argumentation labels used in our work for IMP_HATE (ElSherief et al., 2021). Examples of each strategy are italicised.

Type of Hate	**Description**
white_grievance	The message includes frustration over a minority groups perceived privilege and casting majority groups as the real victims of racism. This language is linked to extremist behavior and support for violence *is zuma talking about seizing white land or not ? are you denying it ?*
incitement	The message includes flaunting in group unity and power or elevating known hate groups and ideologies. *india wud be a peaceful prosperous n progressive country only when each of its citizen is a hindu.that shud b d endall objective 4 d state*
inferiority	The message implies one group or individual is inferior to another, and it can include dehumanization, denial of a person's humanity, and toxic language that compares the target with disease, insects, animals . Related to assaults on human dignity, dominance, and declarations of superiority of the in group *have you ever written about islam being a crutch for a failed civilization? im new to your work, just wondering.*
stereotypical	The message associates a protected class with negative attributes such as crime, or terrorism or includes misinformation that feeds stereotypes and vice versa, like holocaust denial and other forms of historical negationism. *You can't be a person of colour; you're too pale!*
irony	The message uses sarcasm, humor, and satire to attack or demean a protected class or individual. *What's the one good thing about black people? They provide jobs for the prison guards*
threatening	The message conveys a speaker's commitment to a target's pain, injury, damage, loss or violation of rights, threats related to implicit violation of rights and freedoms, removal of opportunities, and more subtle forms of intimidation. *We have this huge military. Why don't we just go down there and create an ethno-state for whites. Most of the blacks weren't even there when South Africa was founded by whites!*

Table 7.8: Description of the persuasion labels used in our work for P4G(Wang et al., 2019a). Examples of each strategy are italicised.

Persuasion Label	**Description**
credibility-appeal	Refers to the uses of credentials and citing organizational impacts to establish credibility and earn the persuadee's trust *It is the worlds first global charity for children, and have credentials to back them up.*
logical-appeal	Refers to the use of reasoning and evidence to convince others. *You are donating money you don't even have yet so it is not like you are missing something.*
foot-in-the-door	Refers to the strategy of starting with small donation requests to facilitate compliance followed by larger requests." *Are you sure, you can do as little as 5 cents???*
emotion-appeal	Refers to the elicitation of specific emotions to influence others in the form of story-telling, empathy, guilt, or anger" *It broke my heart to see that famous photograph of a child with a vulture sitting next to it.*
personal-story	Refers to the strategy of using narrative exemplars to illustrate someone's donation experiences or the beneficiaries' positive outcomes, which can motivate others to follow the actions." *I have three children myself, and the welfare of children around the world is a very important cause to me.*
self-modeling	Refers to the strategy where the persuader first indicates their own intention to donate and chooses to act as a role model for the persuadee to follow" *I think I am going to give a small portion of my hit payment to save the children.*
donation-information	Refers to providing specific information about the donation task, such as the donation procedure, donation range, etc." *The research team will collect all donations and send it to Save the Children.*
source-related-inquiry	Asks about the persuadee's opinion and expectation related to the task." *Iḿ alright, just reading up on this organization called "Save the Children".. have you heard about it?*
task-related-inquiry	Asks if the persuadee is aware of the organization (charity) *Do you need more info about this program?*
personal-related-inquiry	Asks about the persuadee's previous personal experiences relevant to charity donation" *I imagine hospitals are very strict about who gets to be with the little ones.*
other	Does not conform to any persuasion category *I am homeless and at Mcdonalds on the wifi.*

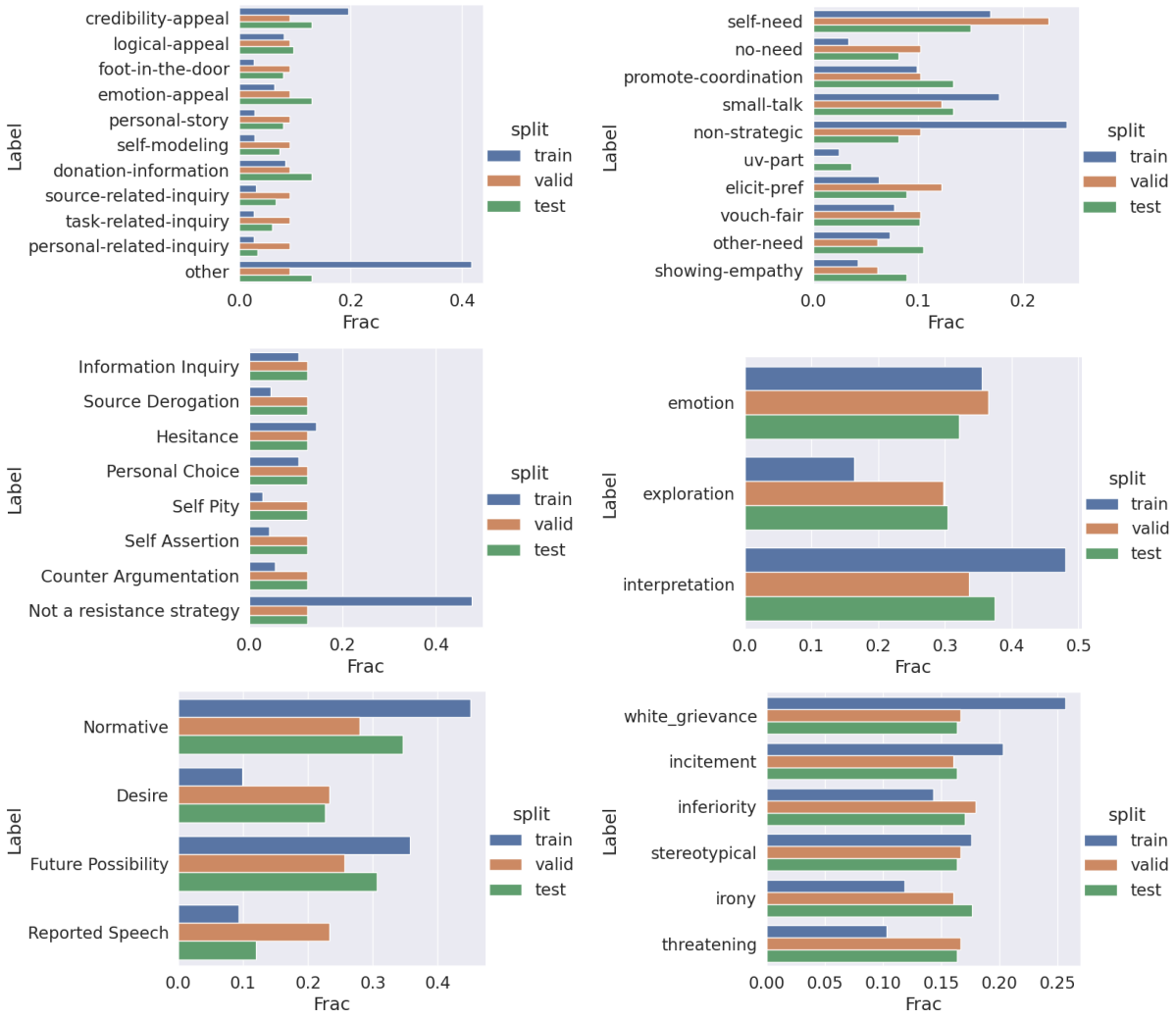


Figure 7.2: Distibution of labels across the different splits for the six datasets or tasks.

Dataset	Avg Words per Turn	Avg Turns per Dialog	# Turns	# Labels
P4G (Wang et al., 2019a)	10.75 / 13.76 / 11.53	18.74 / 15.45 / 17.9	4004 / 110 / 154	11 / 11 / 11
CaSiNo (Chawla et al., 2021)	21.53 / 20.29 / 26.50	5.42 / 4.88 / 5.02	4862 / 49 / 247	10 / 9 / 10
Res_CB (Dutt et al., 2021)	12.22 / 13.63 / 13.71	5.86 / 5.18 / 6.09	6348 / 160 / 160	8 / 8 / 8
PROP (Jo et al., 2020)	12.55 / 14.86 / 15.71	11.66 / 9.47 / 12.21	741 / 43 / 75	4 / 4 / 4
EMH (Sharma et al., 2020)	54.03 / 47.75 / 53.83	1 / 1 / 1	1823 / 104 / 112	3 / 3 / 3
IMP_HATE (ElSherief et al., 2021)	15.79 / 17.18 / 15.39	0 / 0 / 0	3182 / 156 / 153	6 / 6 / 6

Table 7.9: Dataset statistics across the train, validation, and test splits for different cross-task generalization tasks.

different tasks. Additionally, we present the distribution of different label categories for the different datasets in Table 7.2. Overall, we observe a wide variation for different statistics across the different datasets.

7.5 Rationales as Informal Scaffolds

Computational modeling of human communication in social interactions is challenging since most communication employs indirect language whose meaning goes beyond the literal form of the text (Yerukola et al., 2024; Yusupujiang and Ginzburg, 2023; Markowska et al., 2023; Dutt et al., 2024). For the current instance, taken from the IMP_HATE dataset, "we must resist ebolaphobia. these viruses just come here for a better life, to do jobs that american viruses refuse to do", it is necessary to uncover the sarcastic intentions of the speaker to infer the implicit hate toward immigrants. Recognizing such subtle cues is crucial for many tasks, e.g., automated content moderation (Calabrese et al., 2024; Horta Ribeiro et al., 2023), intent resolution (Yerukola et al., 2024; Joshi et al., 2021a), amongst others (Kim et al., 2024; Qian et al., 2024).

Subsequently, designing systems to effectively solve the myriad of social meaning detection tasks involves understanding the underlying meaning during communication. We use the term "rationales" as a broad umbrella term to capture the implicit social meaning behind a message. While prior work have used rationales to refer to "task-specific explanations" for NLI and common-sense reasoning (Rao et al., 2023; Zelikman et al., 2022), or to explain model predictions (Wiegreffe et al., 2021). However, such a definition fails to capture pragmatic aspects of the message and thus sets us apart since we use rationales to refer to the elicited social meaning, i.e. why and how an utterance was conveyed in dialogue.

The idea behind rationales is that they are designed to break through the opaque surface form of the conversation's text and make the social cues more transparent. Since dialogues are often under-specified (Sap et al., 2022), we hypothesize that rationales can serve as effective augmentations to enrich the dialogue context and thus facilitate generalization. An important consideration is for the rationales to be task-agnostic so that they can be seamlessly generated for a given dialogue without considering the prior particulars of the task.

7.6 Conclusion and Takeaways

We thus define informal scaffolds and ground them as "rationales" i.e. free-form textual explanations designed to capture subtle pragmatic cues in conversations. We also motivate social meaning detection as a valid stress-test for generalization in dialogue since social meaning is subtly encoded, and models that rely on surface cues are prone to over-fitting while dealing with such tasks. We further outline two complementary generalization settings (cross-domain and cross-task) and realize them using different datasets. In the subsequent chapters, we operationalize the generation of rationales to capture different perspectives or kinds of social meaning and how the generated rationales can facilitate generalization across domains in Chapter 8 and across tasks in Chapter 9.

Chapter 8

Rationales for Cross Domain Generalization in Conversations

We explore the capabilities of machine-generated rationales to facilitate cross-domain generalization for two social meaning detection tasks, i.e. identifying resisting strategies (RES) and recognizing emotions in conversations (ERC). We propose a prompting framework to generate different kinds of rationales and subsequently examine their validity. We then describe our experimental setup namely the in-domain and transfer settings and the corresponding models and metrics for each setup. Finally, we present our quantitative results and qualitative analyses.

8.1 Prompting Framework

In this section, we propose a prompting framework to generate rationales that can capture the underlying social meaning and assess their validity. We showcase our prompting framework in Figure 8.1.

8.1.1 Prompt Design Motivation

The design for our prompts was grounded in Goffman (2002)'s notion of social meaning in language; the intentional and the implied. Dialogue understanding relies on pragmatic reasoning to recognize subtle clues that are *implicit* or obscured by the surface form, often thought of as "reading between the lines". Accurate interpretation also includes what *assumptions* underlie the choices made by the speaker, and choices that may reveal aspects of the speaker's *intentions*.

Motivated by this conceptualization of social meaning, we prompt the LLM to generate rationales that adhere to the speaker's intention, their underlying assumptions, and any implicit information present in the conversation (henceforth referred to as INT, ASM, and IMP respectively). We briefly describe the three different rationales below.

(i) Intention (INT) refers to the underlying purpose or goal that a speaker seeks to achieve or communicate. It captures the deliberate messages conveyed in the dialogue.

(ii) Assumptions (ASM) refer to the biases or presumptions that the speaker holds. They often reflect the speaker's background, experiences, societal norms, and unacknowledged biases.

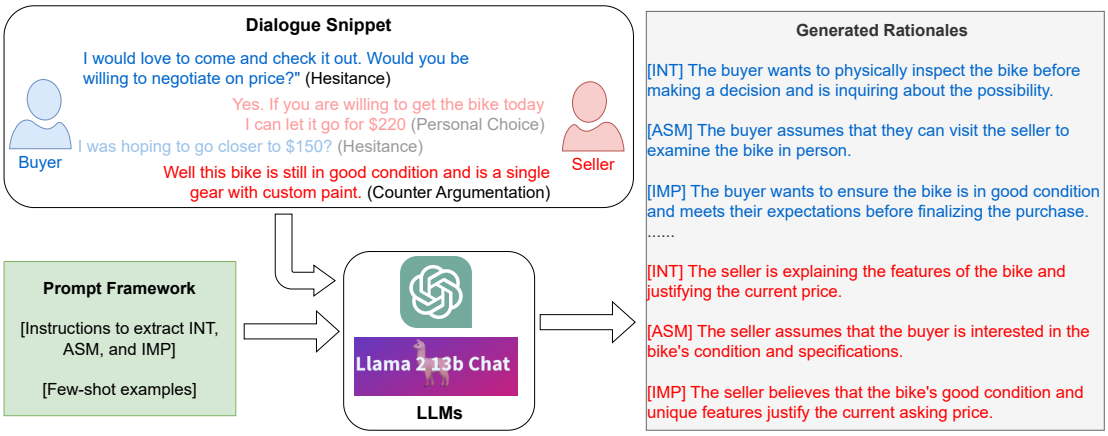


Figure 8.1: We present the prompting framework employed in this work to generate rationales that are subsequently used for dialogue understanding and transfer using pre-existing LLMs such as GPT-3.5-turbo and LLama-2 variants. We feed in the prompt (green box on the left) for a given dialogue to generate the speaker's intentions (INT), assumptions (ASM), and the underlying implicit information (IMP) (gray box in the right). For lack of space we showcase the generated rationales only for the first (in blue) and last utterance(in red).

(iii) Implicit Information (IMP) encompasses the information that, while not overtly expressed, is inferred or understood within the context of the conversation. It offers essential cues about the conversation and its nuances.

8.1.2 Structured Prompting

We adopt a "structured prompting" approach inspired by recent work that craft prompts in a code-like-manner, such as utilizing python's dictionary data structure (Jung et al., 2023; Madaan et al., 2022b) or as pseudo-code (Mishra et al., 2023). In our case, the prompt had the following four components, namely (i) description of the high-level task, i.e. analysis of social meaning in dialogue, (ii) instructions that outline the generation of rationales, i.e. the elicitation of speaker's intention, assumptions, and implicit information (i.e. INT, ASM, and IMP) in a procedural manner, (iii) an output template that specifies the format in which the response is to be structured, and (iv) examples of input-output pairs consistent with the template.

We observed that prompting LLM to generate all three rationales (INT, ASM, and IMP) together, facilitated instruction following. Hence we term our approach as "multi-faceted prompting". These rationales were augmented with the conversational text for two downstream social meaning detection tasks. We provide examples of prompts for the two tasks in Tables 8.1 and 8.2.

8.1.3 Dialogue Context & In-Context Examples

Even for humans, understanding an individual utterance is challenging in absence of the situated dialogue context. Consequently, for our prompting framework, we provide each utterance with the corresponding dialogue history in the form of the five preceding utterances. During development

Table 8.1: Below is an example of our prompt for the task of emotion recognition in conversations (ERC).

Part 1: High level description of the objective	Analyze this dialogue, focusing on any underlying assumptions and implicit information.
Part 2: Instructions	For the final utterance, provide a comprehensive and concise explanation for: a) Speaker's Intention, b) Assumptions about the conversation, and c) Implicit Information
Part 3: Output Template	Please format your response as follows: Speaker's Intention in the final utterance: ¡your response¿ Assumptions about the conversation in the final utterance: ¡your response¿ Implicit Information in the final utterance: ¡your response¿
Part 4: Examples for ICL	Dialogue history: The Interviewer: You must've had your hands full. Chandler: That I did. That I did. The Interviewer: So let's talk a little bit about your duties. Chandler: My duties? All right. Final utterance: The Interviewer: Now you'll be heading a whole division, so you'll have a lot of duties. Speaker's Intention in the Final Utterance: The speaker's intention in the final utterance is to inform Chandler about his upcoming role and the responsibilities associated with it. The speaker is preparing Chandler for a new position. Assumptions about the conversation in the Final Utterance: The assumption here is that Chandler is about to take on a leadership role within the organization, specifically heading a whole division. The speaker assumes that Chandler needs to be aware of the increased responsibilities that come with this new position. Implicit Information in the Final Utterance: The implicit information in the final utterance is that Chandler has been promoted or assigned a higher-level job within the company. Additionally, it suggests that the speaker expects Chandler to be prepared to handle the increased workload and responsibilities that come with leading a division. ...

Table 8.2: Below is an example of our prompt for the task of detecting resisting strategies (RES).

Part 1: High level description of the objective	Analyze this dialogue, focusing on any underlying assumptions and implicit information. Ensure that you address each line individually without skipping or grouping.
Part 2: Step-wise guide	For each line: 1. Provide a comprehensive and concise explanation for: a)Speaker's Intention b)Assumptions about the conversation c)Implicit Information 2. Continue until you have analyzed every line.
Part 3: Output Template	Please format your response as follows: Speaker's Intention: ¡your response¿ Assumptions about the conversation: ¡your response¿ Implicit Information: ¡your response¿
Part 4: Examples for ICL	INPUT: ... Persuadee: They are hungry and injured and also short. Persuader: I'm so sorry, what a terrible thing. ... Output: ... Speaker's Intention: The Persuadee provides additional details about their child's situation, emphasizing the child's needs. Assumptions about the conversation: The Persuadee assumes that sharing these specific details will elicit a stronger empathetic response from the Persuader. Implicit Information: The Persuadee seeks empathy and understanding from the Persuader regarding their child's dire circumstances. Speaker's Intention: The Persuader expresses sympathy and acknowledges the gravity of the Persuadee's situation. Assumptions about the conversation: The Persuader assumes that offering sympathy and acknowledging the seriousness of the situation is an appropriate response. Implicit Information: The Persuader expresses compassion and understanding toward the Persuadee's plight. ...

process, we experimented with different context turns, and five achieved the best result.

Furthermore, since LLMs are effective few-shot learners (Wei et al., 2022a), we also provide the prompts with a few in-context examples to improve response generation. These in-context examples were generated using GPT4 (Achiam et al., 2023).

8.1.4 Validity of Generated Rationales

Table 8.3: Fraction of times ChatGPT-3.5-turbo-16k was chosen over LLama-2-13B-chat based on the quality of the generated rationales.

	CB	P4G	Iemocap	friends
S1	15	16	12	16
S2	13	15	14	19
S3	13	11	12	12
Overall	15	16	12	17

To assess the quality of the generated rationales, we prompted two prevalent pre-trained LLMs in contemporaray NLP research; GPT-3.5-turbo-16k or ChatGPT[1] and the Llama2-13B-Chat (Touvron et al., 2023) to generate rationales. We sampled 20 instances from each dataset (80 in total) to compare the generation quality of the models. The assessment, which involved choosing the output with a higher quality, was carried out by three graduate students proficient in English. The results of our experiments present in Table 8.3 of the Appendix showcases that annotators prefer the ChatGPT model 75% of the times, and hence we adopted it as the LLM of our choice for subsequent experiments.

Furthermore, to measure the generation quality, we provided two annotators with the aforementioned 80 rationales and asked them to score how grammatical, relevant, and factual the rationales are on a Likert scale (from 1-5, with 5 being the best), in accordance with past work on generation.

- **Grammaticality** is defined as how well formed, fluent, and grammatical the response is. It achieves a high score due to the sufficient prowess of contemporary LLMs on text generation.
- **Relevance** indicates whether the rationale generated actually answers the prompt query, i.e. the generated rationale aligns well with a human's view of the speaker's intention, assumption, and implicit information about the conversation.
- **Factuality** indicates whether the rationale generated is consistent with the dialogue history; i.e. it does not hallucinate additional information or talk about cases absent in the text.

Overall, we observe an average score of 5.0, 4.6, and 4.8 for grammaticality, relevance, and factuality respectively. We also compute the inter-rater agreement scores (IRA) for these 3 dimensions using the multi-item agreement measure of Lindell et al. (1999) and observe strong agreement scores for all three criteria: grammaticality (0.99), relevance (0.95), and factuality

[1]https://platform.openai.com/docs/models/gpt-3-5

Table 8.4: We present here the manual evaluation scores (ranging from 1 to 5 with 5 being the best) for ChatGPT-generated rationales on the used datasets.

Dataset	Grammaticality	Relevance	Factuality
Friends	5.00	4.55	4.75
IEMOCAP	4.98	4.92	4.34
P4G	5.00	4.52	4.92
CB	5.00	4.55	5.00

	ERC		Res	
	Friends	**IEMOCAP**	**P4G**	**CB**
Dialogues	1000	151	473	713
Total datapoints	14503	10039	11260	8511
Labels	8	8	8	8
Avg. Turns/Dialogue	14.50	66.49	36.05	11.94
Avg. Words/Turn	7.83	11.57	9.22	12.38
Rationales Generated	97.8%	94.78%	97.90%	86.38%
Avg. Words/Intention	32.56	24.47	15.00	14.07
Avg. Words/Assumption	39.06	31.79	17.46	15.10
Avg. Words/Implicit Information	50.04	44.29	19.41	16.55

Table 8.5: We present here the statistics of the datasets used and the rationales generated.

(0.96). Our qualitative analysis reveals that the rationales generated are of high quality and we use them vis-a-vis for our downstream tasks of social meaning detection.

8.1.5 Rationale statistics

Table 8.5 presents statistics of the datasets and the corresponding rationales. Each dialog is broken into multiple datapoints, one for each turn in it. The average number of turns per dialogue and the number of words per turn are reported, with IEMOCAP seen to have significantly longer dialogues compared to the rest. The number of rationales generated for the dataset are reported – For P4G and CB, we encounter parsing issues with GPT-3.5's generated rationales for some instances, which are ignored during training. The average number of words per generated intention/assumption/implicit information is higher for the emotion datasets compared to the resisting strategies ones, which may have been influenced by the choice of the one-shot example in the prompt. The generated implicit information is found to be longer than intention and assumption, and assumption is found to be longer than intention, across all datasets.

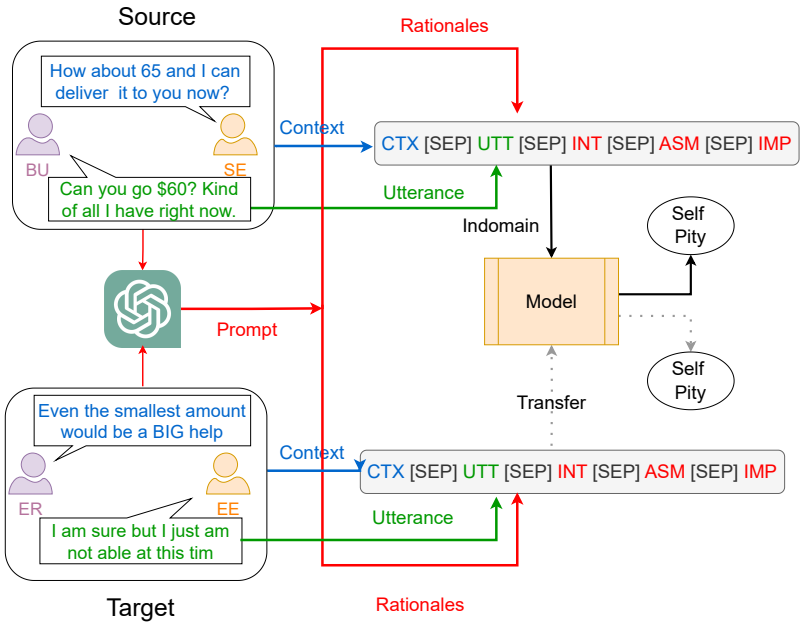


Figure 8.2: Here we illustrate the process of transfer from the source to target. The model is first fine-tuned on the source dialogues, which comprises the current utterance, the previous dialogue context, and the rationales (INT, ASM, and IMP for intentions, assumptions, and implicit information respectively). This fine-tuned model can then be used off-the-shelf for predictions on the target (zero-shot) or further fine-tuned in a few-shot setting.

8.2 Experimental Setup

8.2.1 Settings: In-domain and Transfer

We carry out our experiments in two key settings, namely (i) in-domain (or ID) where the model is evaluated on unseen instances from the same domain or dataset as during training, and (ii) transfer (or TF) where a model that is first finetuned on a domain (say CB) is subsequently used for inference/training on another domain (say P4G).

For both ID and TF scenarios, we simply pass to the model, the concatenated text comprising the past conversational context (whenever applicable), the current utterance, and one or more generated rationales corresponding to the utterance each separated by a [SEP] token. Our baseline is thus simply the text without the generated rationales. For examples, where the generated rationales are invalid, we treat them similar to our baseline.

Additionally, we replicate the experiments for both ID and TF for different N-way, k-shot cases, where k $\in$ 5, 10, 20, 50, and 100. This enables us to diagnose the impact of adding rationales while controlling for data sparsity.

8.2.2 Models and Metrics

We explore both fine-tuning and few-shot prompting, with the latter being used for inference.

Table 8.6: Example of our prompt for the zero-shot and few-shot experiments on LLMs. We illustrate with an example from the P4G dataset.

Part 1: Description of strategy being referenced in the example	These examples pertains to the Source Derogation strategy. For the dataset P4G, the description of Source Derogation is as follows: Attacks/doubts the organisation's credibility.
Part 2: Instructions and output format	Given a response for a particular speaker and recent dialogue context containing the past utterances (wherever available), output 'Yes' if the utterance contains the above strategy, otherwise output 'No'. Your output should contain only 'Yes' or 'No', and no other text.
Part 3: Examples for in-context learning	[CONTEXT] [Persuadee]:Would you like to donate some of your bonus payment to the charity Save the Children today? [Persuadee]:You could choose to donate any amount, from $0 up to $2. [Persuadee]:but i need money because i have finantial problems [Persuadee]:I do understand how that can be. [Persuadee]:But even .10 could go a long way overseas [RESPONSE] [Persuadee]:Thank you for your consideration. [INTENTION] The Persuadee expresses gratitude to the Persuader for considering their request. [ASSUMPTION] The Persuadee assumes that expressing gratitude may positively influence the Persuader's decision. [IMPLICIT INFORMATION] The Persuadee appreciates the Persuader's willingness to consider their request. [OUTPUT] No ...
Part 4: Test datapoint	[CONTEXT] ... [RESPONSE] ... [OUTPUT]

Fine-tuning: We fine-tune three distinct language model families ubiquitous for most NLP applications like Albalak et al. (2022).

(i) Encoder only: We use the base-uncased-version of BERT (Devlin et al., 2019b)

(ii) Decoder only: We employ the base-version of GPT2 (Radford et al., 2019).

(iii) Encoder-Decoder: We utilize the base-version of T5 (Raffel et al., 2020).

Few-shot prompting: We also explore the ability of LLMs, both proprietary and open-source, in a few-shot learning setting. We experiment with GPT-3.5-turbo-16k and the Llama-2-13b-chat-hf (Touvron et al., 2023). We carry out inference in 0-shot and 5-shot setting for LLama-2. We consider only 0-shot for ChatGPT, due to budget restrictions. For 5-shot we randomly sample five positive and five negative instances for a given category from the training split and append them after the task description and instruction. The few-shot prompting framework appears in Table 8.6 in the Appendix.

Metrics: For all settings, we evaluate task performance in terms of the macro-averaged F1 score to account for the uneven distribution of labels for the dataset. We reproduce our experiments across three seeds and report the mean $\pm$ std deviation.

Statistical Analysis: We perform statistical significance using the paired bootstrapped test of Berg-Kirkpatrick et al. (2012) to compare model performance in presence of rationales against the corresponding baseline (absence of any rationale) as stated in Dror et al. (2018).

8.2.3 Hyperparameter Tuning

Hyperparameter	Value
Max sequence length	512
Learning rate	$2e^{-5}$
Batch size	16
Num. epochs	15
Optimizer	Adam

Table 8.7: Hyperparameters used for fine-tuning

We present the hyperparameters for our experiments in Table 11. We carry out the experiments over 3 seeds on a A6000 GPU with early stopping with patience of 5 over the validation set for all experiments. We implement the entire experiments in Python, with help of the Pytorch library and use the pre-trained models as specified in Huggingface under the agreed upon license agreements.

Our experimental suite comprises encompasses 4 datasets in 2 settings (ID/TF) for 3 models (BERT, T5, GPT2) over 5 rationale combinations (none, INT, ASM, IMP, ALL), for 6 few-shot settings (5, 10, 20, 50, 100, and all), and re-evaluated over 3 seeds. This brings the host of experiments to 2160 experiments. There is an additional 180 cases when inferred over 0-shot TF cases, bringing the total to 2340 experiments.

The total cost of the GPT-3.5 credits during the course of our experiments totalled to approx $250 ($200 for generating prompts and $50 for ICL experiments).

8.3 Results

Table 8.8: Performance of the base-variants of models (BERT, GPT2, and T5) on all 4 datasets in an in-domain setting for the entire dataset over three seeds. The rationales (RAT) correspond to intention (INT), assumption (ASM), implicit information (IMP), and the combination of all 3 (ALL) while the absence of any rationale is denoted by -. The best performance for each model category and dataset is denoted in bold, while * signifies the model performs significantly better than the baseline (only the utterance or -).

	CB			P4G			friends			IEMOCAP		
RAT	BERT	GPT2	T5	BERT	GPT2	T5	BERT	GPT2	T5	BERT	GPT2	T5
-	66.7±3.6	60.0±0.9	70.8±1.8	50.6±2.5	35.7±4.4	48.8±0.9	40.9±0.9	26.5±0.8	39.8±3.4	40.7±1.5	35.3±2.4	42.8±1.7
INT	**68.4±1.7**	65.6±2.0*	70.6±2.8	53.0±1.6	45.7±1.6*	51.2±1.4	45.3±0.8*	44.5±1.0*	**44.8±2.6***	**42.6±1.3**	**42.5±2.4***	**45.0±0.7***
ASM	66.6±0.7	65.3±1.3*	69.0±1.8	49.4±8.1	47.7±2.4*	51.1±0.8	44.6±0.1*	43.4±1.2*	39.8±0.6	41.0±1.8	39.3±3.2*	43.1±0.6
IMP	66.9±0.3	64.9±1.6*	69.1±2.6	52.3±1.7	50.1±2.6*	51.7±3.0*	44.7±1.7*	43.3±1.9*	44.1±3.3	42.0±1.2	39.9±0.9*	42.0±0.8
ALL	67.0±0.7	**66.0±1.5***	**72.2±0.5**	**53.2±1.4**	**50.1±1.4***	**53.4±2.7***	**46.2±1.3***	**45.5±0.8***	43.8±3.1	40.4±1.0	39.7±1.8*	44.2±1.2

[RQ1:] What is the impact of rationales on task performance for the in-domain (ID) setting?

We present the results of incorporating rationales on all four datasets for the supervised fine-tuned models in an in-domain setting in Table 8.8. We observe that adding rationales improves model performance across the board over that achieved by the baseline that uses only the utterance. The best F1 score is observed with the combination of all three rationales (ALL) followed by intention (INT).

A more nuanced view reveals that T5 achieves the best task performance followed by BERT and then GPT2. However, we notice a disparate impact of adding rationales on different language model families. GPT2 show significant and consistent improvements across all datasets in presence of any rationale. T5, also benefits largely from rationales where the best ID performance is significant for 3 datasets. In contrast, BERT shows significant performance over the baseline only on the "Friends" dataset. We posit that this could be due to higher quality of rationales generated for the "Friends".

Table 8.9: Task performance in a few-shot prompting setting; 0-shot for GPT-3.5-turbo-16k (GPT-3.5), and both 0-shot and 5-shot for the 13B variant of LLama2-chat model (LLama2-0 and LLama2-5 respectively) . The rationales (RAT) correspond to intention (INT), assumption (ASM), implicit information (IMP), and all 3 (ALL) while the absence of any rationale or the baseline is denoted by -. The best performance for each model is highlighted in bold.

	CB			P4G			Friends			IEMOCAP		
RAT	GPT-3.5	LLama2-0	LLama2-5	GPT-3.5	LLama2-0	LLama2-5	GPT-3.5	LLama2-0	LLama2-5	GPT-3.5	LLama2-0	LLama2-5
-	29.6	18.9	18.7	39.3	1.1	20.3	33.0	18.4	20.2	23.8	16.0	22.4
INT	31.3	14.4	21.5	40.2	1.5	19.1	37.7	**24.3**	24.9	26.5	**25.6**	23.6
ASM	31.2	16.2	21.4	39.6	5.8	19.7	38.8	20.4	23.6	26.2	25.2	22.5
IMP	31.9	18.8	**23.2**	39.7	6.6	**27.7**	39.5	22.2	23.2	26.5	24.5	**24.7**
ALL	**32.4**	**19.2**	19.2	**41.2**	**9.9**	20.9	**39.9**	23.3	**32.5**	**27.0**	24.8	23.1

[RQ2:] How does adding rationales influence few-shot task performance?

We present our results of incorporating rationales on task performance for both in-domain (ID) and transfer (TF) for different k-shot cases in Figure 8.3. We restrict our findings to rationales

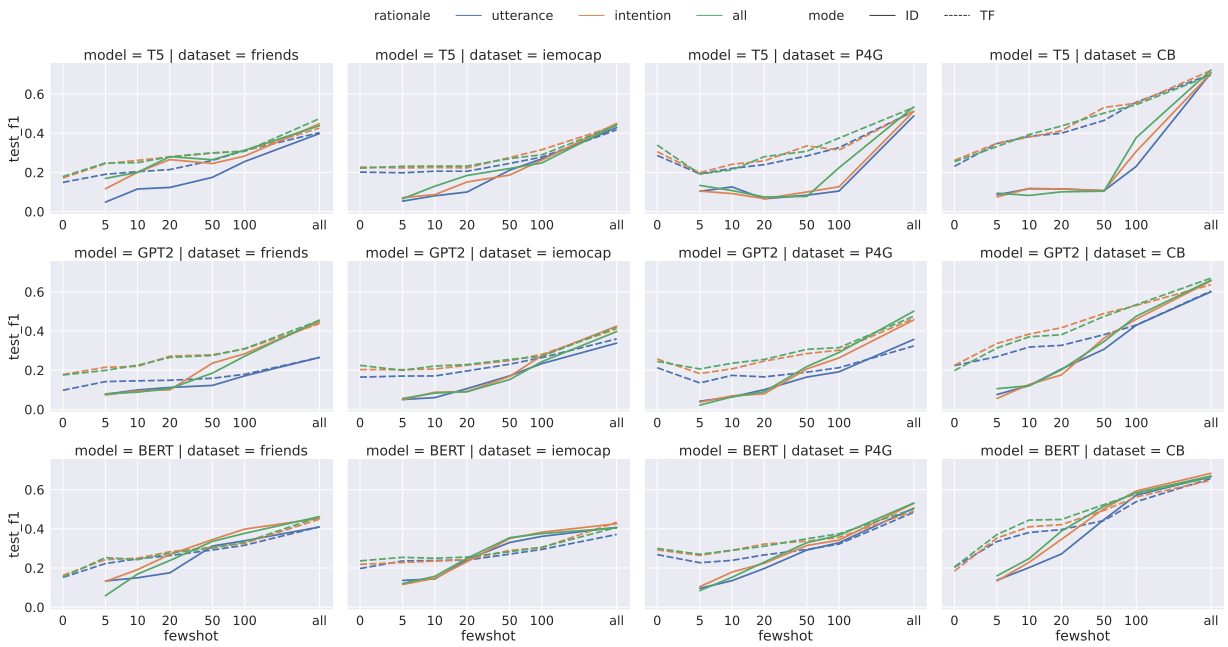


Figure 8.3: Performance of the base-variants of models (BERT, GPT2, and T5) on the four datasets for different few-shot examples. The solid and dashed lines correspond to the indomain (ID) and transfer (TF) case respectively.

corresponding to intention (INT) and combination of all three (ALL) because they had the highest performance in Table 8.8. Our complete set of results are relegated to Figure 1 in the Appendix.

Impact of transfer: One key finding is that the TF performance is consistently higher than in ID (dashed lines score better than the corresponding solid lines) possibly because the model is already trained on the entire source dataset. This is more pronounced in the low data regimes for k-shot corresponding to 5, 10, 20, and 50. and is consistent across all pairs of model and dataset combinations. However, the gain diminishes as the model fine-tuned on the entire dataset (denoted by 'all').

Moreover, adding rationales is better realized for TF than ID; 73.8% of all TF experiments with the rationale ALL had a significantly higher performance over the baseline, while only 1.2% experiments were statistically worse than the baseline. Compare this with 57.0% and 18.1% for ID.

Impact of rationales: Another key finding is the disparate impact of rationales on the task choice. ERC benefits more than RES from adding rationales. For TF, 82.1% and 63.1% of cases that include the rationales are significantly better for ERC and RES respectively; the corresponding proportion in the ID setting is 58.3% and 51.4% respectively. We posit that since the semantic meaning of emotions remains consistent across domains, rationales facilitate transfer better for ERC; or alternately ERC is an easier task than RES.

This observation is echoed vividly in 0-shot transfer where we observe a significant gain 83.3% of the times for ERC as opposed to 41.7% for RES. Nevertheless, in a few-shot setting when the model is exposed to instances from the corresponding target domain, the gains start racking up. We emphasize that across all experiments, rationales perform significantly worse than

the baseline fewer than 10%. Thus, from a big picture view, rationales can indeed facilitate task performance and transfer.

Significant Testing: Considering our massive slew of 2340 experiments, spanning multiple datasets, models, few-shot cases, rationales, and modes (ID/ TF) we also conduct a full-factorial analysis of the experimental suite to obtain a conservative estimate of statistical significance that incorporates the needed adjustments in the face of multiple comparisons in order to avoid type I errors (Gururaja et al., 2023b). For each task, we computed an ANCOVA model with task f1 as the dependent variable, with model (BERT, T5, and GPT2), mode (ID vs TF), rationale (none, INT, ASM, IMP, and ALL) and target domain as independent variables, and few-shot setting nested within mode as a covariate. We also included all 2-way and 3-way interactions between independent variables in the model.

For RES, all independent variables and the covariate were significant, but not the interactions between independent variables. Moreover, performance on CB was consistently higher than P4G, with BERT being the best model. ID was consistently worse than TF. ALL was the best rationale setting, with ASM being the only rationale that was significantly worse than ALL. Including no rationale was significantly worse than all other rationale settings except for ASM.

The story is a little more complicated for the ERC task. We have all the same main effects except dataset – for this task, they are not different from one another. ALL and INT were equally good, and both better than IMP and ASM. All of these were significantly better than including no rationale. There was an interaction between model and these rationales such that the ordering of preferred rationale setting was relatively consistent across different models, but which contrasts were significant varied (note the Tables in the Appendix where different models achieve the best score with different rationales). Nevertheless, including rationales was always better than not including rationales at all, and INT was consistently ranked high. In a nutshell, the rationale INT had the highest impact on model performance.

[RQ3:] How does adding rationales affect few-shot prompting performance for LLMs?

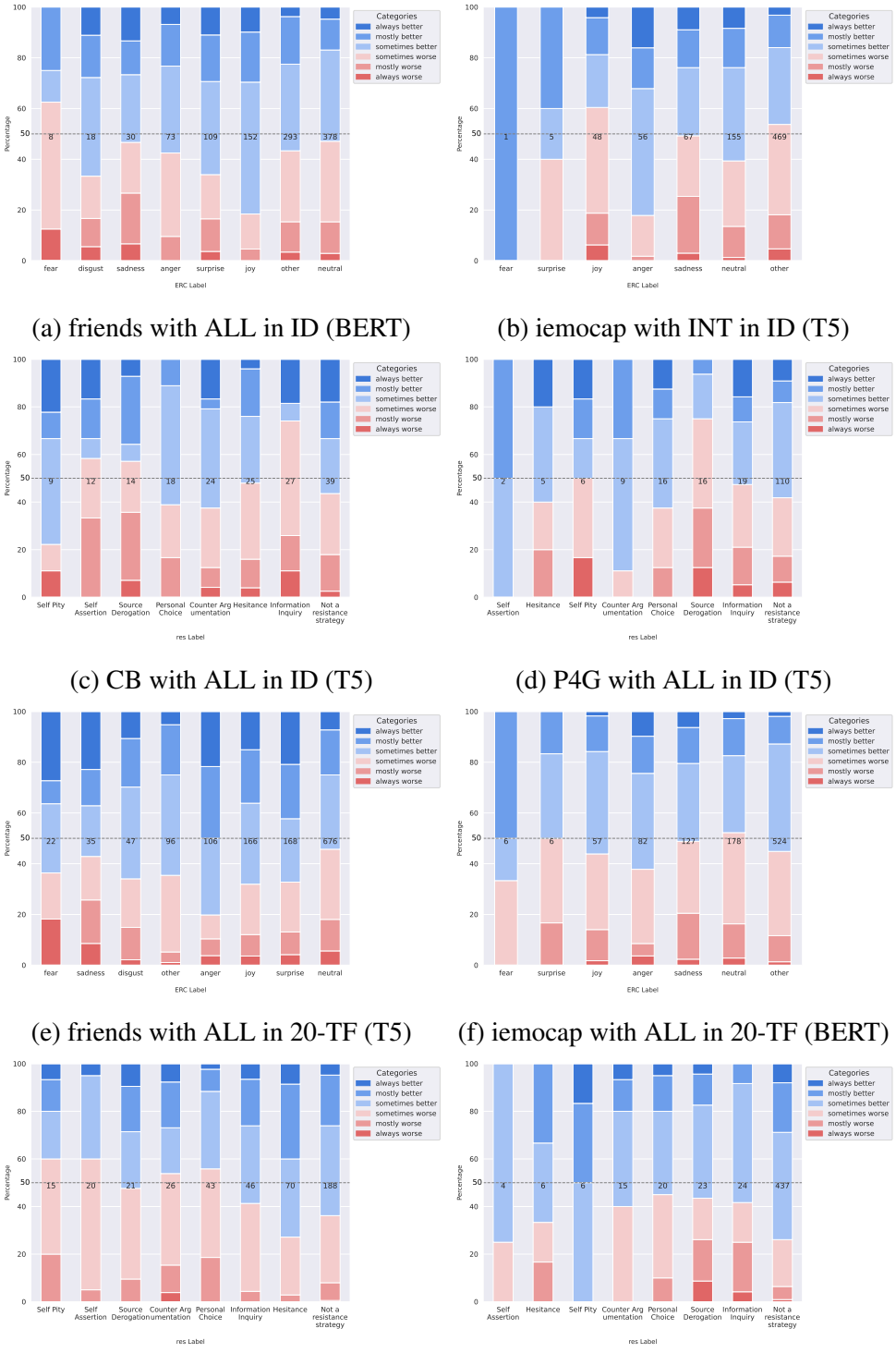
We present our results of using rationales for few-shot prompting in LLMs in Table 8.9. We observe similar trends to the supervised learning set-up wherein the inclusion of rationales improves task performance. Once again, the combination of rationales (ALL) achieves the highest F1 score, while both INT and IMP take a close second. Unsurprisingly, we see the best performance for GPT-3.5 in 0-shot followed by LLama2-13B in a 5-shot setting. Nevertheless, the few-shot prompting results are significantly worse than the fine-tuned supervised models, with results on CB and IEMOCAP being matched by our smaller models at k=5 and k=50 respectively.

[RQ4:] For which cases do rationales show a consistent gain?

Having demonstrated the efficacy of rationales to facilitate understanding of social meaning in dialogue, we do a deep dive on their utility, namely where do rationales help and why.

We investigate the impact rationales have on individual task labels or strategies in ID. For each dataset, we consider the combination of model and rationale pair with the highest ID performance in Table 8.8 and compare their predictions against the baseline (the corresponding model with only UTT). Immediately, we observe that rationales help to shift or re-distribute the prediction probability mass from the majority ("neutral" for ERC and "Not a resistance strategy or NAS" for RES) to others. We highlight examples where adding rationales were consistently better in Table 8.10 and cases where their presence consistently degrades performance in Table 8.11.

Rationales better for ERC: Notably, for ERC, adding rationales is better at identifying the



(a) friends with ALL in ID (BERT)

(b) iemocap with INT in ID (T5)

(c) CB with ALL in ID (T5)

(d) P4G with ALL in ID (T5)

(e) friends with ALL in 20-TF (T5)

(f) iemocap with ALL in 20-TF (BERT)

(g) CB with ALL in 20-TF (BERT)

(h) P4G with INT in 20-TF (BERT)

Figure 8.4: We present here the stacked bar plots that showcases the relative percentage of times a given label was predicted correctly by the best-performing model when augmented with a particular rationale as opposed to the baseline for different datasets. The labels are arranged in increasing order of frequency, with the number inside each bar indicating the frequency of the label.

Table 8.10: Analysis of dialogue utterances with corresponding contextual information and labels when rationales (RAT) are always better.

Dataset	UTT	Context	Model	RAT	GL/RP/UP
friends	Ross: !	Chloe: Do I know why we're rushing? Ross: Yeah, y'know the ah, the girlfriend I told you about last night? Ross: Well it turns out she ah, she wants to get back together with me. Ross: Oh, I found it!	BERT-ALL	The speaker's intention in the final utterance is to express excitement or surprise. The exclamation mark indicates a sudden realization or discovery. Based on the dialogue history, it can be assumed that Ross has found something important or significant. The exclamation suggests that he has found something he was looking for or something that has a positive impact on the situation. The implicit information in the final utterance is that Ross has found something that is relevant to the previous conversation. It implies that this discovery may have an impact on his relationship or the situation he was discussing with Chloe. The content of what Ross has found is not explicitly stated but can be inferred as being significant or positive based on his exclamation.	Surprise / Surprise / Neutral
iemocap	F: Thanks.	F: Thank you M: Of course. M: Is there a funeral planned? or- F: Yeah, they're going to do some sort of memorial service or something. M: Cool, Well, If you want me to go with you, I will.	T5-INT	The speaker intends to express gratitude for the offer of support and condolences given by the person they are speaking to. The speaker assumes that the person they are speaking to genuinely cares about their well-being and is offering to accompany them to the memorial service. The speaker implies that they appreciate the kindness and support shown by the person they are speaking to. The final utterance also suggests that the speaker may be feeling a sense of comfort or relief knowing that they have someone who is willing to be there for them during this difficult time.	Sadness / Sadness / Other
P4G	Persuadee: When finished with this task I will be sure to check the website.	Persuader: Well it's a global non-profit charity based around helping children. Persuadee: What are the donations mostly used for? Persuader: They are used for helping children in a variety of areas: access to education, healthcare, protection from abuse, and relief from natural disasters or conflicts. Persuadee: How are donations solicited? Persuader: Well you can go on their website URL or it can be deducted from this task payment.	T5-ALL	The Persuadee mentions their intention to visit the organization's website after completing the current task. The Persuadee assumes that the Persuader may be interested in supporting the organization and wants to assure them that they will follow up. The Persuadee expresses a commitment to checking the organization's website, indicating a potential interest in making a donation or getting involved.	Hesitance / Hesitance / Not A Resistance Strategy
CB	Buyer: that would be nice. but how about 180 since im the one picking it up and with its one handle missing?	Seller: It is very good. It is actually a family heirloom passed down by my grandmother and she recently passed away 2 days ago. The nightstand is empty however. Buyer: I'm sorry to hear to that. For how much do you wanna give it away? Seller: Well her will listed the value of this item at 200 US dollars, but if you pick it up from her home I am willing to go lower. Buyer: How lower would you go then? Seller: 190$ and I can provide lemonade as you move it out.	T5-ALL	The buyer suggests a further reduction to 180$ due to the missing handle on the nightstand. The buyer assumes that the missing handle affects the value of the nightstand and justifies a lower price. The buyer is negotiating a lower price based on the condition of the nightstand.	Counter Argumentation / Counter Argumentation / Personal Choice

Table 8.11: Analysis of dialogue utterances with corresponding contextual information and labels when rationales are always worse

Dataset	UTT	Context	Model	RAT	GL/RP/UP
friends	Ross: What the hell happened on that beach?!	Ross: Yeah I guess I'll let it go Ross: But you-you understand how-how hard it is to forget about this. Joey: Sure, it's hard to forget! Joey: But that doesn't mean you have to talk about it! Joey: A lot of things happened on that trip that we should never, ever talk about.	BERT-ALL	The speaker's intention in the final utterance is to express frustration and curiosity about the events that transpired on the beach. Ross is seeking answers and wants to know the details of what happened. The assumption here is that something significant or unexpected occurred on the beach, and Ross feels the need to understand and address it. The speaker assumes that there is information that has not been shared or discussed, and Ross wants to uncover the truth. The implicit information in the final utterance is that there was an incident or series of events on the beach that have not been fully disclosed or explained. It suggests that the speaker believes there is a hidden story or secret that needs to be revealed for a complete understanding of the situation. Additionally, it implies that the beach incident is causing tension and unresolved issues among the characters involved.	Surprise / Anger / Surprise
iemocap	F: I'm just worried.	M: If it's rent or something like that, I mean I can help you out a little bit, if- if that will help you you know stick with it. F: I don't know. Maybe I can take out a loan. I just- I need prospects and I don't feel like I have any. M: Well, just don't give up. And there's-you know, You never know, something might be around the corner tomorrow. F: But that doesn't help me when I'm hungry now. M: Well, you know I'm not going to let you starve.	T5-INT	The speaker intends to express their concern and anxiety about their current situation. The speaker assumes that their worries and fears have been discussed and understood by the person they are speaking to. The speaker implies that they have been discussing their difficulties and challenges with the person they are speaking to, and that their worries are related to their current circumstances. The speaker also implies a sense of vulnerability and uncertainty about the future.	Sadness / Other / Sadness
P4G	Persuadee: Perhaps a link to an organization or other agency that rates major charities would be more helpful.	Persuadee: I'm afraid for me, their reputation is still bad. Persuadee: Sorry, no. Persuader: URL Is there website! Persuader: You can check them out. Persuadee: Actually, their own website may be a biased barometer of their giving.	T5-ALL	The Persuadee proposes an alternative approach by suggesting a link to an organization or agency that rates major charities. The Persuadee assumes that relying on an organization or agency that rates major charities would provide a more objective and reliable assessment. The Persuadee values objectivity and reliability when it comes to evaluating the subject's giving and believes that an external organization or agency can provide a more accurate assessment.	Source Derogation / Counter Argumentation / Source Derogation strategy
CB	Buyer: I just want to make sure they work and are quality / not deffective	Seller: Are you interested in the Subwoofer? It's a beauty. Buyer: It looks good, but wondering a few things, how old is it? Seller: I bought it six months ago, but I never actually took it out of the original box. It really has never been used. Buyer: Oh, why is that? Seller: I expected to have more time. I got sent on a 3 month business trip for my work and never got around it.	T5-ALL	The buyer wants to ensure that the Subwoofer is in working condition and of good quality. The buyer assumes that there might be a risk of the Subwoofer being defective or of poor quality. The buyer wants to protect their investment and avoid purchasing a faulty or subpar Subwoofer.	Source Derogation / Information Inquiry / Source Derogation

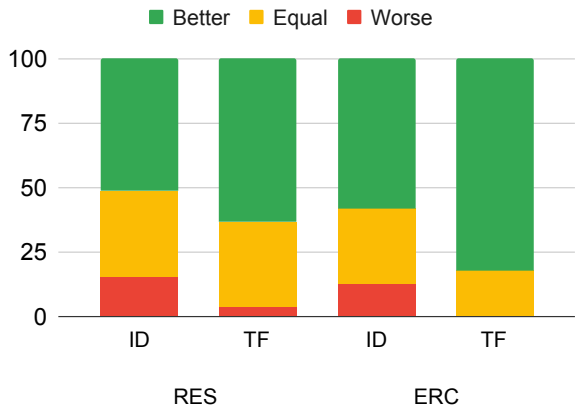


Figure 8.5: Fraction of cases where the classification performance was better, same, or worse, when rationales were augmented, for different tasks, i.e. Resistance strategies (RES) and Emotion Recognition (ERC) and settings i.e. in-domain (ID) and transfer (TF).

emotions "surprise" and "anger". This improved performance can be largely attributed to the fact that the elicited rationales, particularly the intentions (INT), make apparent the emotional state. For instance, the INT rationale interprets the exclamation mark "!" in the utterance for the Friends dataset as an expression of excitement or surprise, and thus corresponds with the actual label (surprise). Likewise, for the utterance "Thanks" from IEMOCAP is characterized in the rationales as reflecting gratitude or acknowledgment of support and condolences, contributing to an overall sentiment of "sadness" in response to a bereavement consolation.

Rationales worse for ERC: The cases where the model mispredicts can be linked to the specific language usage. For example, the utterance in friends "What the hell happened on that beach?!" is erroneously interpreted as anger possibly due to "what the hell." Likewise, for the utterance "I'm just worried," in IEMOCAP, the rationales express a sense of anxiety or uncertainty from "worried" misleading the prediction as "other" than "sadness."

Rationales better for RES: For RES, the integration of rationales notably enhances performance for "Counter Argumentation" and "Hesitance." E.g., in the CB dataset, for the utterance "but how about 180 since I'm the one picking it up and with its one handle missing?", the rationale accurately identifies the buyer's intention to propose a reduced price due to the item's missing handle, and thus aligns with Counter Argumentation. Furthermore, for P4G, "when finished with this task I will be sure to check the website," the rationales portray the speaker's implied conditional interest, indicating Hesitance as the action is deferred until task completion.

Rationales worse for RES: Conversely, the model's performance for the "Source Derogation" strategy is less effective. A typical example is "perhaps a link to an organization or other agency that rates major charities would be more helpful" for P4G. Here, the rationales inaccurately interpret the statement as a mere suggestion for a more efficient information source, and fail to detect the speaker's skepticism about the organization's credibility. We posit that this misprediction is linked to LLM's tendency to generate responses with a positive connotation, leading to a misinterpretation of critical tones as constructive suggestions. This results in erroneous labeling as "Information Inquiry" indicating a request for additional information, or "Counter Argumentation,"

which suggests an alternative factual proposition.

While we note that overall rationales facilitate transfer, the gains observed are not symmetric. Specifically, we observe higher gains for the less frequent classes in the target dataset, such as the emotion "fear" on Friends and "Source Derogation" and "Self Pity" classes on the P4G dataset.

8.4 Conclusion and Takeaways

This chapter serves to showcase how machine-generated rationales can translate implicit pragmatic information into explicit text thereby facilitating generalization. Empirically, we show that rationale augmentation improves performance across both in-domain and (more strongly) transfer settings for RES and ERC, 8.5 with the largest gains appearing in low-data regimes where the inductive bias from scaffolds matters the most. So far we have examined rationales from a specific user-centric angle, i.e. the intentions and assumptions of the speaker. We broaden the scope of our study by considering rationales that capture multi-faceted perspectives. Likewise, we also move towards a more challenging setting, i.e. cross-task generalization as opposed to cross-domain generalization. We attempt to address both of these challenges in Chapter 9.

Chapter 9

SOCIAL SCAFFOLDS: A Generalization Framework for Social Understanding Tasks

Building upon the past chapters, we investigate the capabilities of task-agnostic rationales to facilitate generalization across different social meaning understanding tasks. We provide a pictorial illustration of the same in Figure 9.1. We introduce an automated framework SOCIAL SCAFFOLDS to generate task-agnostic social rationales that are capable of capturing perspectives corresponding to different points of view in narrative modeling. The rest of the chapter discusses the corresponding modeling framework, our experimental setup, and finally our results and analyses.

9.1 Modeling Framework

We present SOCIAL SCAFFOLDS, an automated framework that facilitates task generalization by generating different social rationales to capture the implicit information behind a message.

9.1.1 Rationale Types

We explore three distinct but complementary perspectives to generate the rationales. Motivated by prior work on narrative modeling, we present a one-to-one correspondence of the rationale category with the narrative perspective or point-of-view.

Intentions: Intentions (or INT) refer to the speaker's hidden beliefs and desires, and correspond to the *first-person perspective*. They capture the implied meaning behind the speaker's utterance or signal the outcome the speaker wants (Dutt et al., 2024; Yusupujiang and Ginzburg, 2023).

Hearer Reaction: Hearer reactions (or HR) (Zhou et al., 2023b; Sap et al., 2020b) capture the effect the utterance might have on the listener(s). They provide insight into the listener's emotions or belief states, akin to second-order thinking, and correspond to the *second-person perspective.*

Presuppositions: We use presuppositions (hereafter PreSup) to refer to general facts or truths that participants believe for the utterance to be credible. PreSup not only encapsulates common sense reasoning or social and communal norms often observed in practice (Perez Gomez, 2021;

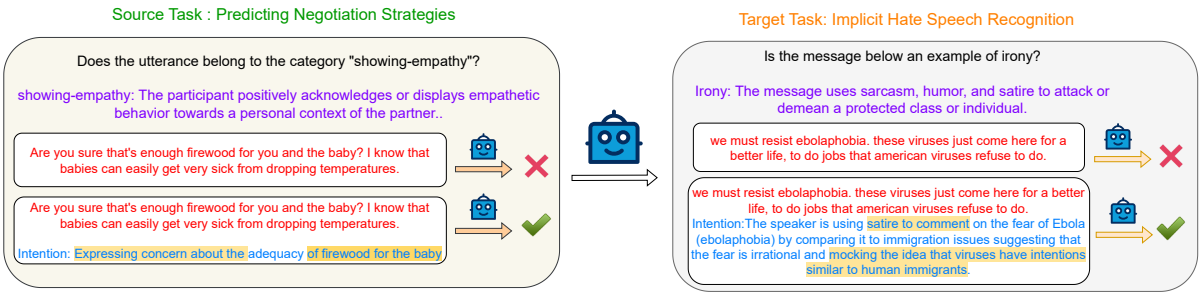


Figure 9.1: We illustrate the phenomena of indirect or subtle language usage in two scenarios; the scenario on the left corresponds to predicting negotiation strategies, whereas the scenario on the right corresponds to identifying different categories of hate. For both cases, we observe that the model fails to associate the input message (in red) with the label description (in purple) due to its inability to capture the hidden cues in the message. Incorporating rationales, as additional inputs, can guide model prediction for both in-domain and cross-task settings.

Kim et al., 2022), but also provides a de-contextualized and impersonal insight and thus serves as a *third-person perspective* (Mulcahy and Gouldthorp, 2016).

9.1.2 Rationale Generation Framework

We describe our prompting setup to automatically generate the different categories of rationales. Figure 9.2 presents a sample negotiation snippet with the corresponding intention, hearer reaction, and presupposition for the seller's last utterance.

SOCIAL SCAFFOLDS takes as input a multiparty dialog and generates rationales using a Large Language Model (such as GPT-4o) on an utterance-by-utterance basis. We employ a structured prompting framework to ensure that the generated rationale aligns with its corresponding utterance. Specifically, we generate rationales in the form of a CSV file and align them corresponding speaker and utterance index. These checks and measures help ensure that each utterance has a corresponding rationale and enables us to revisit erroneous cases. We address these cases by prompting the framework to regenerate the rationales iteratively. We stop after 3 iterations.

We generate each rationale category (e.g, intentions or presuppositions) using our framework separately to prevent any ordering effects. We do not provide few-shot instances to avoid biasing the generations with previously seen examples, unlike Dutt et al. (2024). Such a setting enables us to compare and contrast (i) different categories of rationales and (ii) rationales of the same category but generated by different LLMs. We explore both proprietary models, such as GPT-4o and GPT-3.5-turbo, and open-weight LLMs, such as Gemma-2-27B-it, as the backbone of SOCIAL SCAFFOLDS. We showcase the exact prompt to generate the rationales in Figure 9.3.

9.1.3 Assessment of Rationale Quality

Metrics for Annotating Rationales

Since our framework automatically generates rationales without any human supervision, we develop a rigorous annotation manual to assess the validity of those generations based on three

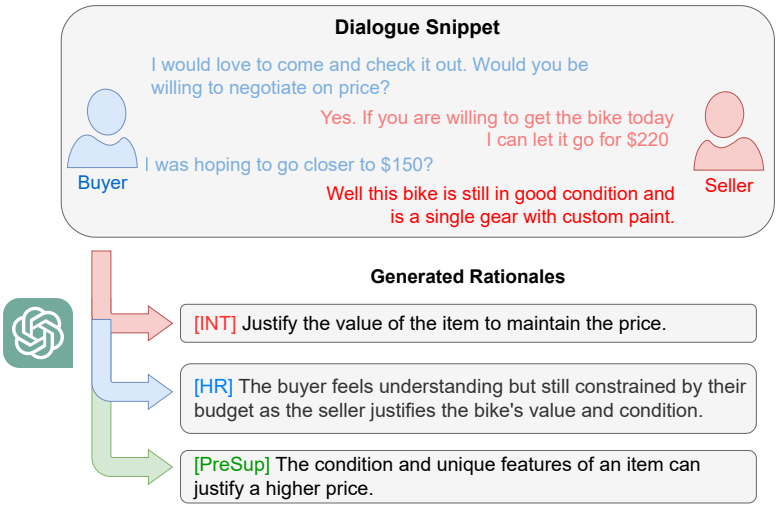


Figure 9.2: An overview of SOCIAL SCAFFOLDS for a negotiation snippet between a buyer and a seller. We prompt an LLM to generate rationales corresponding to the speaker's intentions (INT), the hearer's reaction (HR), and the presuppositions (PreSup) for a given dialogue. For brevity, we show only the rationales corresponding to the seller's last utterance.

criteria: soundness, informativeness, and relevance.

- **Soundness:** Soundness reflects whether the rationale adheres to the definition provided during prompting, i.e. whether the generated rationale reflects the speaker's intentions, the hearer's reactions, and the presuppositions about the world. In some cases, the rationale generated might not contain any additional subtext beyond the literal rephrasing of the utterance. Such instances are scored high on soundness.
- **Informativeness:** The information conveyed by the rationales should comply with the context of the current dialogue. The information should be correct, i.e. rationale should not exhibit hallucination, (present additional information that has not been encountered so far in the dialogue), and complete, i.e. they should not omit important information that could change the meaning of the utterance.
- **Relevance:** A rationale is relevant when it goes beyond the utterance text and presents information that is not only factual and sound but also provides additional subtext. We include this metric to assess whether the rationale is useful or not for the current scenario by providing important information or cues that are not directly observable.

We score each rationale for each criterion using a Likert scale of 1 to 3, with one being the lowest and three the highest. Our two annotators or evaluators had a graduate-level proficiency in English and at least five years of experience in computational linguistics and NLP. Due to the highly subjective nature of the task, we relied on these professional annotators as an alternative to crowd-sourcing or employing an automated annotation framework.

Prompt for generating rationales

System message: ”You are a helpful agent that produces consistent and structured output.”
User message: ”Analyse the dialogue below enclosed within the ¡dialog¿ and ¡/dialog¿ tags and identify the speaker’s intentions for each utterance. Go over the entire conversation on an utterance-by-utterance basis without grouping or skipping, and generate the corresponding speakers’ intentions for each utterance iteratively. Return the result in the format of an csv file, with the headers corresponding to the following columns: i.e. utterance_idx, speaker, and intentions.”

Figure 9.3: The prompt we pass to our framework to generate the rationales of a corresponding category.

Annotation Guidelines

We present the flowchart for annotating rationales according to soundness, informativeness, and relevance.
Step 1: Read the dialogue history, utterance and the rationale; start with judging the Speaker Intention rationale. Perform Steps 2-4 for the Speaker Intention rationale and then reiterate for Hearer Reaction and Presuppositions.
Step 2: Check for Soundness criteria if the generated rationale encapsulates the meaning of the rationale category. When checking for Speaker Intention rationales, see if it is about the speaker’s beliefs, goals, objectives, outcomes. When checking for Hearer Reaction see if it is about the belief of the hearer or their interpretation. When checking for Presuppositions see if it reflects the general world view or the assumptions shared by the participants.

- If the rationale is ascribing the correct perspective, we assign a 3 to Soundness.
- If the perspective appears to be ambiguous, we assign 2 for Soundness.
- If the perspective is blatantly incorrect, for example the Hearer Reaction actually reflects the speaker’s intentions we assign 1 to Soundness.
- If Soundness is 1 all criteria should be assigned 1, since it does not make sense to evaluate a wrong rationale.

Step 3: We now check whether the rationale is Informative or not, i.e. whether the information present in the rationale is accurate.

- If all the details have been carried over from the utterance, with an appropriate level of generalization assign a 3 to Informativeness.
- If the generalization has omitted some information/details that are important to the meaning of the utterance, assign a 2 for Informativeness.
- If the rationale hallucinates information, i.e. presents information that cannot be inferred from the current dialogue context, or is otherwise just wrong, assign a 1 for Informativeness.

Note that Informativeness and Relevance are always 1 when the Soundness is 1.

Step 4: We finally check for Relevance.

- If the utterance has a subtext and the rationale has identified a subtext not overtly stated in the utterance text, assign a 3 for Relevance.
- If the rationale includes information that appears earlier in the dialogue history whether it is subtext or not, but is not in the particular utterance, assign a 3 for Relevance.
- If the utterance lacks subtext, but the rationale presents an expression or action not found in the utterance, such as expressing agreement or an opinion, assign a 3 for Relevance.
- If the utterance lacks subtext and the rationale simply summarizes the details of the given utterance without adding anything new at all, assign a 2 for Relevance.
- If the utterance has an underlying subtext but that is not captured by the rationale, or an incorrect subtext is present, assign a 1 for Relevance.

Mean Likert Scores			
Metric	INT	HR	PreSup
Soundness	3.00	2.85	3.00
Informativeness	2.62	2.72	2.93
Relevance	2.43	2.67	2.72
IRR Score			
Metric	INT	HR	PreSup
Soundness	1.00	0.95	1.00
Informativeness	0.70	0.80	0.82
Relevance	0.78	0.86	0.51

Table 9.1: Annotation results for the different types of rationales based on different criterion.

Scoring the Generated Rationales

We compute inter-rater reliability scores (IRR) using the multi-item agreement measure of Lindell et al. (1999) following prior work of Dutt et al. (2024) and observe moderate to strong agreement scores for all three criteria: soundness (0.983), informativeness (0.763), and relevance (0.697). We present a detailed breakdown of the mean Likert scores and the corresponding measure of IRR agreement for the three different categories of rationales in different dimensions in Table 9.1. We observe that the intention rationale has the lowest score on both informativeness and relevance. However, the rationale that exhibits the highest disagreement is the presuppositions on the relevance metric.

We inspect the disagreement cases between annotators and present some instances in Table 9.2. We showcase examples of disagreement for both informativeness and relevance. Since the IRR agreement were the lowest for (i) INTs on the informativeness metric and (ii) the presuppositions on the relevance metric, we have more instances of those categories in our Table.

Table 9.2: Instances of annotator disagreement for the different datasets

Dataset	Utterance	Rationale	Category	A	B
		Relevance Disagreements			
IMP_HATE	#dayafterchristmas in america during the current year . join the #altright white people . #malls	There is a significant cultural or social event related to the day after Christmas in America that is relevant to the alt-right and white people and it involves malls.	PreSup	2	3
PROP	CLINTON would ask them then draw their own conclusion	Clinton believes voters should make an informed decision based on her record.	PreSup	2	3
casino	I completely understand where you are coming from. How much food do you need?	mturk_agent_1 feels understood and is prompted to specify needs.	HR	2	3
P4G	Almost 87% of all dollars donated go directly the programs supporting the children, so they have a very low overhead.	A high percentage of donations go directly to the programs indicating low administrative costs.	PreSup	3	2
		Informativeness Disagreements			
res_CB	Look it up, these are very high end bikes	Encouraging the buyer to verify the information	INT	3	2
EMH	Be strong my friend, Everytime I come in here, I don't feel so alone. This may sound selfish, but it's good for me to know that there's people out there that feels the pain that I feel every single day. Anyway, have a great day, my friends.	Offering encouragement and sharing personal experience to provide comfort	INT	3	2

We observe that annotator B was more critical of the annotation framework; they honed in on specific cues surface cues to illustrate why the rationale is relevant. For example, the phrase "significant cultural and social event" provides an additional subtext in the first instance. Likewise, terms such as "Clinton believes" or "feels understood" expresses additional emotions that were absent from the utterance. On the other hand, annotator A judged the rationale as relevant if it introduced new information. They were also a bit more relaxed in critiquing the informativeness score, rating the rationale to be highly informative if it was able to capture the essence of the utterance. However, annotator B rated the two intentions in Table 9.2 with a score of two, because the rationales had omitted specific information such as the "high-end price of the bike" in the former case and because the term "personal experience" was an overgeneralization of the responder's experience for the latter.

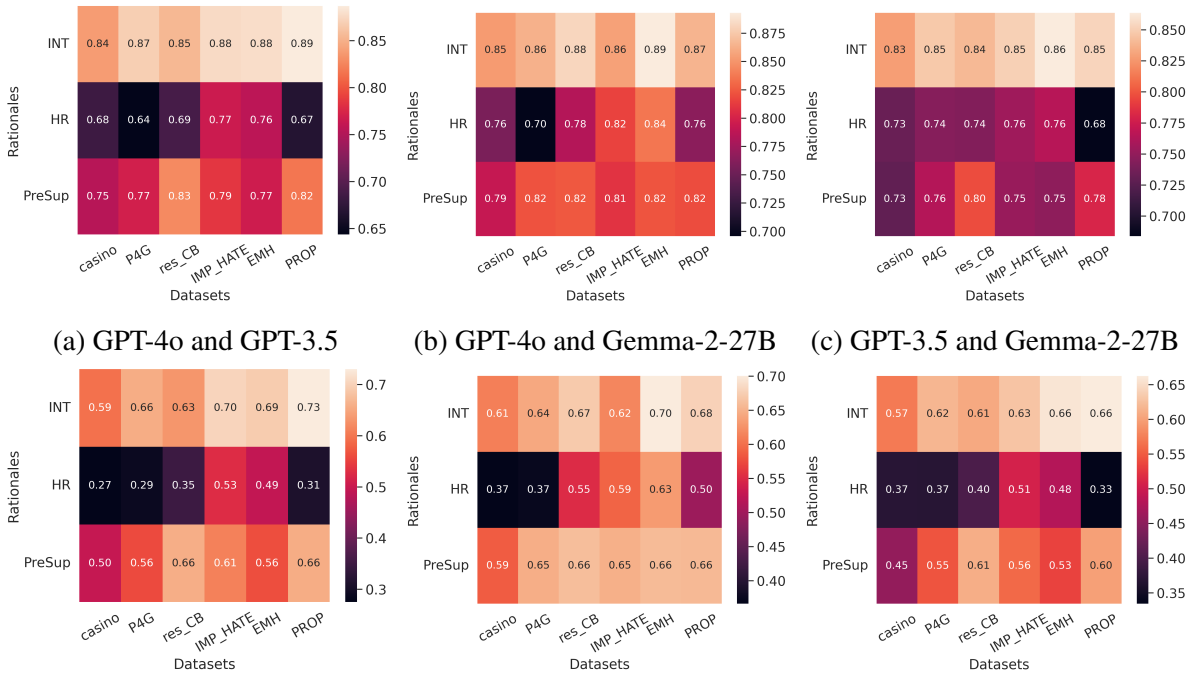
.

9.1.4 Characteristics of the generated rationales

We measure the similarity of the generated rationales across three fronts:

- How similar are the three different categories of rationales to each other?
- How similar are the rationales generated by different LLMs for the same rationale category?
- How similar is a generated rationale to its corresponding utterance?

We use cosine distance between the sentential representations as the metric for quantifying similarity. We explore two models to generate these representations, i.e., the popular MPNET



(a) GPT-4o and GPT-3.5 (b) GPT-4o and Gemma-2-27B (c) GPT-3.5 and Gemma-2-27B

(d) GPT-4o and GPT-3.5 (e) GPT-4o and Gemma-2-27B (f) GPT-3.5 and Gemma-2-27B

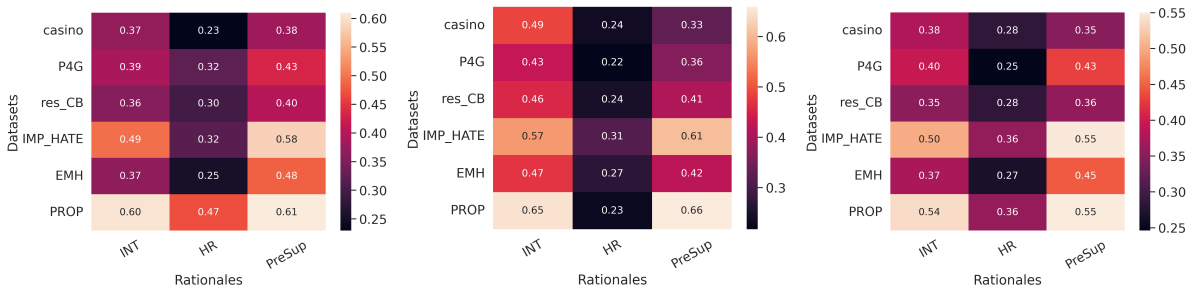
Figure 9.4: Cosine similarities between rationales generated by three LLMs, i.e. GPT-4o, GPT-3.5-turbo (GPT-3.5) and Gemma-2-27B-it (Gemma-2-27B), across different datasets and rationale categories. The figures displayed on the left and right correspond to the models Mistral and MPNET, respectively.

model of (Reimers and Gurevych, 2019) for its simplicity and the instruction-tuned version of Mistral-7B (Wang et al., 2023a) for its superior performance on the MTEB leaderboard (Muennighoff et al., 2023). We present the similarity scores across different LLMs, different rationale categories, and between the utterance and the rationale in Figures 10, 9.6, and 9.5 respectively.

We observe similar trends in the scores regardless of the model used to generate the representations, i.e., MPNET and Mistral. The rationales generated by GPT-4o and GPT-3.5-turbo vary considerably in their similarity scores depending on their category; those corresponding to the speaker's intentions (INT) are the most similar, followed by presuppositions (PreSup), while the hearer reactions (HR) are highly dissimilar. Furthermore, we note a low similarity between rationales corresponding to different categories (the weakest scores occur between PreSup and HR) and between the rationale and the original utterance. Overall, these results highlight that the categories capture perspectives distinct from each other and the original utterance.

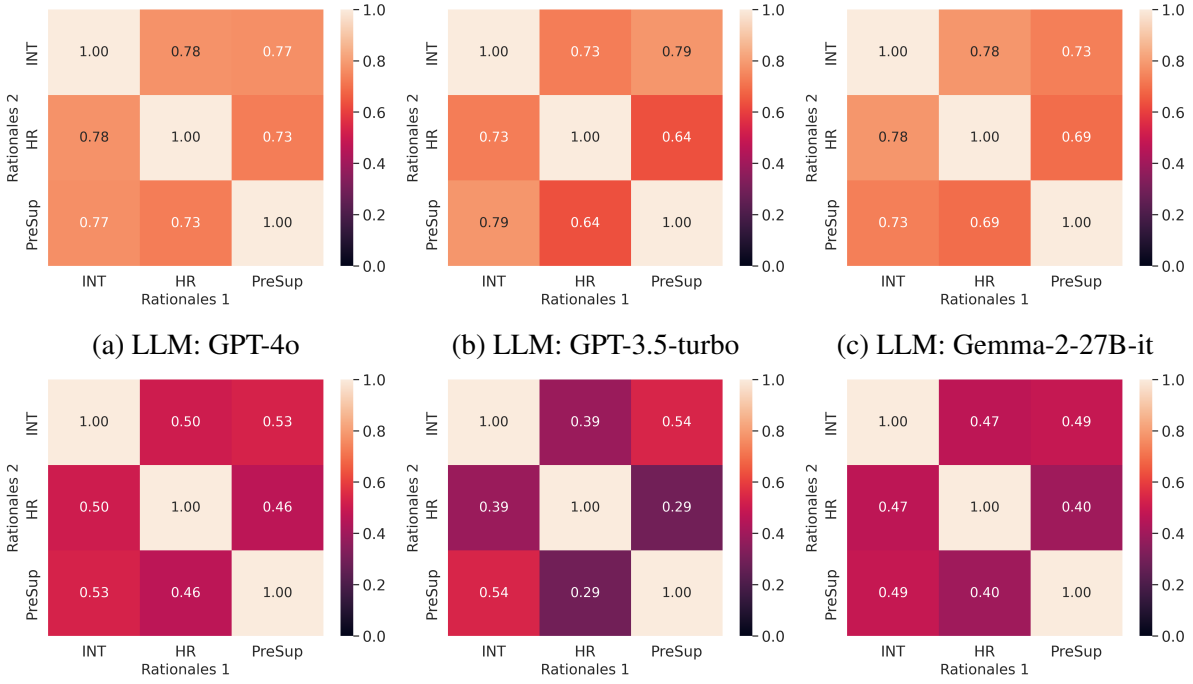
9.2 Experimental Setup

We outline the details of our methods or experimental setup for investigating the role of rationales in aiding generalization for understanding tasks. We describe the tasks, models, settings, and



(a) GPT-4o (b) GPT-3.5-turbo (c) Gemma-2-27B-it

Figure 9.5: Cosine similarities between the original utterance and the rationales generated by different LLMs and evaluated by the sentence transformers MPNET.



(a) LLM: GPT-4o (b) LLM: GPT-3.5-turbo (c) LLM: Gemma-2-27B-it

(d) LLM: GPT-4o (e) LLM: GPT-3.5-turbo (f) LLM: Gemma-2-27B-it

Figure 9.6: Cosine similarities between different categories of rationales corresponding to intentions, hearer reactions, and presuppositions as generated by three LLMs, GPT-4o and GPT-3.5-turbo, and Gemma-2-27B-it, and evaluated by the sentence transformers, i.e. Mistral (top 3) and MPNET (bottom 3).

metrics.

9.2.1 Tasks and Datasets

Our datasets include (i) P4G (Wang et al., 2019a) to identify persuasive strategies in charitable donations, (ii) CaSiNo (Chawla et al., 2021) to detect negotiation tactics during camping, (iii) Res_CB (Dutt et al., 2021) to categorize strategies employed to resist persuasion in online bargaining, (iv) EMH (Sharma et al., 2020) to understand different dimensions of empathy, (v) PROP (Jo et al., 2020) to categorize different kinds of argumentation, and (vi) IMP_HATE (ElSherief et al., 2021) to classify different kinds of implicit hate speech. We discussed these in detail in Chapter 7.

9.2.2 Configurations: SFT and ICL

We test the impact of rationales on downstream task performance in two distinct configurations. The first is a supervised fine-tuning (SFT) setup (Figure 9.7); we instruct-tune a pre-trained language model on a given source task (say persuasion) and then subsequently evaluate it on a new target task (say argumentation) in a 0-shot or few-shot setting. We also explore parameter efficient fine-tuning of instruct-tuned LLMs as part of SFT. [1] The second setup is in-context learning (ICL), where we prompt an LLM with 0-shot or few-shot examples with the rationale as a control condition.

Since we investigate task transferability, it is imperative for us to map tasks with distinct label categories into a common shared space. **We format each task as binary classification**, such that the model outputs "Yes" or "No", depending on whether the utterance complies with the label definition. The input to the model is the label definition, the utterance, the dialog context, and the corresponding rationale. We adopt the binary classification framework for both SFT and ICL settings. Such a design would allow for a fair comparison of the two paradigms. Moreover, fine-tuned LMs with a single multiclass classification head is unlikely to generalize in a 0-shot setting. We show an example of how these tasks have been setup in Figure 9.1.

9.2.3 Models and Metrics

For the standard SFT setup, we employ the base version of Flan-T5 (Chung et al., 2022) as our primary instruction-tuned model. We also explore parameter efficient fine tuning (PEFT) a pre-trained LLama-3-8B-it model (AI@Meta, 2024) with 4-bit double quantization and low-rank adapter (LoRA) (Hu et al., 2021b; Dettmers et al., 2024). Finally, Gemma-2-9B-it (Team, 2024) and LLama-3-8B-it (AI@Meta, 2024) serve as our main models for ICL. All these models have been trained to follow instructions and thus serve as strong baselines for the respective experimental paradigms. We measure the performance change from adding rationales (i.e., INT, HR, and PreSup) as part of the input text over only the utterance (i.e. the baseline).

Due to the skewed label distribution, we use the macro-F1 score as our evaluation metric for each of these six tasks. Following the recommendations in Dror et al. (2018), we employ the

[1] Additional details of our experiments are in Appendix 2.3

Dataset	Dialog History	Speaker: UTT	INT	PreSup	HR
CaSiNo (Chawla et al., 2021)	**mturk_agent_1:** I am running low on firewood. I need more to keep the fire going and cook food.	**mturk_agent_2:** Yeah I need firewood too. It's only 50 degrees at night on the mountain, even in the summer.	Agreeing on the need for firewood and providing additional context	It gets cold at night on the mountain even in the summer.	mturk_agent_1 feels validated in their need for firewood and understands the shared predicament.
P4G (Wang et al., 2019a)	**ER:** Save the Children is an amazing charity that helps kids who are in desperate need. **ER:** They can help with safety, education and more. **ER:** You can donate some of your earnings to this amazing charity. **EE:** I believe in this charity, but still wonder how much of the money I donate actually helps. **ER:** Every little bit makes a difference.	**ER:** When you have people who are so poor, it's amazing what a tiny amount can do.	Emphasizing the value of small contributions	Even minimal financial aid can greatly benefit those in extreme poverty.	EE is encouraged by the impact of small donations.
res_CB (Dutt et al., 2021)	**Buyer:** Hi there. I was looking for ads and this one caught my attention. Is it in a good and working condition? **Seller:** It is, it's been used a lot less than its age would suggest. I only rode it a few times a month. Are you interested?	**Buyer:** Yes very much so.	Express strong interest in the item	The buyer is expressing a strong interest in the item indicating a willingness to negotiate further.	The seller feels encouraged by the buyer's interest and is prompted to discuss the price.

Table 9.3: Examples of rationales generated by GPT-4o for six utterances, each coming from a different dataset and task. For each utterance, we provide the dialog history and the corresponding intention, presupposition, and hearer reaction abbreviated as INT, PreSup, and HR respectively. The rationales score high on factuality, soundness, and relevance as evaluated by two annotators.

Dataset	Dialog History	Speaker: UTT	INT	PreSup	HR
IMP_HATE (ElSherief et al., 2021)		**Poster:** flynn's resignation set a dangerous precedent for the administration . #sessions is essential to justice 4 white america . he must not #resign	Expressing concern about the implications of Flynn's resignation and emphasizing the importance of Sessions to their view of justice for white America while urging that Sessions should not resign.	The resignation of a high-ranking official can have significant and potentially negative consequences for the administration.	The hearer may feel concerned about the implications of Flynn's resignation and the potential impact on the administration. They may also feel a sense of urgency or importance regarding Sessions' role and the need for him to remain in his position.
EMH (Sharma et al., 2020)	**Seeker:** Why do I always have good news followed by a shit night, followed by sitting up at 2am wanting to kill myself? Why is life so difficult? Why is it so impossible to be fucking happy for once in my shit fucking life? What's the point anymore?	**Responder:** well not for nothing but you made it extremely difficult to read your post by only using a period in the title. JUST saying not judging.	Pointing out the difficulty in reading the post due to formatting while attempting to clarify that they are not judging.	Clear communication is important for understanding and responding to others' concerns effectively.	The Seeker may feel invalidated or criticized as the Responder's comment focuses on the format of the post rather than addressing the Seeker's emotional distress.
PROP (Jo et al., 2020)	**S_1:** It is called the Constitution of the United States **S_2:** unfortunately, those few months gave us OBAMA **S_3:** We're going to win when we unite people with a hopeful, optimistic message **S_3:** we had high sustained economic growth	**S_3:** We created 1.3 million jobs	Emphasizing job creation	Creating jobs is a positive achievement.	Impression of job creation success

Table 9.4: Examples of rationales generated by GPT-4o for six utterances, each coming from a different dataset and task. For each utterance, we provide the dialog history and the corresponding intention, presupposition, and hearer reaction abbreviated as INT, PreSup, and HR respectively. The rationales score high on factuality, soundness, and relevance as evaluated by two annotators.

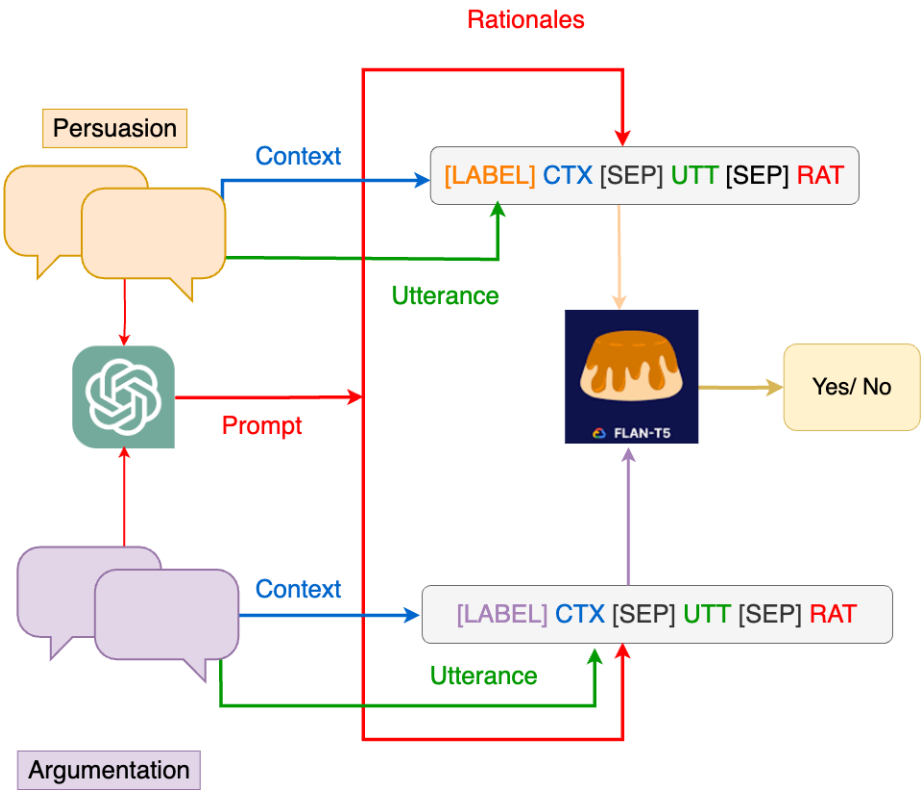


Figure 9.7: Overview of our SFT setting. For a source task, we instruction-tune FLAN-T5 with the label definition, dialogue context, utterance, and rationale as input and predict "yes" or "no" for the corresponding label. This model is then deployed for a new target task.

nonparametric bootstrap test of Berg-Kirkpatrick et al. (2012) to measure whether the rationale-augmented model's performance was statistically significant from the baseline. We reject the null hypothesis for cases with p-value ≤ 0.05. We perform each experiment for three seeds to account for variations over runs.

9.3 Results & Analysis

We present our experimental results with the rationales generated by the most advanced LLM in our study, i.e. GPT-4o. Appendix 2.3 shows similar trends with rationales generated by other LLMs.

9.3.1 Impact of Rationales in an SFT Setup

We inspect the impact of adding rationales on task performance in a supervised fine-tuning setup for both in-domain and cross-task transfer. The in-domain results serve to validate prior work that social rationales can enhance task performance whereas the transfer results showcase whether these rationales can facilitate task generalization.

Rationale	P4G	CaSiNo	res_CB	PROP	EMH	IMP_HATE
UTT	69.70 +/- 2.42	71.22 +/- 1.70	66.77 +/- 1.02	82.38 +/- 1.21	**90.91 +/- 0.13**	62.68 +/- 0.79
+ INT	69.36 +/- 1.45	**72.35 +/- 0.50**	**70.91 +/- 0.71**	84.66 +/- 1.07	89.35 +/- 1.35	67.91 +/- 1.49
+ HR	70.54 +/- 1.70	71.71 +/- 0.84	68.80 +/- 0.97	82.88 +/- 1.69	90.26 +/- 0.32	65.08 +/- 0.34
+ PreSup	68.12 +/- 2.30	71.81 +/- 1.39	69.69 +/- 1.51	80.11 +/- 2.86	89.37 +/- 0.16	62.88 +/- 2.55
+ ALL	**70.67 +/- 2.08**	70.68 +/- 1.12	67.72 +/- 2.59	**86.25 +/- 3.28**	90.46 +/- 1.12	**68.21 +/- 0.97**

Table 9.5: Performance of FLAN-T5 model in an in-domain setting with GPT-4o rationales across six tasks. The baseline includes only the utterance (UTT) which we compare by adding rationales, i.e. intentions (INT), hearer-reactions (HR), presuppositions (PreSup), and all three (ALL). We note the mean and s.d. across three runs.

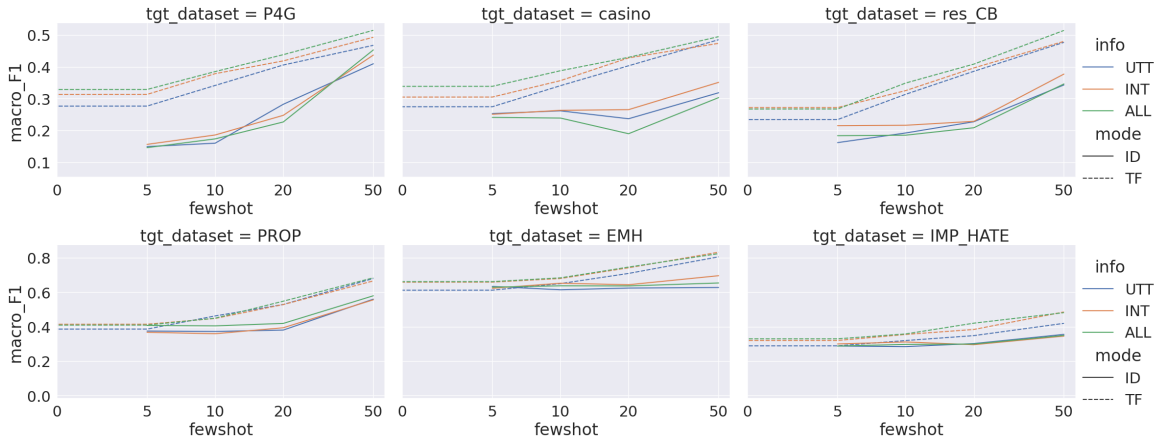


Figure 9.8: Impact of GPT-4o rationales on cross-task performance for different tasks and fewshot settings. TF and ID corresponds to the cross-task transfer and in-domain setting respectively. For better readability, we show results for only the intentions (INT) and all three categories (ALL).

In-domain Results: Table 9.5 shows that rationales improve in-domain performance on five of six tasks with significant gains for res_CB, PROP, and IMP_HATE, and a significant drop for EMH. The rationale with the greatest impact on performance varies across tasks (e.g. intentions are helpful for CaSiNo and res_CB, while the hearers' reaction aids P4G), implying that no individual category acts as a silver bullet. Nevertheless, **adding all three rationale categories (ALL) has the most in-domain benefit**, followed by intentions. Appendix 2.3 shows that our chosen FLAN-T5 model exhibits competitive in-domain task performance and surpasses prior baselines for all tasks.

Cross-Task Transfer Results: Our transfer experiments over the six tasks yield 30 unique source-target pairs. Figure 9.8 shows the aggregate impact of adding rationales for the six target datasets.[2] Against the utterance-only baseline, we see consistent and significant gains during transfer (in dotted lines) over the in-domain setting (in solid lines) for different zero-shot and few-shot cases. A similar trend is seen for PEFT models, albeit with not as pronounced gains (Figure 9 in Appendix 2.3).

The impact of rationales is highest for target datasets that exhibit a high skew in their label

[2]Additional results for the HR and PreSup rationales are in Figures 5 and 6 of the Appendix.

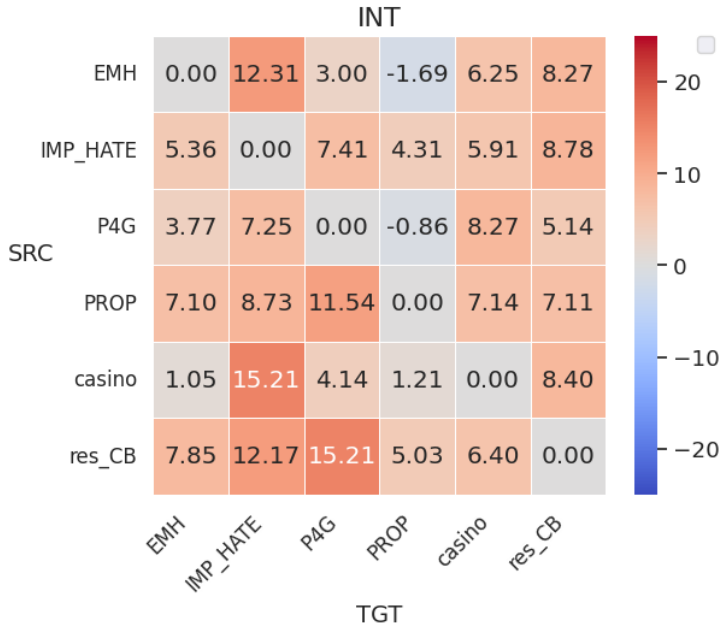


Figure 9.9: Net performance gains across different source and target tasks from adding speakers' intentions.

distribution (such as P4G, res_CB, and IMP_HATE). Label-wise F1 scores in Figures 15 and 16 reveal that the rationales improve performance for impoverished label classes such as "foot-in-the-door" for P4G, "Self-Assertion" and "Self-Pity" for res_CB, and "threatening" for IMP_HATE. We thus posit that rationales help more complex dialogue tasks for both in-domain and cross-task settings.

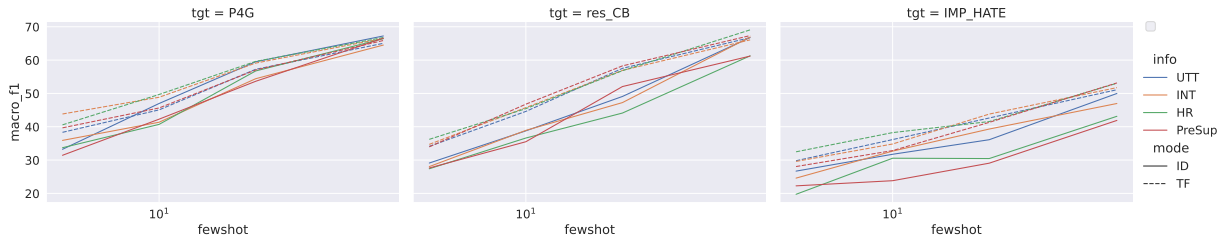


Figure 9.10: Impact of GPT-4o rationales on both in-domain (ID) and cross-task (TF) performance for PEFT-based LLama models across the three datasets for different few-shot settings.

We investigate whether a model's in-domain performance on a source task correlates with their transfer performance on a target task. Likewise, we explore whether rationales that yield in-domain gains are good predictors of transfer success. We observe negligible correlation in Table 6 on both fronts using Spearmann's ranked correlation. However, we observe from Figure 9.9 that adding intentions results in an overall positive impact for 28 of the 30 source target pairs.
PEFT Results: We also explore the impact of rationales in a PEFT (parameter efficient fine-tuning) setup. Due to the limited compute budget and large number of experiments (360 in-domain

	Gemma-2-9B-it						Llama-3-8B-it					
RAT	P4G	CaSiNo	res_CB	PROP	EMH	HATE	P4G	CaSiNo	res_CB	PROP	EMH	HATE
UTT	29.2	35.9	33.8	47.6	48.8	33.9	20.1	30.1	26.9	43.9	66.8	32.4
+ INT	31.3	38.5	35.4	**51.0**	55.0	**38.8**	20.9	31.3	**29.5**	**47.2**	67.2	32.7
+ HR	28.0	38.8	35.1	44.2	**56.6**	35.1	**21.2**	29.3	28.3	43.9	67.1	32.9
+ PreSup	32.3	40.5	**38.2**	45.2	53.3	38.3	20.1	**32.2**	28.2	45.6	**67.6**	**33.4**
+ ALL	**33.7**	**40.6**	33.9	43.8	55.5	37.1	21.1	28.9	27.9	44.8	**67.6**	31.9

Table 9.6: Zero-shot performance of models in an in-context learning setup with GPT-4o rationales.

Table 9.7: Performance of PEFT-based LLama model for different datasets when augmented with rationales corresponding to intentions, hearer reactions, and presuppositions. We present the mean performance and standard deviation across three seeds.

Rationale	P4G	res_CB	IMP_HATE
UTT	69.4 +/- 1.5	71.5 +/- 2.6	66.5 +/- 0.6
+ INT	71.2 +/- 1.6	71.3 +/- 1.9	66.0 +/- 2.0
+ HR	**72.6 +/- 1.8**	**72.8 +/- 1.8**	68.1 +/- 1.1
+ PreSup	66.6 +/- 2.7	68.7 +/- 2.0	**68.6 +/- 1.4**

and 1440 cross-task transfer runs) in the SFT setting, we experiment on only three out of six datasets, i.e., P4G, res_CB, and IMP_HATE. We chose these datasets since they had the lowest in-domain performance, and hence were the most challenging.

We report the in-domain results in Table 9.7 and the cross-task transfer performance in Figure 9. We observe trends similar to our instruction-tuned results, i.e., rationales aid dialogue understanding and generalization for PEFT based models.

9.3.2 Impact of Rationales in an ICL Setup

Intentions improve performance on target datasets 91.7% of the time in an ICL paradigm (see Tables 9.6, 8 and 9) across different few-shot settings and models. Presuppositions and hearer reactions fare better at 0-shot and 5-shot settings, respectively. Surprisingly, adding ALL does not bring significant gains as in SFT, possibly due to context-distraction (Shi et al., 2023). Table 10 in the Appendix highlights how adding rationales yields gains comparable to Chain-of-Thought (CoT) prompting. Moreover, these gains incur significantly fewer output tokens (e.g., 109.2 versus 2.1 for INT and CoT respectively, with the Gemma-2 model). Nevertheless, SFT models in a cross-task transfer setting, with only a mere 20 or 50 few-shot examples, can surpass ICL performance.

9.3.3 Factors affecting Task Performance

We inspect factors that impact performance at the instance-wise and global level for SFT and ICL. **Instance-wise Correlations:** We investigate whether certain rationale characteristics correlate with task performance. These include (i) the length of the rationale, (ii) the length of the dialogue

context, (iii) similarity between the rationale and the utterance, (iv) similarity between the rationale and the label description, (v) readability scores via Flesch's readability ease (Farr et al., 1951; Kincaid, 1975), (vi) valence, arousal, and dominance scores via the NRC lexicon (Mohammad, 2018), and (vii) emotional intensity, emotional polarity, and empathy scores (Wu et al., 2024).

We measure the point bi-serial correlation between each individual factor and instance-wise accuracy. A low (almost zero) correlation for all the factors in Table 13, signals that task performance is **not dependent on these data artifacts**. Our rationales are also **task-agnostic**; the similarity between a given rationale and the task-specific label is not predictive of task performance.

Global Generalization Characteristics: We perform a multivariate ANOVA analysis where our dependent variable is the relative change in performance from adding the rationales. Our covariates or independent variables include the rationale category, the LLM used to generate the rationales, the source dataset, and the target dataset [3], and the number of few-shot examples. We also include the pairwise interaction effects of these covariates. We note the F-statistic and their corresponding p-value for in-domain, cross-task and ICL setting respectively in Tables 14, 15, and 16 in the Appendix 2.3. We consider covariates to have a significant effect when their corresponding p-values are ≤ 0.05.

Dataset	Label	Utterance text	Rationale Text	CAT
casino	vouch-fair	hey buddy I hope we both end up with a good deal:)	Expressing hope for a **mutually beneficial outcome**	INT
IMP_HATE	white_grievance	but that wouldn't enable them to destroy white neighbourhoods .	There is a belief or concern that certain actions or policies could lead to **the destruction of white neighborhoods.**	PreSup
P4G	foot-in-the-door	Every little bit help.	EE feels reassured **that their small donation** is still valuable.	HR
P4G	foot-in-the-door	Every little bit help.	**Reassure the listener that any contribution is valuable.**	INT
res_CB	Self Pity	at this i can only pay about 1600 could you do that	Seller **realizes the buyer's budget constraints.**	HR
res_CB	Source Derogation	Yes. What didn't your wife like about the bed?	Seller feels questioned about the **reason for selling the bed.**	HR

Table 9.8: We present instances across different datasets where adding the rationale information was crucial in predicting the correct label always. We compute Shapley values for each token in the rationale to observe its contribution to the model's decision; the highlighted portions correspond to high positive associations with the label.

In a nutshell, across all the different experimental setups, **the rationale category significantly influences task performance.** Unanimously across all settings, intentions yield the highest positive gains on average, followed by the hearer's reactions and then the presuppositions. We summarize the fraction of cases where adding rationales improves task performance for both SFT , which includes in-domain (ID) and cross-task transfer (TF) settings, and ICL setups in Figure 9.11.

[3]We have the source dataset only for cross task transfer

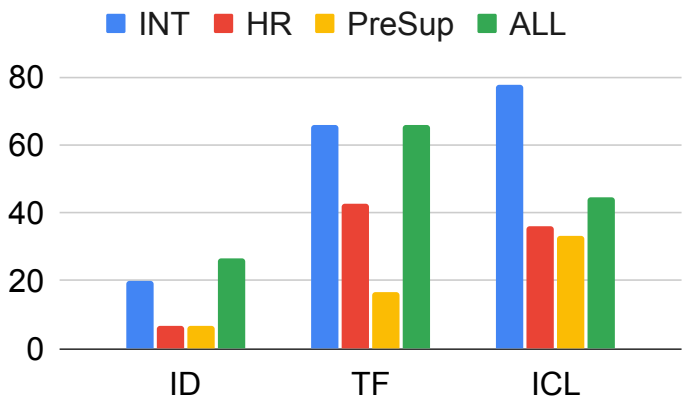


Figure 9.11: Fraction of cases where rationales improves performance for in-domain (ID), cross-task transfer (TF), and in-context learning settings (ICL).

9.3.4 Necessity and Sufficiency of Rationales

Having demonstrated the practical utility of adding rationales, we now examine if the information encoded in the rationale is sufficient or necessary.

Sufficiency Claims: We investigate the sufficiency claims of rationales, i.e. whether the rationales can meaningfully capture all the information in the utterance. We carry out two ablation experiments to examine the relative change in task performance compared to the baseline (i.e. when only the utterance is included). In the first experiment, we train the model using both the corresponding rationale and utterance, but provide only the rationale information during testing. In the second experiment, we omit out the utterance completely and train on only the rationales. For both cases, task performance degrades significantly highlighting that **rationales are insufficient by themselves and cannot match the baseline task performance**.

Necessary Claims: We investigate whether the rationale text is useful or necessary in guiding model prediction. We perform sensitivity analysis by perturbing the rationale in different ways such as synonym replacement or deletion. Additional details of our experiment appear in Appendix 2.3. We note a deterioration in task performance as the proportion of text perturbed increases; specifically, deletions have the greatest impact while synonym replacement has the least (see Figure 14). Our findings thus highlight that models do indeed rely on the text in the rationales for classification.

9.3.5 Qualitative Analysis

We conduct qualitative analysis to investigate the cases where rationales actively improve the model's predictions. We consider only those instances where the baseline (i.e., only the utterance text) fails to predict the correct label a majority of times, but succeeds with the rationale. We restrict our analysis predominantly to in-domain cases to avoid conflating the source's influence (as in the transfer setting) on the target task's performance.

The rationale with the greatest impact on performance is dependent on the nature of the task. Hearer reactions or HR has the highest impact on P4G, possibly because it captures the

thought processes of the persuadee (EE) as they are being persuaded to donate. E.g., the utterance *"Anything would help even small donations add up when everyone pitches in."* evokes a sense of reassurance from EE that any contribution is valuable and is recognized as a "foot-in-the-door" strategy. Presuppositions or PreSup are useful for IMP_HATE, a dataset that directly references stereotypes and thus requires generic knowledge to infer the type of implicit hatred. Tasks that are geared towards the speakers' interests, i.e., strategies employed to resist persuasion (res_CB), or signaling empathy to someone in therapy (EMH) benefit mostly from intentions. Furthermore, similar tasks, e.g., CaSiNo and res_CB which deal with negotiation have similar relative performance for the same rationales.

Rationales corresponding to different categories will likely yield different predictions, despite being sound or relevant. We hypothesize that **certain tokens in the rationale might facilitate predicting the label category**. E.g., the phrase *"feels questioned"* in the HR for the res_CB example in Table 9.8 hints at source derogation, which we did not observe for the other rationale categories. Likewise, the wording of *"how one might treat a dog"* in the PreSup for IMP_HATE conveys a sense of inferiority more prominently than generic mistreatment.

We carry out interpretability analysis using SHAPLEY (Roth, 1988) for instances where the rationales consistently yielded the correct answer. We observe the SHAPLEY values for the highlighted tokens in the rationales that guide model prediction. We present examples spanning different rationales and datasets in Table 9.8 with additional examples in Table 17 in the Appendix. We observe that the highlighted tokens in the rationale text align with human intuition to explain the label category. For example, the phrase *"destruction of white neighborhoods"* acts as a signal for white aggression and *"that their small donation"* for foot-in-the-door strategy, respectively, in Table 9.8.

9.4 Conclusion

In this chapter we introduce SOCIAL SCAFFOLDS, a prompting framework motivated by narrative modeling principles, to generate rationales that capture multiple perspectives. Our comprehensive evaluation suite that spans 5,400 supervised fine-tuning and in-context learning experiments and demonstrate that rationales aid task performance in both experimental setups. In particular, incorporating only the speaker's intentions and all three rationale categories yields significant cross-task transfer gains (31.3% and 44.0% of the times). Our analysis also reveals that rationales are task-agnostic and complement the utterance. We thus concludes the second part of our dissertation, i.e. the role of informal scaffolds in aiding generalization across tasks.

Part III

Future Directions

Chapter 10

Towards a Holistic Evaluation of Generalization in Pretrained Language Models: A Case Study of NLI and MRC

This stand-alone chapter presents the first step to systematically evaluate the generalization capabilities of pretrained language models. Following Hupkes et al. (2023), we investigate three kinds of generalization common in NLP, i.e. domain adaptation, robustness, and compositional generalization for two NLU tasks, i.e. natural language inference (NLI) and machine reading comprehension (MRC). For each task, we have a representative dataset that serves as a source and other target datasets that correspond to different generalization categories. Our generalization experiments are based on transformer architectures spanning different families (encoder-only or decoder-only), sizes (base or large), and training strategies (full fine-tuning or PEFT). We design a comprehensive evaluation suite to investigate whether pretrained language models can generalize across all scenarios or not. The remaining chapter outlines on the experimental setup including the different datasets and models we explore and the corresponding analyses and results.

10.1 Tasks, Datasets & Models

We consider two representative NLU tasks: NLI and MRC. The NLI task involves determining if the meaning of one text fragment (hypothesis) can be inferred from another (premise). Independent of any specific application, this task is designed to encapsulate the essential inferences about the variability of semantic expression frequently required for various settings (Dagan et al., 2006). MRC is another common task – many NLU tasks have been formulated as MRC (He et al., 2015) or models trained on MRC format data have shown good performance on NLU tasks (McCann et al., 2018). We use the extractive version of MRC, where the input consists of a context (passage) and a question, and the answer has to be extracted from the context.

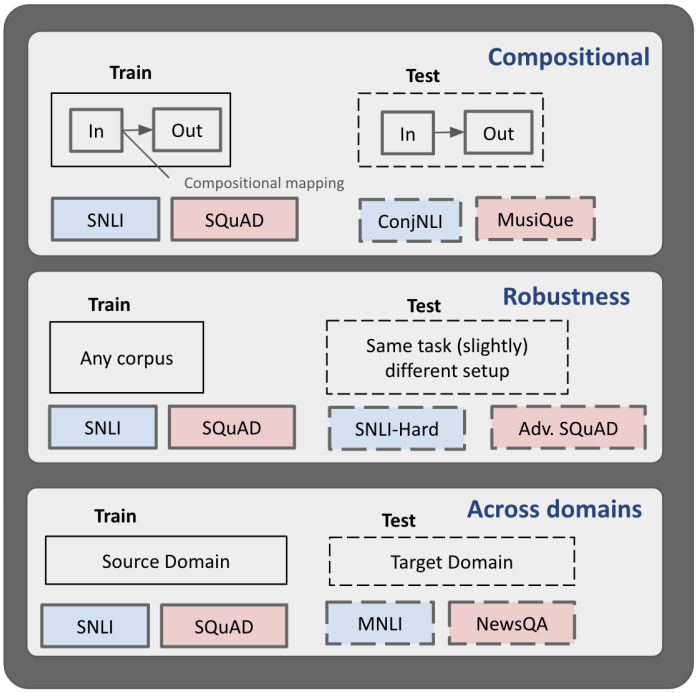


Figure 10.1: Hupkes et al. (2023) categorizes the generalization scenarios in NLP into *six* types. We chose *three* that cover many important scenarios. We trained models on SNLI and SQuAD, and tested them on various datasets corresponding to these dimensions. The datasets were chosen so as not to confound the dimensions. For example, the compositional test dataset for MRC (MusiQue) is a derivative of the source dataset SQuAD – there is no domain shift, and the dataset does not contain robustness testing perturbations.

10.1.1 NLI Datasets

We consider SNLI (Bowman et al., 2015) as the source dataset, which is annotated with the labels corresponding to whether the hypothesis entails, is neutral, or contradicts the premise.

- **Domain:** We use both the matched and mismatched splits of the Multi-Genre NLI (MNLI) dataset (Williams et al., 2018) to test the generalization of an SNLI-trained model to different domains. We also use the TaxiNLI dataset (Joshi et al., 2020) that provides a hierarchical taxonomy of a subset of the MNLI dataset and categorizes the data points based on whether they require linguistic, logical, or world knowledge.
- **Robustness:** We cover the robustness scenarios by testing the models on four datasets. SNLI-H (Gururangan et al., 2018) is a set of SNLI test instances that common heuristics can not classify. The SNLI-CF dataset (Kaushik et al., 2019) comprises of "counter-factual" perturbations, where the annotators are asked to make minimal changes to an instance such that the label changes – a model can only classify these instances correctly if it understands the reasoning behind the NLI task. SNLI-BT is generated by back-translating the original SNLI test instances from `En->Pt->En` using a pre-trained multi-lingual BART model – this tests the models' ability to generalize against adversarial perturbations. Finally, HANS (McCoy et al., 2020b) is built from

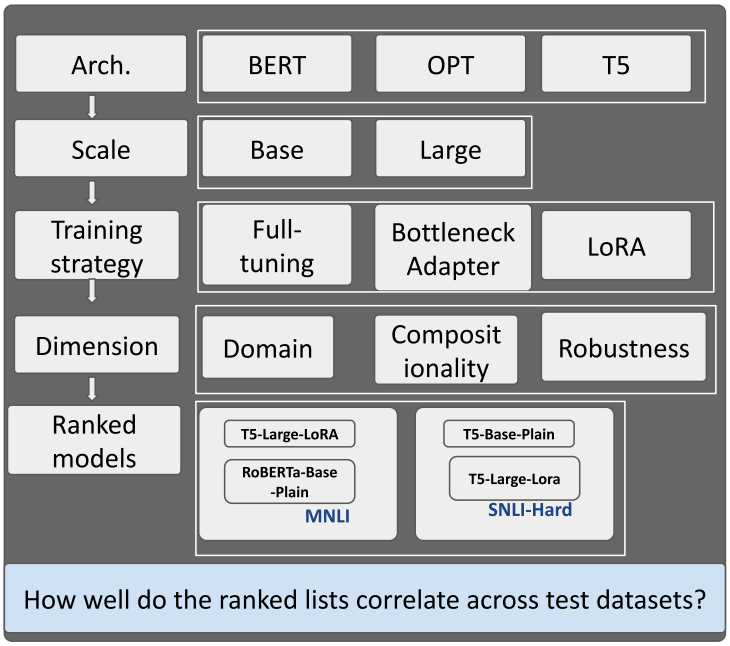


Figure 10.2: Our framework: we train 72 models on 2 base datasets, test them on 15 datasets corresponding to different dimensions of generalization, and analyze the results.

templates constituting different syntactic heuristics in NLI, such as lexical overlap or common subsequences between the premise and hypothesis.

- **Compositionality:** It is non-trivial to meaningfully combine SNLI instances, but in a compositional NLI dataset such as MoNLI (Geiger et al., 2020) all words or phrases of a composed instance come from SNLI. Consider a sentence from SNLI "The children are holding plants". Assume the phrase "flowers", which is a hyponym (per Wordnet) to the phrase "plants", appears in SNLI. Now the pair (premise: "The children are holding flowers", and hypothesis: "The children are holding plants") will have an entailment relation as every flower is a plant. Consequently, the label would change to neutral when the premise and hypothesis are reversed. Since the phrase that determines this relation exists in SNLI, the new dataset is merely a composition of the known constituents.[1] CONJNLI (Saha et al., 2020) focuses on conjunctive sentences – premises and hypotheses vary through the addition, removal, or substitution of conjuncts such as "and," "or", "but", and "nor" alongside elements like quantifiers and negations. This also presents a challenge in compositional generalization.

10.1.2 MRC Datasets

We train the MRC models on a popular extractive dataset SQuAD (Rajpurkar et al., 2016).

- **Domain:** NewsQA is a crowd-sourced dataset of approximately 100K human-generated QA

[1]This is the *PMoNLI* part of the dataset. Negations would change the direction of the monotone operator: *not* holding plants ⇒ *not* holding flower, but not the other way around. These instances comprise the *NMoNLI* dataset, which we do not use.

pairs, where the context comes from 10K news articles from CNN. In SQuAD contexts are paragraphs from Wikipedia articles, therefore NewsQA presents a significant domain shift.

- **Robustness:** Adversarial Squad (Adv-SQuAD) is a robustness challenge set built on SQuAD insofar it adds a sentence that contains a phrase that a shortcut-dependent model (eg., one that chooses a phrase that is proximal to a key phrase from the question) would select (Jia and Liang, 2017). The HotpotQA dataset (Yang et al., 2018) was designed to test the multi-hop reasoning abilities of MRC models, i.e., a model should only be successful if it understands relations between entities that span multiple sentences. Similar to Jia and Liang (2017), Jiang and Bansal (2019) built a challenge set (Adv-HotpotQA) by adding a new passage to the context with a fake answer. The modifications in both Adv-HotpotQA and Adv-SQuAD do not change the original answer. Therefore, a model using the expected reasoning strategies would still be able to answer correctly, but a model dependent on shortcuts would fail.
- **Compositionality:** MusiQue (Trivedi et al., 2022) is designed to test compositionality in reading comprehension. The dataset is built on multiple MRC datasets (SQuAD, HotpotQA and three others) in a "bottom-up" approach. Pairs of *connected* single-hop questions are combined to create 2-hop questions first and are subsequently combined to produce k-hop questions recursively. We only choose the questions that are produced by combining SQuAD questions.

We use the validation or test (when available) split of the generalization datasets. In NLI, most datasets for compositional and robustness generalization are derivatives of the SNLI dataset itself, except for HANS and CONJNLI. They come from non-SNLI sources, but the distribution is not significantly different. *This allows us to not confound different dimensions of generalizability*. This is true for MRC as well, Adv-SQuAD and MusiQue (the portion we use) come from the base dataset SQuAD, and both Adv-HotpotQA and SQuAD come from the same domain. HANS has 2 labels (as opposed to 3 for SNLI), so the predicted labels of neutral and contradiction are merged. For consistency, we only use instances with a max tokenized sequence length of 512 (see the appendix for details).

10.1.3 Models & Training

We explore three popular families of transformer-based neural architectures, i.e., encoder-only (EO), decoder-only (DO), and encoder-decoder (ED) models. As the most popular/powerful representative for each architecture, we include `RoBERTa` (Liu et al., 2019b) and `BERT` (Devlin et al., 2019a) for EO, `OPT` (Zhang et al., 2022b) for DO, and `T5` (Raffel et al., 2020) for (ED).

NLI is modeled as a sequence classification problem, and a linear layer is used as the classifier over the base encoders. MRC is modeled as a token classification problem with a linear layer, and the models are trained to predict a token's probability for being the start and end of an answer phrase (Devlin et al., 2019a). We use the base and large versions for each model, and specifically for `BERT` these are the `cased` ones.

The models are trained by changing the full parameters as well as a fraction of them using two PEFT methods: Bottleneck adapters (Houlsby et al., 2019b) and LoRA (Hu et al., 2021b). Adapters introduce bottleneck feed-forward layers in each layer of a transformer model as the only trainable parameters. These adapter layers consist of a down-projection matrix $W_{\text{down}} : (d_{\text{hidden}}, d_{\text{bottleneck}})$, a RELU non-linearity ($f$) and an up-projection matrix $W_{\text{up}} : (d_{\text{bottleneck}}, d_{\text{hidden}})$,

with the final equation: $h \leftarrow W_{\text{up}} \cdot f(W_{\text{down}} \cdot h)$. We use a reduction factor ($\frac{d_{\text{hidden}}}{d_{\text{bottleneck}}}$) of 16 for all models. Similar to Bottleneck adapters, LoRA injects trainable low-rank decomposition matrices into the layers of a pre-trained model. Any linear layer of the form $(h = W_0 x)$ is re-parameterized as: $h = W_0 x + \frac{\alpha}{r} BAx$ where $(A \in R^{r \times k})$ and $(B \in R^{d \times r})$ are the trainable decomposition matrices and r is the low-dimensional rank of the decomposition. We set the rank at 16 and α at 32.

Each model is initialized with three seeds, and the training data sequence is shuffled. The models are trained with AdamW (Loshchilov and Hutter, 2019) optimizer, batch sizes varying between 32 and 64, and a learning rate of 2e-5 with a stepwise learning rate decay (Howard and Ruder, 2018) using the HuggingFace Transformers library (Wolf et al., 2019) .

10.2 Results

10.2.1 RQ1: Does one model instance generalize well across generalization dimensions?

Our first hypothesis is a model instance [2] generalizes well across different types. We test this by investigating whether the rankings of model instances are consistent, i.e. are well-correlated, across datasets that characterize different types of generalization.

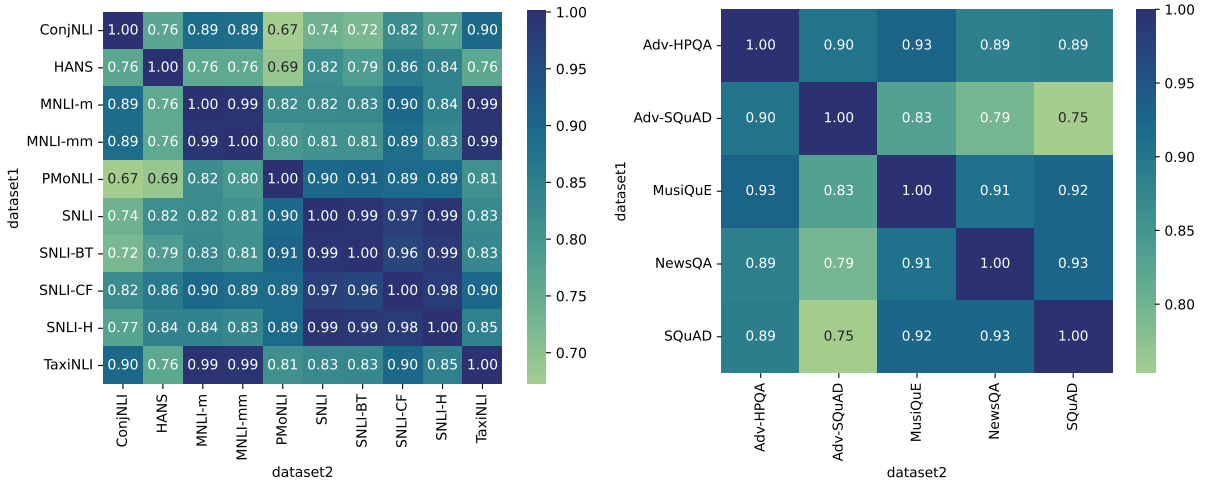
We evaluate 72 model instances on each dataset corresponding to a task. Subsequently, for a given dataset pair in a task, we compute Spearman's rank correlation coefficient (ρ) of the corresponding model instances' scores (accuracy for NLI and F1-Score for MRC) for the two datasets. We are more interested in the rankings (relative performance) of model instances than the absolute scores since the datasets are not well calibrated amongst themselves. We present a heatmap of the correlation scores between pairs of datasets for NLI and MRC in Figures 10.3a and 10.3b, respectively.

We observe a strong to very-strong correlation ($\rho \geq 0.6$) [3] for all dataset pairs for both NLI and MRC tasks. For each of these comparisons, the correlation was statistically significant with a p-value lower than 0.05, **implying that we can reject the null hypothesis that the performances of model instances are not monotonically correlated**.

For NLI, the datasets derived from the same source, e.g., SNLI-CF, SNLI-BT, and PMoNLI from SNLI, or datasets that are created in a similar fashion like matched and mismatched splits of MNLI exhibit very strong correlation ($\rho \geq 0.90$). On the other hand, datasets derived from a different source like Wikipedia for CONJNLI or constructed in a templatized fashion like HANS demonstrate a more uniform correlation. We thus infer that the rankings of model instances depend more on the source than the type of generalization for NLI. For example, although PMoNLI and CONJNLI both test compositionality, the instances have the lowest correlation score ($\rho = 0.67$).

[2] 1. **model instance**: a particular instance of a trained model, e.g., a $\texttt{T5}_{\texttt{base}}$ model with LoRA trained on SNLI with a seed of 42. 2. **architecture**: model architecture, e.g., `RoBERTa`, `T5`. 3. **model configuration**: a combination of architecture-size-training strategy ($\texttt{T5}_{\texttt{base}}$ fully fine-tuned). 4. **architecture family**: types of architectures – encoder only (`BERT`, `RoBERTa`)/decoder-only (`OPT`).

[3] `https://www.statstutor.ac.uk/resources/uploaded/spearmans.pdf`



(a) NLI Datasets (b) MRC Datasets

Figure 10.3: Spearmann's Rank Correlation ρ between the source and the target datasets for NLI and MRC on a per-instance basis.

Table 10.1: Performance of NLI models when trained on the SNLI and evaluated on different datasets in terms of accuracy. We report the mean and standard deviation across three seeds. The best model is highlighted in bold, the second-best model is underlined, and the worst model is highlighted in red. `Adap` and `LoRA` refers to the adapter and LoRA training strategies.

	ID	**OOD**			**Robustness**				**Compositionality**	
Model	SNLI	MNLI-m	MNLI-mm	TaxiNLI	SNLI-BT	SNLI-CF	SNLI-H	HANS	ConjNLI	PMoNLI
$\text{BERT}_{\text{base}}$+ Adap	85.1±0.1	65.1±0.1	68.0±0.1	64.7±0.1	80.0±0.1	68.5±0.1	71.0±0.2	50.0±0.0	52.2±0.7	91.7±1.1
$\text{BERT}_{\text{base}}$+ LoRA	81.3±0.2	59.2±0.5	61.1±0.1	54.6±0.6	76.6±0.2	64.2±0.3	65.7±0.5	50.0±0.0	49.1±1.4	85.9±0.5
$\text{BERT}_{\text{base}}$	90.6±0.1	73.5±0.4	73.6±0.2	73.4±0.1	84.3±0.2	76.1±0.2	80.2±0.1	58.1±1.2	58.6±0.6	95.1±0.3
$\text{BERT}_{\text{large}}$+ Adap	88.8±0.2	72.8±0.8	73.2±0.8	72.8±1.0	83.1±0.2	73.3±0.2	77.3±0.3	50.3±0.4	56.7±1.1	96.1±0.3
$\text{BERT}_{\text{large}}$+ LoRA	86.2±0.4	68.3±0.5	69.2±0.7	67.7±1.5	80.9±0.1	69.3±0.4	73.1±0.6	50.1±0.2	54.3±1.5	94.7±1.2
$\text{BERT}_{\text{large}}$	91.1±0.1	76.6±0.1	76.2±0.3	76.5±0.4	84.7±0.1	77.4±0.3	81.7±0.2	58.1±1.2	61.1±0.8	97.6±0.4
$\text{RoBERTa}_{\text{base}}$+ Adap	88.3±0.1	75.8±0.6	75.9±0.3	74.4±0.2	83.0±0.0	72.9±0.2	76.1±0.2	50.3±0.1	54.8±0.6	95.1±0.1
$\text{RoBERTa}_{\text{base}}$+ LoRA	87.1±0.0	73.6±0.0	74.9±0.1	72.3±0.3	81.8±0.0	71.8±0.2	75.2±0.1	50.1±0.0	52.2±0.9	94.4±0.2
$\text{RoBERTa}_{\text{base}}$	91.4±0.0	80.2±0.2	79.9±0.2	80.1±0.2	85.2±0.1	77.9±0.1	82.1±0.1	65.9±2.0	60.8±0.4	96.6±0.2
$\text{RoBERTa}_{\text{large}}$+ Adap	91.7±0.0	83.8±0.4	83.0±0.4	83.9±0.1	85.4±0.0	79.9±0.5	82.4±0.1	67.8±1.4	61.4±0.2	**98.5±0.1**
$\text{RoBERTa}_{\text{large}}$+ LoRA	90.8±0.1	81.7±0.4	81.8±0.2	81.1±0.5	84.5±0.1	78.8±0.2	81.0±0.1	65.3±0.8	58.5±0.9	98.0±0.2
$\text{RoBERTa}_{\text{large}}$	**92.6±0.0**	85.0±0.0	84.3±0.1	85.0±0.1	**85.7±0.0**	**81.3±0.2**	**84.7±0.0**	**73.7±1.0**	65.5±0.3	**98.5±0.1**
OPT_{base}+ Adap	82.8±3.0	56.7±1.8	57.5±1.9	55.2±3.7	77.5±2.8	66.7±2.4	68.6±3.1	52.3±3.3	49.2±4.3	88.4±2.3
OPT_{base}+ LoRA	78.1±3.7	53.8±1.5	55.7±2.3	52.8±1.1	72.4±4.0	63.2±2.3	65.0±2.9	50.4±0.6	47.4±1.9	86.6±3.1
OPT_{base}	89.6±0.1	71.3±0.7	72.9±0.9	71.3±0.9	83.7±0.2	74.5±0.3	78.8±0.1	59.1±4.2	57.5±0.3	95.6±0.8
$\text{OPT}_{\text{large}}$+ Adap	88.6±0.2	66.6±1.3	69.2±0.8	66.0±2.1	81.9±0.5	73.4±0.3	77.5±0.2	61.7±6.8	55.4±1.0	90.9±1.5
$\text{OPT}_{\text{large}}$+ LoRA	83.6±2.2	63.5±3.6	65.0±3.4	60.7±4.7	78.0±2.5	69.5±1.2	71.4±2.1	60.1±2.3	56.7±3.1	91.9±3.0
$\text{OPT}_{\text{large}}$	90.4±0.4	75.5±0.4	77.3±0.3	75.4±0.3	84.1±0.3	76.5±0.8	80.7±0.5	65.8±0.6	60.7±1.3	95.2±2.0
T5_{base}+ Adap	88.6±0.0	80.1±0.1	80.3±0.1	80.3±0.3	82.9±0.0	74.8±0.2	77.7±0.1	60.2±0.1	64.0±0.9	94.6±0.4
T5_{base}+ LoRA	85.8±0.0	80.6±0.4	80.9±0.3	80.6±0.5	80.7±0.2	72.8±0.2	74.1±0.3	57.2±0.7	65.2±0.7	92.1±0.8
T5_{base}	89.7±0.1	81.4±0.1	80.9±0.2	81.2±0.1	83.7±0.1	75.9±0.2	79.5±0.1	63.3±0.3	65.2±0.9	95.3±0.3
T5_{large}+ Adap	91.8±0.0	86.2±0.1	85.5±0.3	86.6±0.4	85.4±0.1	80.3±0.3	82.7±0.1	68.2±1.1	66.0±0.1	98.1±0.2
T5_{large}+ LoRA	90.5±0.0	**87.5±0.1**	**87.5±0.3**	87.8±0.3	84.2±0.0	79.4±0.1	81.0±0.1	64.7±0.1	66.3±0.5	98.1±0.1
T5_{large}	92.1±0.1	87.3±0.1	86.8±0.2	**87.9±0.2**	85.5±0.0	81.0±0.2	83.3±0.1	71.6±0.6	**67.2±0.3**	98.0±0.1

Table 10.2: Performance of MRC models when trained on the SQuAD (ID) and evaluated on different datasets. We report the mean F1 score across three seeds (the stds vary between 0.0 and 3.2). The best model is highlighted in bold, the second-best is underlined, and the worst is highlighted in red. OOD, Rob, and Comp imply generalization across domains, robustness, and compositionality, respectively. Adap and LoRA refers to the adapter and LoRA training strategies.

	OOD	**Rob**		**Comp**	**ID**
Model	NQA	AHQ	ASQ	MsQ	SQ
$\text{BERT}_{\text{base}}$+ Adap	52.7	22.9	45.5	41.1	77.8
$\text{BERT}_{\text{base}}$+ LoRA	12.8	9.7	17.9	12.8	24.7
$\text{BERT}_{\text{base}}$	62.2	34.7	61.8	50.2	87.6
$\text{BERT}_{\text{large}}$+ Adap	60.1	25.0	64.3	50.2	85.6
$\text{BERT}_{\text{large}}$+ LoRA	42.2	17.0	46.3	37.0	67.1
$\text{BERT}_{\text{large}}$	65.2	39.4	72.5	62.2	90.7
$\text{RoBERTa}_{\text{base}}$+ Adap	55.0	26.3	63.5	51.5	85.8
$\text{RoBERTa}_{\text{base}}$+ LoRA	43.6	22.2	50.2	47.3	78.7
$\text{RoBERTa}_{\text{base}}$	63.3	39.0	73.0	61.4	92.0
$\text{RoBERTa}_{\text{large}}$+ Adap	66.8	46.6	82.5	65.5	93.4
$\text{RoBERTa}_{\text{large}}$+ LoRA	54.3	34.8	70.7	57.8	88.7
$\text{RoBERTa}_{\text{large}}$	**70.0**	**51.4**	**84.1**	**74.6**	**94.6**
OPT_{base}+ Adap	48.4	31.0	64.5	40.9	75.2
OPT_{base}+ LoRA	47.5	25.9	61.8	41.0	71.9
OPT_{base}	58.9	37.7	78.6	59.0	83.6
$\text{OPT}_{\text{large}}$+ Adap	55.1	34.7	79.0	47.0	83.5
$\text{OPT}_{\text{large}}$+ LoRA	57.9	33.5	79.0	45.6	83.3
$\text{OPT}_{\text{large}}$	62.4	42.0	81.6	68.7	85.9
T5_{base}+ Adap	67.2	37.8	74.2	61.1	90.3
T5_{base}+ LoRA	64.8	33.6	69.8	57.8	87.5
T5_{base}	67.5	38.6	74.8	64.0	90.9
T5_{large}+ Adap	69.7	46.5	82.3	69.9	93.7
T5_{large}+ LoRA	69.5	42.8	79.6	68.4	92.8
T5_{large}	69.9	47.9	**84.1**	73.6	93.9

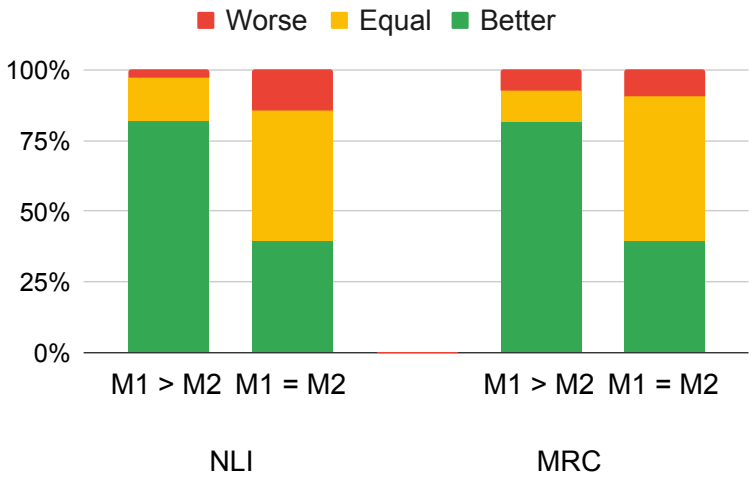
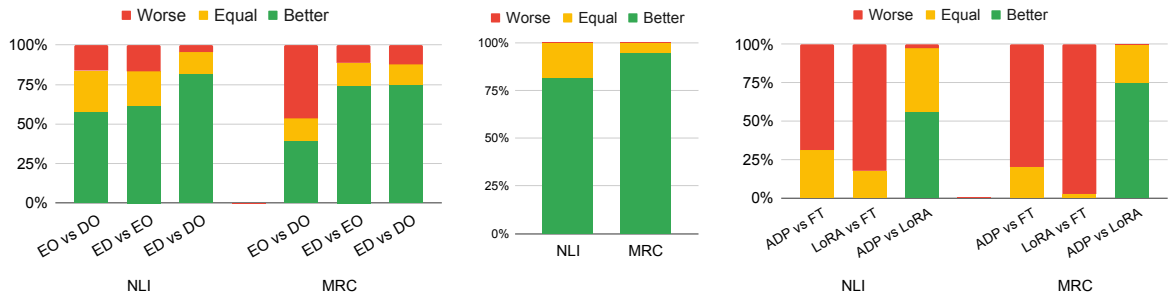


Figure 10.4: Fraction of cases where one model is significantly better, worse, or as good as the other on different target datasets. We consider two scenarios, (i) where one of the models was already significantly better on the source dataset ($M_1 > M_2$) and (ii) where the models had similar source performance ($M_1 = M_2$).



(a) Architecture Types (b) Model Scale (c) PEFT

Figure 10.5: Fraction of times the given architecture configuration or training strategy is statistically better, equal, or worse for the two tasks of NLI and MRC.

However, this observation is not as pronounced for MRC, where the model rankings correlate more with the generalization type than the dataset source. For example, we observe a higher correlation between Adv-HotpotQA and Adv-SQuAD ($\rho = 0.90$) than between Adv-SQuAD and SQuAD ($\rho = 0.75$). We also note a higher correlation across domains for MRC ($\rho = 0.92$ between SQuAD and NewsQA) than for NLI ($\rho \approx 0.8$ between MNLI and SNLI).

Having ranked the model instances in decreasing order of performance for each of the 10 NLI datasets, we can obtain a global (or unified) ranked list by aggregating these individual rankings. We employ the MC4 algorithm of Dwork et al. (2001) that constructs the ranking preferences based on a simple majority vote across the individual rankings to obtain the aggregated ranked list of instances. We do the same for the 5 datasets to create an aggregate ranked list for MRC. Spearmann's rank correlation coefficient between these two aggregated ranked lists for MRC and NLI is 0.93, which implies that the model instances also exhibit high correlation across tasks.

10.2.2 RQ2: Do model configurations generalize well across scenarios?

We extend our previous hypothesis to investigate whether certain model configurations (a combination of model architectures, scale, and training strategies) generalize well across different scenarios. We start by averaging the performance of a model configuration (architecture-size-training strategy combination) across three seeds and report the results in Tables 10.1 and 10.2 for NLI and MRC, respectively. Interestingly, we do not see a significant variation across instances from different seeds (as evidenced by low standard deviations) – a finding different from prior work of McCoy et al. (2020a).

We also compute the Spearman's rank correlation coefficient between two dataset pairs for NLI and MRC in Figures 10.7a and 10.7b respectively. The heatmaps indicate a strong positive correlation ($\rho \geq 0.7$) between all dataset pairs and inform us that the relative performance of these model configurations remains consistent across the target datasets and domains.

We further carry out a pair-wise comparison of model configurations to investigate whether the relative performance of a model pair on the source dataset (SNLI and SQuAD for NLI and MRC, respectively) persists across different target datasets. Simply put, if the performance of a model M_1 is significantly better than M_2 on the source dataset, does the situation remain the same across other targets? We adopt the non-parametric paired bootstrap test of Berg-Kirkpatrick et al. (2012) to check for statistical significance (p-value ≤ 0.05) in line with prior work (Dror et al., 2018). We note that M_1 has a similar performance with M_2 if we cannot reject the null hypothesis that one has a significantly higher performance than the other.

Figure 10.4 illustrates the fraction of cases where the relative performance of a model architecture pair is better, worse, or the same on the target datasets compared to the original source conditions. We observe that the models retain their relative performance for a majority of cases for both NLI and MRC, i.e. if M_1 is significantly better than M_2 on the base dataset, it will follow a similar trend across targets and vice versa. The notable exceptions are the PEFT-tuned versions of `T5` model which exhibit significantly higher performance than other models (such as `BERT` or `OPT` variants) on the target datasets for NLI despite a significantly worse performance on the SNLI source dataset. A similar finding holds for the fully-tuned `OPT` models that significantly outperform others (such as `BERT` and `T5`-PEFT variants) on MRC datasets.

10.2.3 RQ3: Architecture, Scale, and PEFT

Model Architecture: From Tables 10.1 and 10.2, we see that when controlled for the model size (base v large) and training strategy (full vs PEFT), certain models almost always perform better than the others, e.g., in NLI, the base versions of `T5` models (ED) are better than `RoBERTa` (EO) models in 7 out of 9 datasets, and `RoBERTa` is better than `OPT` (DO) in 8 out of 9. To formalize this, we compare the performance of a pair of models from different architectures (e.g., $\texttt{T5}_{\texttt{base}}$ vs. $\texttt{OPT}_{\texttt{large}}$) for a given dataset. Each architecture has instances from all sizes and training strategies, so we do not have to control for them explicitly.

We adopt the paired bootstrap test to compute the fraction of datasets where models corresponding to one family (say EO) are significantly better, worse, or equal compared to models of another family (say ED). Overall, we observe (Figure 10.5a) that ED models outperform both the EO and DO significantly on both tasks. On the other hand, models corresponding to the EO fare

better for NLI as opposed to DO and vice-versa for MRC.

Scale: We compute the fraction of cases where the large variant of a model architecture is significantly better, worse, or equal to the corresponding base variant for a given dataset and task while controlling for the training strategy. Figure 10.5b shows that for both tasks, the large variants of models are significantly better than their corresponding base variants in a huge majority of cases. In fact, the base variant is never significantly better, although there are a few ties. This performance gain is also significantly higher in the generalization datasets compared to the base ones.

Parameter efficient fine-tuning (PEFT): We also explore whether PEFT models (i.e., Adapters and LoRA) are more adept at generalization than the corresponding fully fine-tuned (FT) variants. For each model pair, we compute the fraction of cases where the PEFT variant, i.e., Adapter vs. FT or LoRA vs. FT, was significantly better, equal, or worse than the corresponding fine-tuned variant. Figure 10.5c shows that PEFT models are indeed significantly worse. Moreover, this poorer performance is more pronounced for the LoRA models than for Adapters, such that adapter models are significantly better than LoRA models for both tasks.

10.2.4 RQ4: Difficult types of generalization

We inspect the absolute generalization performance of models on different datasets to investigate whether certain generalization categories or dimensions are more challenging than others. We characterize a dataset to be challenging for a given model based on the relative drop in performance of the model on the dataset compared to its' source performance (e.g., the performance of a model on SNLI and SQuAD respectively). We coin this performance difference as normalized source drop or NSD (Calderon et al., 2023) defined below, where M_s and M_t correspond to the performance of the model on the source and the target, respectively.

$$NSD = \frac{M_t - M_s}{M_s}$$

We carry out a two-way ANOVA analysis with NSD as the dependent variable with the generalization category (OOD, robustness, compositionality, or in-domain), architecture type (EO, ED, or DO), scale (large or base), and training strategy (FT, LoRA, or Adapter) as the independent covariates. We observe a significant association for all the covariates (p-value ≤ 0.05), with the generalization category exhibiting the greatest significance, followed by the architecture type, training strategy, and scale for MRC. NLI exhibits a similar trend, with the only difference being that the scale is more significant than the training strategy.

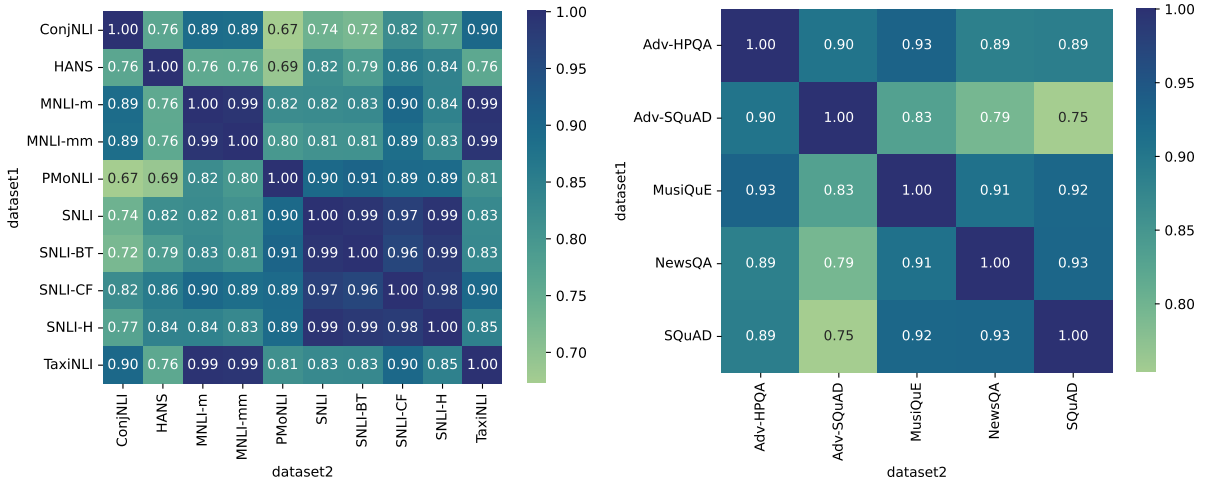
Considering the in-domain category (i.e., performance on the base dataset) as the baseline, we observe a negative correlation for all the other generalization categories. The robustness category is the most challenging (with a larger negative coefficient), followed by compositionality and OOD for MRC. For NLI, the robustness category again incurs the highest negative correlation, followed by OOD and compositionality. We hypothesize that the general prowess of models on the PMoNLI dataset, surpassing even the ID performance, is responsible for the skewed trend. We also observe positive coefficients for the larger model variant, the ED model family, and the fully fine-tuned (FT) training strategy which is consistent from our past observations. We present the intercept values of our analysis in Table 10.3.

Category	NLI	MRC
Intercept	-0.052	-0.015
Gen-type: Comp	-0.132	-0.354
Gen-type: ROB	-0.170	-0.388
Gen-type: OOD	-0.158	-0.313
Arch-family: ED	0.073	0.023
Arch-family: EO	0.024	-0.047
Fine-tuning: FT	0.023	0.047
Fine-tuning: LoRA	-0.00	-0.020
Scale: Large	0.028	0.047

Table 10.3: Coefficients for the ANOVA analysis for NLI and MRC.

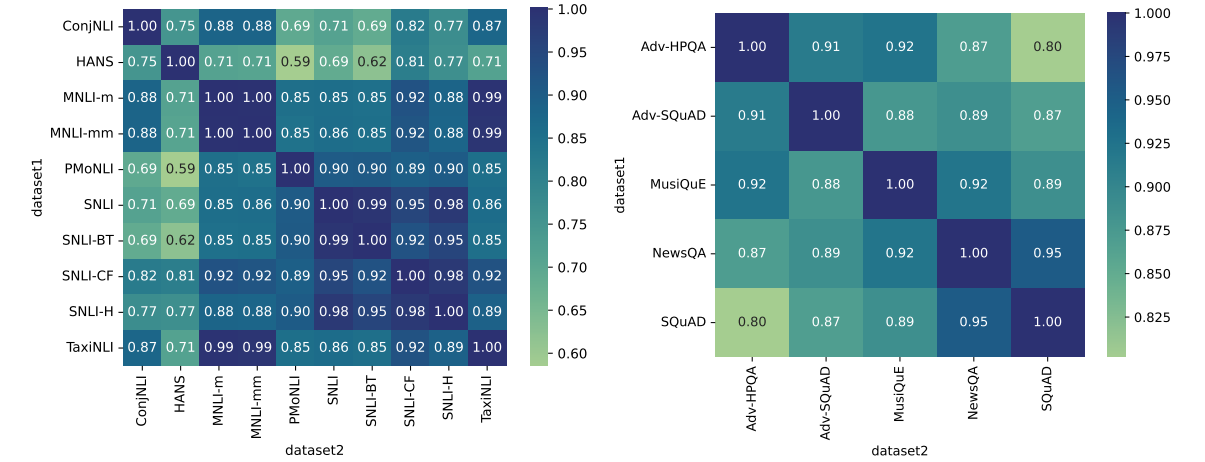
10.3 Conclusion and Takeaways

We present a systematic study on the multi-dimensional (domain, robustness, and compositional) generalization abilities of common models used in NLP. Our main conclusions are: 1. Generalizability is a model instance characteristic and not generalization type-dependent – an instance typically does not generalize well in one dimension and poorly in others. 2. It is well correlated with model size, and certain architectures and training strategies generalize better than others. 3. Certain dimensions of generalization is harder to achieve compared to the others. We hope to inspire future work that looks further into the multi-dimensional aspect of generalizability and tries to understand why certain models generalize better than others.



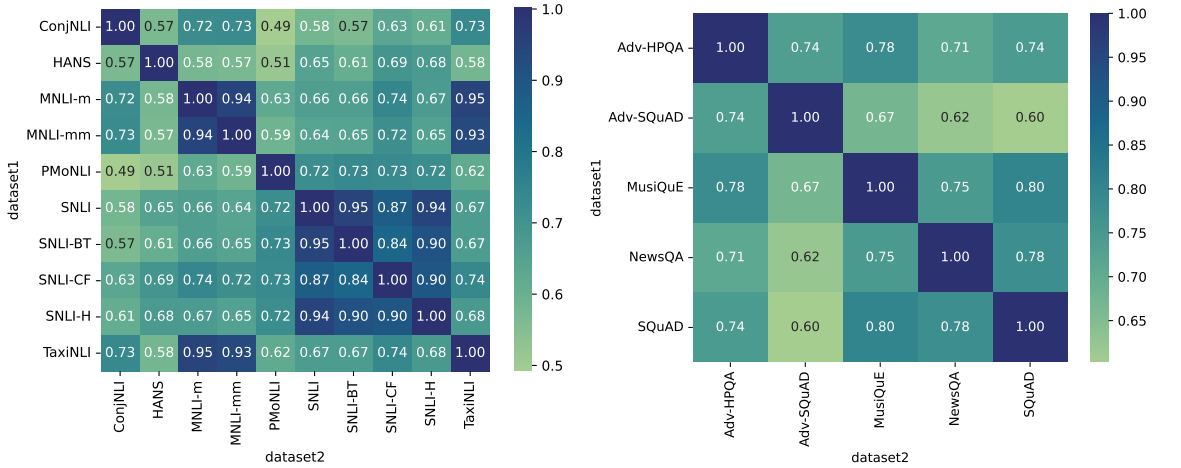
(a) NLI Datasets: Spearman's ρ

(b) MRC Datasets: Spearman's ρ



(c) NLI Datasets: Pearson's r

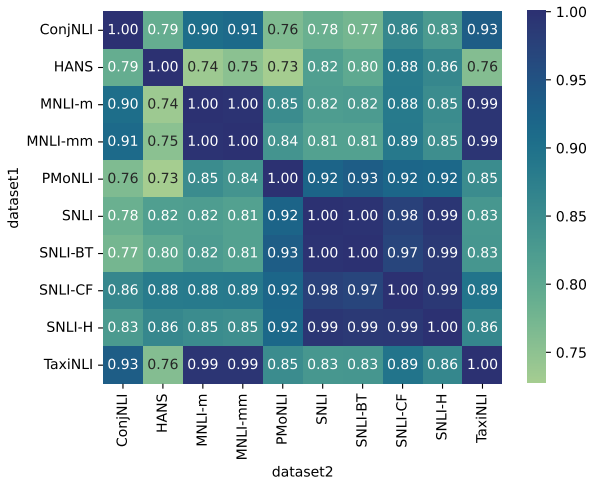
(d) MRC Datasets: Pearson's r



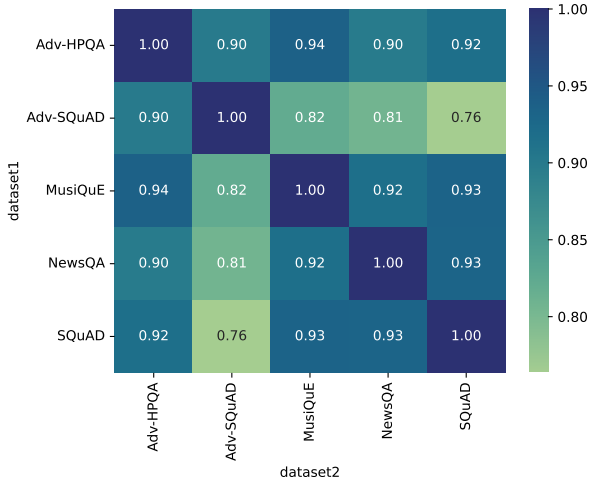
(e) NLI Datasets: Kendall's τ

(f) MRC Datasets: Kendall's τ

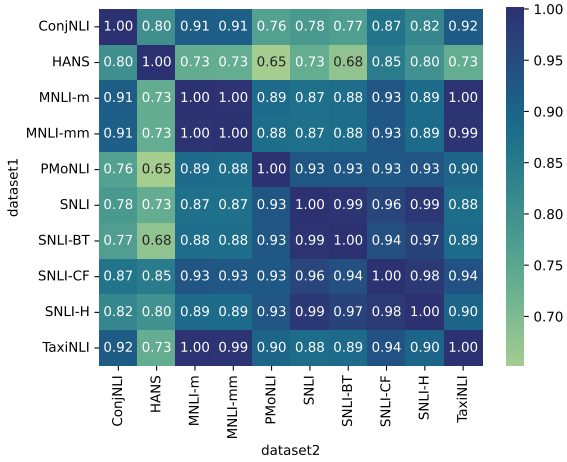
Figure 10.6: Correlation between the source and the target datasets for NLI and MRC on a per-instance basis for different kinds of correlation.



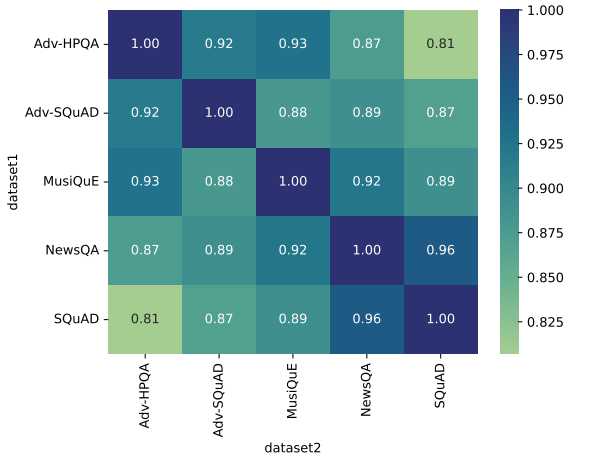
(a) NLI Datasets: Spearman's ρ



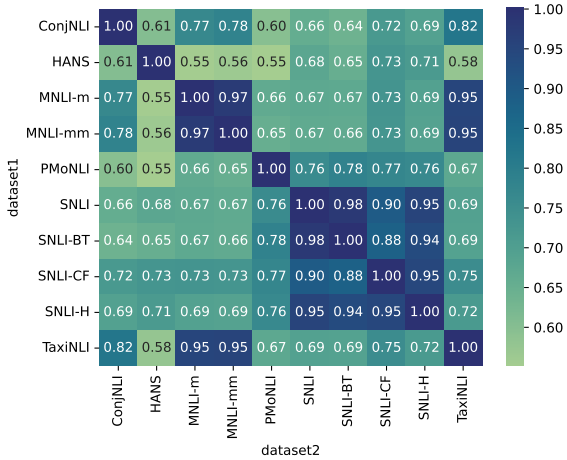
(b) MRC Datasets: Spearman's ρ



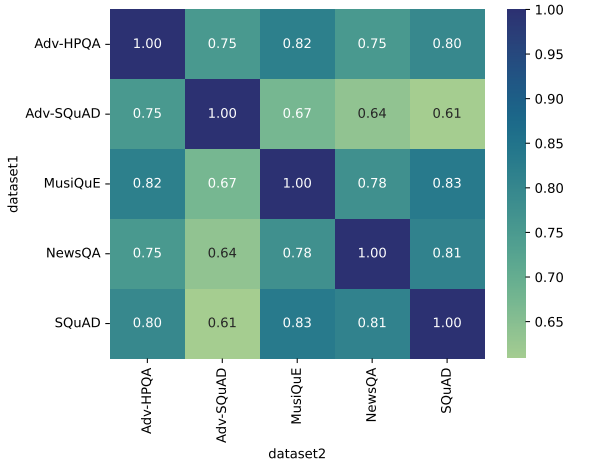
(c) NLI Datasets: Pearson's r



(d) MRC Datasets: Pearson's r



(e) NLI Datasets: Kendall's τ



(f) MRC Datasets: Kendall's τ

Figure 10.7: Correlation between the source and the target datasets for NLI and MRC on a per-architecture basis for different kinds of correlation.

Chapter 11

Conclusion

11.1 Contributions

At the outset, we put forward the central contributions of our thesis.

- We formalize scaffolding as a general approach to facilitate targeted or task-aware generalization in NLP. Rather than treating scaffolds as ad-hoc prompts or auxiliary features, our thesis frames them as task-dependent interventions that introduces the necessary inductive bias to address the challenges arising from a shift in data distribution.
- We propose a taxonomy of categorizing scaffolds into formal and informal scaffolds, and showcase how they complement one another. Formal scaffolds that comprise graphs, reasoning patterns, and linguistic frameworks are best suited to tasks that require factual information that can be grounded in a pre-existing structure. Informal scaffolds in the form of free-text rationales are best suited to tasks that require pragmatic and social understanding. A core takeaway is that these two scaffolds accomplish complementary objectives.
- We emphasize taking a broader view of generalization in the context of NLP tasks. The thesis demonstrates that "generalization" encompasses multiple dimensions such as robustness, compositionality, and domain adaptation among others. We thus need to be cognizant of the phrase "What does it mean to generalize well?" and advocate for a more holistic and systematic evaluation of generalization capabilities. Likewise, we argue that reliance on individual leaderboards (such as GrailQA for KBQA) can paint a deceptive view of progress. We thus motivate the need to design more comprehensive and challenging benchmarks (such as GrailQA++) and stricter evaluation protocols that are task-aware in nature.
- We highlight how structured sources of knowledge can be useful for generalization and diagnosis. Across knowledge-intensive settings, the thesis demonstrates that making the structure explicit enables both finer-grained analysis of failure (e.g the use of isomorphisms as diagnostic tools for KBQA) and more reliable transfer under distribution shift (e.g. the role of linguistic frameworks in cross-domain extraction, or the role of graph structure and path information in PERKGQA).
- We introduce new NLP tasks throughout the thesis such as (i) isomorphism prediction to categorize the kind of reasoning path one needs to traverse to reach the answer in KBQA, (ii)

PERKGQA or question answering over personalized knowledge graphs, and (iii) identifying strategies to foil or resist persuasion in conversations. Each of these tasks were realized by different datasets, that we make public as an additional contribution.

- We design modeling frameworks such as SOCIAL SCAFFOLDS inspired by socio-linguistic theories or narrative modeling principles to generate rationales that can capture multiple perspectives. These rationales improve social understanding by making latent assumptions (in the form of user intents, presuppositions, or hearer reactions) explicit in dialogues.
- We conduct several experiments that highlight the effectiveness of rationales in facilitating transfer for different social meaning detection tasks. Rationales bring about significant gains especially in low data regimes both across domains and across tasks.

11.2 Future Work and Directions

As this dissertation has illustrated, generalization spans multiple facets in NLP such as adapting to new domains and tasks to avoiding spurious "shortcuts". While we lay the foundation of targeted/task-aware generalization through formal and informal scaffolds, several promising avenues remain open for exploration. In this chapter, we outline two forward-looking directions for future research namely, (1) personalization as a more directed form of generalization, and (2) understanding the best way to integrate structured knowledge more effectively into language models. Each of these directions builds upon the themes and findings of this thesis and points toward impactful extensions that address current limitations in NLP systems.

11.2.1 Personalization as an extension of Generalization

One compelling future direction is to move beyond one-size-fits-all LLMs and toward personalized NLP systems. Personalization can be viewed as a form of targeted generalization: instead of generalizing uniformly across a broad population, we aim for models that adapt to individual users and their unique contexts. By tailoring a system's behavior to a user's persona, preferences, and prior history, we hypothesize that the model will be more attuned to that user's needs. This perspective shifts the focus from global performance (averaged over many users) to local effectiveness (optimizing for each user), and it aligns with the idea of scaffolding, where a user's personal context serves as a key scaffold guiding the model's behavior.

Realizing personalized generalization will require future research into persona-driven language models and long-term user modeling. LLMs augmented with explicit user personas or profiles could maintain persistent memory of a user's interactions, allowing them to recall facts or preferences from past sessions. Recent work on LLM-based agents with contextual user memory offers a glimpse of this potential. For instance, systems like AUGUSTUS (Jain et al., 2025) maintain a graph-structured memory of each user's dialogue history and relevant multimodal data, enabling concept-driven retrieval of information to produce personalized responses. Such an approach treats a user's past conversations, documents, or telemetry data (e.g. application usage patterns) as part of the model's context. By retrieving and leveraging this personal context at inference time, an LLM can specialize its responses to the individual.

Moving to personalized models also opens new avenues in domains like healthcare and productivity. Consider a medical QA system that adapts to a patient's health records and demographics: by incorporating those personal details, the model can generalize medical knowledge to recommendations suitable for that specific patient (e.g. considering their allergies, prior conditions, lifestyle). This is especially useful for healthcare settings where even the same symptoms could lead to different outcomes based on the individual's demographics (Logé et al.; Bardhan et al., 2024).

Similarly, an AI-assisted writing tool might learn a particular author's style and context (via emails, drafts, or editing telemetry) and thus provide recommendations or suggestions in a way that fits the specific user's style and tone. In each case, personalization acts as a means to improve generalization performance on user-specific tasks or distributions that differ from the population average. A valid example is our PERKGQA setting where each user query comes with its own knowledge graph, and the QA system has to answer questions over this unseen personal knowledge graph during inference. This effectively treated each user's data as a new domain, demonstrating that it is feasible to generalize to new users by leveraging structured personal context. Future systems can extend this idea by maintaining richer user models (beyond a static knowledge graph) and dynamically updating them as the user interacts with the AI.

To achieve robust personalization, several research challenges must be addressed. One is data privacy and security: personalized models will by nature handle sensitive user-specific data, so methods like federated learning or on-device adaptation may be needed to avoid centralized storage of personal information (Khan et al., 2023a; Jouini et al., 2024). Another challenge is preventing overfitting to a user's idiosyncrasies. A personalized model should still retain general language understanding and only adjust where appropriate. Techniques for continual learning or meta-learning could be employed so that the model learns how to learn from individual users efficiently. Finally, evaluation metrics for personalized generalization need to be developed. Instead of average accuracy on a shared benchmark, we might measure success by improvements in each user's satisfaction or task success rate, emphasizing **personal-level generalization**, similar to the role of isomorphisms in KBQA. Overall, personalization represents a promising means to harness generalization: by training and deploying models that generalize to individual rather than hypothetical "average" users, NLP systems can become more useful and aligned with diverse human needs.

11.2.2 Integrating structured knowledge in LLMs

A recurring theme in this thesis has been the use of structured knowledge such as linguistic parses and knowledge graphs to improve generalization. Our results in earlier chapters demonstrate that incorporating structured information is often beneficial than relying solely on the plain text. For example, augmenting text with syntactic dependency trees or abstract meaning representations led to substantial gains when generalizing information extraction systems across different kinds of procedural texts. In the realm of KBQA, we showcased how leveraging the schema of a knowledge graph (its ontology of entities, relations, and classes) enables a model to handle unseen entities or even new relation types at inference time. These successes underscore a broader point: structured knowledge provides a powerful inductive bias for NLP models, and future research should explore the best ways to represent and inject such knowledge into language models on a

wider scale.

There exists several hybrid modeling techniques that combine LLMs with structured knowledge modules or representations. The two most common approaches include: (i) prompting or serializing structured data as additional text input to an LLM and (ii) architectural modifications where the external knowledge is encoded by specialized neural networks and fused with the representations obtained from text. Both strategies have their strengths and weaknesses. Prompt-based methods (for example, linearizing a knowledge graph's triples into text sentences) are simple to implement and do not require changing the underlying LLM, but they often lose the fidelity of relational structure in the conversion to text (Coppolillo, 2025). On the other hand, architectural models that explicitly combine text and other modalities has empirically shown higher performance on a wide variety of tasks (Li et al., 2025) but require training before deployment and cannot easily handle OOD data shifts.

Thus a potential avenue of research is identifying the efficiency and cost trade-offs of injecting structured knowledge into LMs. There may not be a one-size-fits-all answer, as the optimal method could depend on the type of structured data and the application. Knowledge graphs, for example, encode factual relationships that might best be utilized through a dedicated retrieval or embedding mechanism. A forward-looking idea is to use pre-trained knowledge graph embeddings or encoding models to supply an LLM with vector representations of relevant entities and relations at inference time. For example the work of Coppolillo (2025) exemplifies this by generating graph-aware tokens: vectors from a knowledge graph embedding model are transformed into special token embeddings that are inserted into the LLM's input, thereby providing structured context without fine-tuning the LLM itself. This approach was found to improve reasoning accuracy on QA tasks while remaining resource-efficient, since the LLM remains frozen and only the smaller KG encoder is trained.

Another potential research direction for in-context learning or prompt based methods is the manner in which structured representation is provided as input to the models. While the common trend is to represent the node and edge information as linearized tuples (e_1, r, e_2), alternative ways of representing the data such as via structured formats like YAML/JSON (`'head': 'e_1', 'relation': 'rel', 'tail': 'e_2'`) or in natural language text (e_1 is connected to e_2 via relation 'r') could be useful in certain scenarios, since certain forms of representation are more expressive than others. This line of research stems from our findings in Dutt et al. (2025) where we observed the manner in which dependency parses are added to multi-lingual language models impact performance significantly.

Finally, it is not just sufficient to improve performance by injecting structured knowledge but also investigating how the integration shapes the way the representations end up interacting with one another. Recent work has highlighted the effectiveness of contrastive co-distillation approach as a technique to attempt to bring the graph and text representations into the shared same space and how their interplay during the training dynamics can inform whether the representations undergo alignment or separation (Wu et al., 2025). A promising line of research is to extend beyond these text and graph modalities and observe how representations arising from different modalities interact with each other.

.

Bibliography

Robert P Abelson and James C Miller. 1967. Negative persuasion via personal insult. *Journal of Experimental Social Psychology*, 3(4):321–333. 7.3.1

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*. 1.2, 8.1.3

Rohini Ahluwalia. 2000. Examination of psychological processes underlying resistance to persuasion. *Journal of Consumer Research*, 27(2):217–232. 7.3.1

AI@Meta. 2024. Llama 3 model card. 9.2.3, 2.3

Alon Albalak, Yi-Lin Tuan, Pegah Jandaghi, Connor Pryor, Luke Yoffe, Deepak Ramachandran, Lise Getoor, Jay Pujara, and William Yang Wang. 2022. FETA: A benchmark for few-sample task transfer in open-domain dialogue. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10936–10953, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. 2.3, 8.2.2

Amal Alqahtani, Efsun Sarioglu Kayi, Sardar Hamidian, Michael Compton, and Mona Diab. 2022. A quantitative and qualitative analysis of schizophrenia language. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, pages 173–183. 2.2.1, 7.2

Ghulam Ahmed Ansari, Amrita Saha, Vishwajeet Kumar, Mohan Bhambhani, Karthik Sankaranarayanan, and Soumen Chakrabarti. 2019. Neural program induction for kbqa without gold programs or query annotations. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4890–4896. International Joint Conferences on Artificial Intelligence Organization. 2.1.2

Jean-michel Attendu and Jean-philippe Corbeil. 2023. NLU on data diets: Dynamic data subset selection for NLP classification tasks. In *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, pages 129–146, Toronto, Canada (Hybrid). Association for Computational Linguistics. 2.3

Xuefeng Bai, Yulong Chen, Linfeng Song, and Yue Zhang. 2021. Semantic representation for dialogue modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4430–4445, Online. Association for Computational Linguistics. 2.1.1, 4.1.1

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching

the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics. 2.1.1

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186. 3.2.1, 4.1, 4.1.1

Jayetri Bardhan, Kirk Roberts, and Daisy Zhe Wang. 2024. Question answering for electronic health records: Scoping review of datasets and models. *Journal of Medical Internet Research*, 26:e53636. 11.2.1

Larry M Bartels. 2006. Priming and persuasion in presidential campaigns. *Capturing campaign effects*, 1:78–114. 7.3.1

Elisa Bassignana, Filip Ginter, Sampo Pyysalo, Rob van der Goot, and Barbara Plank. 2023. Silver syntax pre-training for cross-domain relation extraction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6984–6993, Toronto, Canada. Association for Computational Linguistics. 2.1

Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967, Copenhagen, Denmark. Association for Computational Linguistics. 2.1

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005, Jeju Island, Korea. Association for Computational Linguistics. 5.2.3, 8.2.2, 9.2.3, 10.2.2

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. In *Proceedings of AAAI*. 4.1.1

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. 4.1.1

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128, Sydney, Australia. Association for Computational Linguistics. 2.3

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270. 2.1

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. 2.1

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko.

2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26. 3.3.1, 5.2.1, 5.2.2

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics. 10.1.1

Luke Breitfeller, Emily Ahn, David Jurgens, and Yulia Tsvetkov. 2019. Finding microaggressions in the wild: A case for locating elusive phenomena in social media posts. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1664–1674, Hong Kong, China. Association for Computational Linguistics. 2.2.1

Samuel Broscheit, Daniel Ruffinelli, Adrian Kochsiek, Patrick Betz, and Rainer Gemulla. 2020. Libkge-a knowledge graph embedding library for reproducible research. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 165–174. 5.2.2

Penelope Brown, Stephen C Levinson, and Stephen C Levinson. 1987. *Politeness: Some universals in language usage*, volume 4. Cambridge university press. 2.2.2

Jan Buchmann, Max Eichler, Jan-Micha Bodensohn, Ilia Kuznetsov, and Iryna Gurevych. 2024. Document structure in long document transformers. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1056–1073, St. Julian's, Malta. Association for Computational Linguistics. 2.1

Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y. Hammerla. 2019. Relational graph attention networks. 2

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42:335–359. (document), 2.2.1, 7.3.2, 7.2

Rui Cai and Mirella Lapata. 2019. Syntax-aware semantic role labeling without parsing. *Transactions of the Association for Computational Linguistics*, 7:343–356. 2.1

Agostina Calabrese, Leonardo Neves, Neil Shah, Maarten Bos, Björn Ross, Mirella Lapata, and Francesco Barbieri. 2024. Explainability and hate speech: Structured explanations make social media moderators faster. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 398–408, Bangkok, Thailand. Association for Computational Linguistics. 7.5

Nitay Calderon, Naveh Porat, Eyal Ben-David, Zorik Gekhman, Nadav Oved, and Roi Reichart. 2023. Measuring the robustness of natural language processing models to domain shifts. *arXiv preprint arXiv:2306.00168*. 2.3, 10.2.4

Erik Cambria, Qian Liu, Sergio Decherchi, Frank Xing, and Kenneth Kwok. 2022. SenticNet 7: A commonsense-based neurosymbolic AI framework for explainable sentiment analysis. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages

3829–3839, Marseille, France. European Language Resources Association. 2.1

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 152–161, Melbourne, Australia. Association for Computational Linguistics. 2.2

Kushal Chawla, Jaysa Ramirez, Rene Clever, Gale Lucas, Jonathan May, and Jonathan Gratch. 2021. CaSiNo: A corpus of campsite negotiation dialogues for automatic negotiation systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3167–3185, Online. Association for Computational Linguistics. (document), 2.2.1, 2, 7.4, **??**, 9.2.1, **??**

Chih-Yao Chen and Cheng-Te Li. 2021. ZS-BERT: Towards zero-shot relation extraction with attribute representation learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3470–3479, Online. Association for Computational Linguistics. 2.1.1

Hanjie Chen, Faeze Brahman, Xiang Ren, Yangfeng Ji, Yejin Choi, and Swabha Swayamdipta. 2023. REV: Information-theoretic evaluation of free-text rationales. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2007–2030, Toronto, Canada. Association for Computational Linguistics. 2.3

Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. 2017. Improved neural machine translation with a syntax-aware encoder and decoder. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1936–1945, Vancouver, Canada. Association for Computational Linguistics. 2.1

Mingda Chen, Zewei Chu, Karl Stratos, and Kevin Gimpel. 2020. Mining knowledge for natural language inference from Wikipedia categories. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3500–3511, Online. Association for Computational Linguistics. 2.2

Pei Chen, Soumajyoti Sarkar, Leonard Lausen, Balasubramaniam Srinivasan, Sheng Zha, Ruihong Huang, and George Karypis. 2024. Hytrel: Hypergraph-enhanced tabular data representation learning. *Advances in Neural Information Processing Systems*, 36. 1.2

Qianglong Chen, Feng Ji, Xiangji Zeng, Feng-Lin Li, Ji Zhang, Haiqing Chen, and Yin Zhang. 2021a. KACE: Generating knowledge aware contrastive explanations for natural language inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2516–2527, Online. Association for Computational Linguistics. 2.2

Shuang Chen, Qian Liu, Zhiwei Yu, Chin-Yew Lin, Jian-Guang Lou, and Feng Jiang. 2021b. Retrack: a flexible and efficient framework for knowledge base question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 325–336. 2.1.2

Tianyu Chen, Shaohan Huang, Furu Wei, and Jianxin Li. 2021c. Pseudo-label guided unsupervised

domain adaptation of contextual embeddings. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 9–15, Kyiv, Ukraine. Association for Computational Linguistics. 2.3

Xiang Chen, Yue Cao, and Xiaojun Wan. 2021d. WIND: Weighting instances differentially for model-agnostic domain adaptation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2366–2376, Online. Association for Computational Linguistics. 2.3

Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758. 6.2.2, 6.2.2

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561, Austin, Texas. Association for Computational Linguistics. 2.1

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*. 1.2, 9.2.3

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3586–3596, Minneapolis, Minnesota. Association for Computational Linguistics. 1.3

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics. 1

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Unsupervised cross-lingual representation learning at scale. 4.2.1

John W Cooley. 1993. A classical approach to mediation-part i: Classical rhetoric and the art of persuasion in mediation. *U. Dayton L. Rev.*, 19:83. 7.3.1

Erica Coppolillo. 2025. Injecting knowledge graphs into large language models. *arXiv preprint arXiv:2505.07554*. 11.2.2

William Croft. 2022. Morphosyntax: constructions of the world's languages. 2.1

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5636–5646, Hong Kong, China. Association for Computational Linguistics. 7.3.1

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg. 10.1

Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 250–259. 2.2.2

Devleena Das and Sonia Chernova. 2020. Leveraging rationales to improve human task performance. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pages 510–518. 2.2

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2018. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *International Conference on Learning Representations*. 2.1.2

Rajarshi Das, Ameya Godbole, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2020. Non-parametric reasoning in knowledge bases. In *Automated Knowledge Base Construction*. 2.1.3, 5.1.1

Rajarshi Das, Ameya Godbole, Ankita Naik, Elliot Tower, Manzil Zaheer, Hannaneh Hajishirzi, Robin Jia, and Andrew Mccallum. 2022. Knowledge base question answering by case-based reasoning over subgraphs. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4777–4793. PMLR. 3.3.2

Rajarshi Das, Manzil Zaheer, Dung Thai, Ameya Godbole, Ethan Perez, Jay Yoon Lee, Lizhen Tan, Lazaros Polymenakos, and Andrew McCallum. 2021a. Case-based reasoning for natural language queries over knowledge bases. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9594–9611, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 2.1

Rajarshi Das, Manzil Zaheer, Dung Thai, Ameya Godbole, Ethan Perez, Jay-Yoon Lee, Lizhen Tan, Lazaros Polymenakos, and Andrew McCallum. 2021b. Case-based reasoning for natural language queries over knowledge bases. 2.1.3

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic. Association for Computational Linguistics. 2.3

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36. 2.3, 9.2.3, 2.3

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. 4.2.1

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. 1.2, 2.3, 10.1.3

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. 4.1.1, 8.2.2

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

Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, Daniel Maturana, Martin Thoma, Eric Battenberg, Jack Kelly, Jeffrey De Fauw, Michael Heilman, Diogo Moitinho de Almeida, Brian McFee, Hendrik Weideman, Gábor Takács, Peter de Rivaz, Jon Crall, Gregory Sanders, Kashif Rasul, Cong Liu, Geoffrey French, and Jonas Degrave. 2015. Lasagne: First release. 2.1.2

Jiwei Ding, Wei Hu, Qixin Xu, and Yuzhong Qu. 2019. Leveraging frequent query substructures to generate formal queries for complex question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2614–2622, Hong Kong, China. Association for Computational Linguistics. 2.1.2

Yijiang Dong, Lara Martin, and Chris Callison-Burch. 2023. CoRRPUS: Code-based structured prompting for neurosymbolic story understanding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13152–13168, Toronto, Canada. Association for Computational Linguistics. 2.1

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics. 5.2.3, 8.2.2, 9.2.3, 10.2.2

Wenyu Du, Zhouhan Lin, Yikang Shen, Timothy J. O'Donnell, Yoshua Bengio, and Yue Zhang. 2020. Exploiting syntactic structure for better language modeling: A syntactic distance approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6611–6628, Online. Association for Computational Linguistics. 2.1

Zhichao Duan, Xiuxing Li, Zhenyu Li, Zhuo Wang, and Jianyong Wang. 2022. Not just plain text! fuel document-level relation extraction with explicit syntax refinement and subsentence modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages

1941–1951, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. 2.1

Ritam Dutt, Kasturi Bhattacharjee, Rashmi Gangadharaiah, Dan Roth, and Carolyn Rose. 2022. Perkgqa: Question answering over personalized knowledge graphs. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 253–268. 3.3

Ritam Dutt, Rishabh Joshi, and Carolyn Rose. 2020a. Keeping up appearances: Computational modeling of face acts in persuasion oriented discussions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7473–7485, Online. Association for Computational Linguistics. 2.2.2

Ritam Dutt, Rishabh Joshi, and Carolyn Penstein Rosé. 2020b. Keeping up appearances: Computational modeling of face acts in persuasion oriented discussions. *arXiv preprint arXiv:2009.10815*. 2.2.1

Ritam Dutt, Sopan Khosla, Vinayshekhar Bannihatti Kumar, and Rashmi Gangadharaiah. 2023. GrailQA++: A challenging zero-shot benchmark for knowledge base question answering. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 897–909, Nusa Dua, Bali. Association for Computational Linguistics. 3.3

Ritam Dutt, Sayan Sinha, Rishabh Joshi, Surya Shekhar Chakraborty, Meredith Riggs, Xinru Yan, Haogang Bao, and Carolyn Rose. 2021. ResPer: Computationally modelling resisting strategies in persuasive conversations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 78–90, Online. Association for Computational Linguistics. (document), 2.2.1, 7.3, 7.1, 3, **??**, 9.2.1, **??**

Ritam Dutt, Shounak Sural, and Carolyn Rose. 2025. Can dependency parses facilitate generalization in language models? a case study of cross-lingual relation extraction. In *Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing*, pages 317–337, Albuquerque, New Mexico, USA. Association for Computational Linguistics. 11.2.2

Ritam Dutt, Zhen Wu, Jiaxin Shi, Divyanshu Sheth, Prakhar Gupta, and Carolyn Rose. 2024. Leveraging machine-generated rationales to facilitate social meaning detection in conversations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6901–6929, Bangkok, Thailand. Association for Computational Linguistics. 2.2.2, 7.5, 9.1.1, 9.1.2, 9.1.3

Cynthia Dwork, Ravi Kumar, Moni Naor, and Dandapani Sivakumar. 2001. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, pages 613–622. 10.2.1

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics. 2.3

Santiago Egea Gómez, Euan McGill, and Horacio Saggion. 2021. Syntax-aware transformers for neural machine translation: The case of text to sign gloss translation. In *Proceedings of*

the 14th Workshop on Building and Using Comparable Corpora (BUCC 2021), pages 18–27, Online (Virtual Mode). INCOMA Ltd. 2.1

Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. 2021. Latent hatred: A benchmark for understanding implicit hate speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 345–363, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. (document), 2.2.1, 6, 7.7, **??**, 9.2.1, **??**

James N Farr, James J Jenkins, and Donald G Paterson. 1951. Simplification of flesch reading ease formula. *Journal of applied psychology*, 35(5):333. 9.3.3, 2.3

Shangbin Feng, Vidhisha Balachandran, Yuyang Bai, and Yulia Tsvetkov. 2023. FactKB: Generalizable factuality evaluation using language models enhanced with factual knowledge. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 933–952, Singapore. Association for Computational Linguistics. 2.1

Song Feng, Siva Sankalp Patel, Hui Wan, and Sachindra Joshi. 2021. MultiDoc2Dial: Modeling dialogues grounded in multiple documents. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6162–6176, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 2.2

Matthias Fey and Jan E. Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*. 5.2.2

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the Abstract Meaning Representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland. Association for Computational Linguistics. 4.1.1

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378. 7.3.1

Maxwell Forbes, Jena D. Hwang, Vered Shwartz, Maarten Sap, and Yejin Choi. 2020. Social chemistry 101: Learning to reason about social and moral norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 653–670, Online. Association for Computational Linguistics. 2.1

Marieke L Fransen, Edith G Smit, and Peeter WJ Verlegh. 2015a. Strategies and motives for resistance to persuasion: an integrative framework. *Frontiers in psychology*, 6:1201. 7.3.1

Marieke L Fransen, Peeter WJ Verlegh, Amna Kirmani, and Edith G Smit. 2015b. A typology of consumer strategies for resisting advertising, and a review of mechanisms for countering them. *International Journal of Advertising*, 34(1):6–16. 7.3.1

Timo Freiesleben and Thomas Grote. 2023. Beyond generalization: a theory of robustness in machine learning. *Synthese*, 202(4):109. 1.1

James Paul Gee. 2014. *An introduction to discourse analysis: Theory and method*. routledge. 2.2.1, 7.2

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference

models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online. Association for Computational Linguistics. 10.1.1

Deepanway Ghosal, Somak Aditya, and Monojit Choudhury. 2023. Prover: Generating intermediate steps for NLI with commonsense knowledge retrieval and next-step prediction. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 872–884, Nusa Dua, Bali. Association for Computational Linguistics. 2.2

Deepanway Ghosal, Navonil Majumder, Alexander Gelbukh, Rada Mihalcea, and Soujanya Poria. 2020. COSMIC: COmmonSense knowledge for eMotion identification in conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2470–2481, Online. Association for Computational Linguistics. 7.3.2

Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. 2019. DialogueGCN: A graph convolutional neural network for emotion recognition in conversation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 154–164, Hong Kong, China. Association for Computational Linguistics. 2.1

Erving Goffman. 2002. Front and back regions of everyday life [1959]. *The everyday life reader*, pages 50–57. 2.2.1, 7.2, 8.1.1

Shreya Goyal, Sumanth Doddapaneni, Mitesh M. Khapra, and Balaraman Ravindran. 2023. A survey of adversarial defenses and robustness in nlp. *ACM Comput. Surv.*, 55(14s). 2.3

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal

Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan,

Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. 4.2.2, 4.2.3

Ralph Grishman. 1996. The role of syntax in information extraction. In *TIPSTER TEXT PROGRAM PHASE II: Proceedings of a Workshop held at Vienna, Virginia, May 6-8, 1996*, pages 139–142, Vienna, Virginia, USA. Association for Computational Linguistics. 2.1

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. 5.1.2

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. Beyond iid: three levels of generalization for question answering on knowledge bases. In *Proceedings of the Web Conference 2021*, pages 3477–3488. (document), 2.1.2, 2.1.3, 3.3, 3.3.2, 3.7, 3.3.2, 3.3.2, 3.3.2, 3.3.2, 3.3.2

Yu Gu, Vardaan Pahuja, Gong Cheng, and Yu Su. 2022a. Knowledge base question answering: A semantic parsing perspective. *arXiv preprint arXiv:2209.04994*. 2.1.3, 6.1.1

Yu Gu and Yu Su. 2022. Arcaneqa: Dynamic program induction and contextualized encoding for knowledge base question answering. *arXiv preprint arXiv:2204.08109*. 3.3.2, 6.1.1, 6.1.2

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022b. PPT: Pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8410–8423, Dublin, Ireland. Association for Computational Linguistics. 2.3

Yong Guan, Shaoru Guo, Ru Li, Xiaoli Li, and Hongye Tan. 2021. Frame semantic-enhanced sentence modeling for sentence-level extractive text summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4045–4052, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 2.1

Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey Bigham. 2022. InstructDial: Improving zero and few-shot generalization in dialogue through instruction tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 505–525, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. 2.2.2

Sireesh Gururaja, Ritam Dutt, Tinglong Liao, and Carolyn Rosé. 2023a. Linguistic representations for fewer-shot relation extraction across domains. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7502–7514, Toronto, Canada. Association for Computational Linguistics. 1.3

Sireesh Gururaja, Ritam Dutt, Tinglong Liao, and Carolyn Rose. 2023b. Linguistic representations for fewer-shot relation extraction across domains. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7502–7514. 2.1, 8.3

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics. 2.3

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112. 10.1.1

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. WARP: Word-level Adversarial ReProgramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4921–4933, Online. Association for Computational Linguistics. 2.3

Jiale Han, Bo Cheng, and Xu Wang. 2020. Open domain question answering based on text enhanced knowledge graph with hyperedge infusion. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1475–1481, Online. Association for Computational

Linguistics. 2.1

Boran Hao, Henghui Zhu, and Ioannis Paschalidis. 2020. Enhancing clinical BERT embedding using a biomedical knowledge base. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 657–661, Barcelona, Spain (Online). International Committee on Computational Linguistics. 2.1

Taher H Haveliwala. 2003. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE transactions on knowledge and data engineering*, 15(4):784–796. 2.1.2, 5.2.1

Devamanyu Hazarika, Soujanya Poria, Rada Mihalcea, Erik Cambria, and Roger Zimmermann. 2018. Icon: Interactive conversational memory network for multimodal emotion detection. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2594–2604. 2.2.1, 7.2

Devamanyu Hazarika, Soujanya Poria, Roger Zimmermann, and Rada Mihalcea. 2021. Conversational transfer learning for emotion recognition. *Information Fusion*, 65:1–12. 2.2.2

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018a. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343, Brussels, Belgium. Association for Computational Linguistics. (document), 7.1

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018b. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343, Brussels, Belgium. Association for Computational Linguistics. 2.2.1

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018c. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343. 7.3.1

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics. 10.1

Teresa Heath, Robert Cluley, and Lisa O'Malley. 2017. Beating, ditching and hiding: consumers' everyday resistance to marketing. *Journal of Marketing Management*, 33(15-16):1281–1303. 7.3.1

Manoel Horta Ribeiro, Justin Cheng, and Robert West. 2023. Automated content moderation increases adherence to community guidelines. In *Proceedings of the ACM web conference 2023*, pages 2666–2676. 7.5

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems*, 33:20179–20191. 2.2.2

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019a. Parameter-efficient transfer

learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR. 2.3

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019b. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR. 10.1.3

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics. 10.1.3

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017. 2.2

Chao-Chun Hsu, Sheng-Yeh Chen, Chuan-Chun Kuo, Ting-Hao Huang, and Lun-Wei Ku. 2018. EmotionLines: An emotion corpus of multi-party conversations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA). 7.3.2

Chao-Chun Hsu and Lun-Wei Ku. 2018. SocialNLP 2018 EmotionX challenge overview: Recognizing emotions in dialogues. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 27–31, Melbourne, Australia. Association for Computational Linguistics. 7.3.2

I-Hung Hsu, Zhiyu Xie, Kuan-Hao Huang, Prem Natarajan, and Nanyun Peng. 2023. AMPERE: AMR-aware prefix for generation-based event argument extraction model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10976–10993, Toronto, Canada. Association for Computational Linguistics. 1.2, 2.1

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021a. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*. 2.3

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021b. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*. 9.2.3, 10.1.3, 2.3

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore. Association for Computational Linguistics. 2.2

Xiaofeng Huang, Jixin Zhang, Zisang Xu, Lu Ou, and Jianbin Tong. 2021. A knowledge graph based question answering method for medical domain. *PeerJ Computer Science*, 7:e667. 2.1.2

Pere-Lluís Huguet Cabot, Simone Tedeschi, Axel-Cyrille Ngonga Ngomo, and Roberto Navigli.

2023. RED^{fm}: a filtered and multilingual relation extraction dataset. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4326–4343, Toronto, Canada. Association for Computational Linguistics. 3.2.2

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, et al. 2023. A taxonomy and review of generalization research in nlp. *Nature Machine Intelligence*, 5(10):1161–1174. (document), 1.1, 1.1, 1.3, 2.3, 10, 10.1

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics. 2.2

Jitesh Jain, Shubham Maheshwari, Ning Yu, Wen-mei Hwu, and Humphrey Shi. 2025. Augustus: An llm-driven multimodal agent system with contextualized user memory. *arXiv preprint arXiv:2510.15261*. 11.2.1

Parag Jain and Mirella Lapata. 2021. Memory-based semantic parsing. *Transactions of the Association for Computational Linguistics*, 9:1197–1212. 2.1

Sahil Jayaram and Emily Allaway. 2021. Human rationales as attribution priors for explainable stance detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5540–5554, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 2.2

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics. 10.1.2

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. 4.2.2, 4.2.3

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271, Prague, Czech Republic. Association for Computational Linguistics. 2.3

Longquan Jiang and Ricardo Usbeck. 2022. Knowledge graph question answering datasets and their generalizability: Are they enough for future research? *arXiv preprint arXiv:2205.06573*. 3.3.2

Yichen Jiang and Mohit Bansal. 2019. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2726–2736, Florence, Italy. Association for Computational Linguistics. 10.1.2

Yiwei Jiang, Klim Zaporojets, Johannes Deleu, Thomas Demeester, and Chris Develder. 2020. Recipe instruction semantics corpus (RISeC): Resolving semantic structure and zero anaphora

in recipes. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 821–826, Suzhou, China. Association for Computational Linguistics. 3.2.1

Wenxiang Jiao, Michael Lyu, and Irwin King. 2020. Real-time emotion recognition via attention gated hierarchical memory network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8002–8009. 2.1

Wenxiang Jiao, Haiqin Yang, Irwin King, and Michael R. Lyu. 2019. HiGRU: Hierarchical gated recurrent units for utterance-level emotion recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 397–406, Minneapolis, Minnesota. Association for Computational Linguistics. 2.1

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press. 2.3

Woojeong Jin, Yu Cheng, Yelong Shen, Weizhu Chen, and Xiang Ren. 2022. A good prompt is worth millions of parameters: Low-resource prompt-based learning for vision-language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2763–2775, Dublin, Ireland. Association for Computational Linguistics. 2.3

Yohan Jo, Elijah Mayfield, Chris Reed, and Eduard Hovy. 2020. Machine-aided annotation for fine-grained proposition types in argumentation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1008–1018, Marseille, France. European Language Resources Association. (document), 2.2.1, 5, 7.6, **??**, 9.2.1, **??**

Brihi Joshi, Aaron Chan, Ziyi Liu, Shaoliang Nie, Maziar Sanjabi, Hamed Firooz, and Xiang Ren. 2022. ER-test: Evaluating explanation regularization methods for language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3315–3336, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. 2.2

Brihi Joshi, Ziyi Liu, Sahana Ramnath, Aaron Chan, Zhewei Tong, Shaoliang Nie, Qifan Wang, Yejin Choi, and Xiang Ren. 2023. Are machine rationales (not) useful to humans? measuring and improving human utility of free-text rationales. *arXiv preprint arXiv:2305.07095*. 2.2

Pratik Joshi, Somak Aditya, Aalok Sathe, and Monojit Choudhury. 2020. TaxiNLI: Taking a ride up the NLU hill. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 41–55, Online. Association for Computational Linguistics. 10.1.1

Ratnesh Joshi, Arindam Chatterjee, and Asif Ekbal. 2021a. Towards explainable dialogue system: Explaining intent classification using saliency techniques. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 120–127, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI). 7.5

Rishabh Joshi, Vidhisha Balachandran, Shikhar Vashishth, Alan Black, and Yulia Tsvetkov. 2021b.

Dialograph: Incorporating interpretable strategy-graph networks into negotiation dialogues. In *International Conference on Learning Representations*. 7.3.1

Oumayma Jouini, Kaouthar Sethom, Abdallah Namoun, Nasser Aljohani, Meshari Huwaytim Alanazi, and Mohammad N Alanazi. 2024. A survey of machine learning in edge computing: Techniques, frameworks, applications, issues, and research directions. *Technologies*, 12(6):81. 11.2.1

Haein Jung, Heuiyeen Yeen, Jeehyun Lee, Minju Kim, Namo Bang, and Myoung-Wan Koo. 2023. Enhancing task-oriented dialog system with subjective knowledge: A large language model-based data augmentation framework. In *Proceedings of The Eleventh Dialog System Technology Challenge*, pages 150–165, Prague, Czech Republic. Association for Computational Linguistics. 8.1.2

Dan Jurafsky, Rajesh Ranganath, and Dan McFarland. 2009. Extracting social meaning: Identifying interactional style in spoken conversation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 638–646. 2.2.1, 7.2

Jungo Kasai, Dan Friedman, Robert Frank, Dragomir Radev, and Owen Rambow. 2019. Syntax-aware neural semantic role labeling with supertags. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 701–709, Minneapolis, Minnesota. Association for Computational Linguistics. 2.1

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*. 10.1.1

Efsun Sarioglu Kayi, Mona Diab, Luca Pauselli, Michael Compton, and Glen Coppersmith. 2017. Predictive linguistic features of schizophrenia. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, pages 241–250. 2.2.1, 7.2

Pei Ke, Hao Zhou, Yankai Lin, Peng Li, Jie Zhou, Xiaoyan Zhu, and Minlie Huang. 2022. Ctrleval: An unsupervised reference-free metric for evaluating controlled text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2306–2319. 6.1.2

Mashal Khan, Frank G Glavin, and Matthias Nickles. 2023a. Federated learning as a privacy solution-an overview. *Procedia Computer Science*, 217:316–325. 11.2.1

Shakir Khan, Mohd Fazil, Agbotiname Lucky Imoize, Bayan Ibrahimm Alabduallah, Bader M Albahlal, Saad Abdullah Alajlan, Abrar Almjally, and Tamanna Siddiqui. 2023b. Transformer architecture-based transfer learning for politeness prediction in conversation. *Sustainability*, 15(14):10828. 2.2.2

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Nearest neighbor machine translation. In *International Conference on Learning Representations*. 2.2

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. In *International*

Conference on Learning Representations. 1.3, 2.1

Sopan Khosla, Ritam Dutt, Vinayshekhar Bannihatti Kumar, and Rashmi Gangadharaiah. 2023. Exploring the reasons for non-generalizability of kbqa systems. In *The Fourth Workshop on Insights from Negative Results in NLP*, pages 88–93. 3.3.2, 6.1.2

Sopan Khosla and Rashmi Gangadharaiah. 2022. Benchmarking the covariate shift robustness of open-world intent classification approaches. *AACL-IJCNLP 2022*, page 14. 2.2.2

Hyunwoo Kim, Youngjae Yu, Liwei Jiang, Ximing Lu, Daniel Khashabi, Gunhee Kim, Yejin Choi, and Maarten Sap. 2022. ProsocialDialog: A prosocial backbone for conversational agents. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4005–4029, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. 9.1.1

Jaekyeom Kim, Dong-Ki Kim, Lajanugen Logeswaran, Sungryull Sohn, and Honglak Lee. 2024. Auto-intent: Automated intent discovery and self-exploration for large language model web agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16531–16541, Miami, Florida, USA. Association for Computational Linguistics. 7.5

Joongwon Kim, Akari Asai, Gabriel Ilharco, and Hannaneh Hajishirzi. 2023. TaskWeb: Selecting better source tasks for multi-task NLP. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11032–11052, Singapore. Association for Computational Linguistics. 2.3

JP Kincaid. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. *Chief of Naval Technical Training*. 9.3.3, 2.3

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980. 4.1.2

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA). 3.2.1, 3.2.1

Silvia Knobloch-Westerwick and Jingbo Meng. 2009. Looking the other way: Selective exposure to attitude-consistent and counterattitudinal political information. *Communication Research*, 36(3):426–448. 7.3.1

Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2019. Improved document modelling with a neural discourse parser. In *Proceedings of the 17th Annual Workshop of the Australasian Language Technology Association*, pages 67–76, Sydney, Australia. Australasian Language Technology Association. 2.1

Sawan Kumar and Partha Talukdar. 2020. NILE : Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8730–8742, Online. Association for Computational Linguistics. 2.2

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2022.

Complex knowledge base question answering: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(11):11196–11215. (document), 3.5

Yunshi Lan and Jing Jiang. 2020. Query graph generation for answering multi-hop complex questions from knowledge bases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 969–974, Online. Association for Computational Linguistics. 2.1.2, 3.3.2

Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316. 2.2.2

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 2.3

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474. 2.2

Guozheng Li, Peng Wang, and Wenjun Ke. 2023. Revisiting large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6877–6892, Singapore. Association for Computational Linguistics. 2.1.1

Mingchen Li and Jonathan Shihao Ji. 2022. Semantic structure based query graph prediction for question answering over knowledge graph. *arXiv preprint arXiv:2204.10194*. 3.3.2

Mingyang Li, Louis Hickman, Louis Tay, Lyle Ungar, and Sharath Chandra Guntuku. 2020. Studying politeness across cultures using english twitter and mandarin weibo. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2):1–15. 2.2.2

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics. 2.3

Zichao Li, Zong Ke, and Puning Zhao. 2025. Injecting structured knowledge into LLMs via graph neural networks. In *Proceedings of the 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025)*, pages 16–25, Vienna, Austria. Association for Computational Linguistics. 11.2.2

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2018. Multi-hop knowledge graph reasoning with reward shaping. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3243–3253, Brussels, Belgium. Association for Computational Linguistics. 2.1.2

Michael K Lindell, Christina J Brandt, and David J Whitney. 1999. A revised index of interrater agreement for multi-item ratings of a single target. *Applied Psychological Measurement*, 23(2):127–135. 8.1.4, 9.1.3

Trond Linjordet and Krisztian Balog. 2022. Would you ask it that way? measuring and improving question naturalness for knowledge graph question answering. *arXiv preprint arXiv:2205.12768*. 6.1.2

Robert Litschko, Ivan Vulić, Simone Paolo Ponzetto, and Goran Glavaš. 2021. On cross-lingual retrieval with multilingual text encoders. 4.2.1

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965. 2.3

Yang Liu, Jinpeng Hu, Xiang Wan, and Tsung-Hui Chang. 2022b. A simple yet effective relation information guided approach for few-shot relation extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 757–763, Dublin, Ireland. Association for Computational Linguistics. 2.1.1

Ye Liu, Semih Yavuz, Rui Meng, Dragomir Radev, Caiming Xiong, and Yingbo Zhou. 2022c. Uni-parser: Unified semantic parser for question answering on knowledge base and database. *arXiv preprint arXiv:2211.05165*. 3.3.2

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692. 5.1.2, 5.2.1, 5.2.2

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*. 10.1.3

Cécile Logé, Emily Ross, David Yaw Amoah Dadey, Saahil Jain, Adriel Saporta, Andrew Y Ng, and Pranav Rajpurkar. Q-pain: A question answering dataset to measure social bias in pain management. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*. 11.2.1

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. 10.1.3

Niklas Lüdemann, Ageda Shiba, Nikolaos Thymianis, Nicolas Heist, Christopher Ludwig, and Heiko Paulheim. 2020. A knowledge graph for assessing agressive tax planning strategies. In *International Semantic Web Conference*, pages 395–410. Springer. 2.1.2

Andrew Luttrell and Vanessa Sawicki. 2020. Attitude strength: Distinguishing predictors versus defining features. *Social and Personality Psychology Compass*, 14(8):e12555. 7.3.1

Fukun Ma, Xuming Hu, Aiwei Liu, Yawen Yang, Shuang Li, Philip S. Yu, and Lijie Wen. 2023. AMR-based network for aspect-based sentiment analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 322–337, Toronto, Canada. Association for Computational Linguistics. 2.1

Aman Madaan. 2024. *Enhancing Language Models with Structured Reasoning*. Ph.D. thesis, Language Technologies Institute, Carnegie Mellon University. 1.2

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022a. Memory-assisted prompt editing to improve GPT-3 after deployment. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2833–2861, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. 2.1

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36. 2.2

Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. 2022b. Language models of code are few-shot commonsense learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1384–1403, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. 2.1, 8.1.2

Gaurav Maheshwari, Priyansh Trivedi, Denis Lukovnikov, Nilesh Chakraborty, Asja Fischer, and Jens Lehmann. 2019. Learning to rank query graphs for complex question answering over knowledge graphs. In *International semantic web conference*, pages 487–504. Springer. 2.1.2

Franc Mairesse, Marilyn Walker, et al. 2006. Words mark the nerds: Computational models of personality recognition through language. In *Proceedings of the annual meeting of the cognitive science society*, volume 28. 2.2.1, 7.2

Bodhisattwa Prasad Majumder, Oana Camburu, Thomas Lukasiewicz, and Julian Mcauley. 2022. Knowledge-grounded self-rationalization via extractive and natural language explanations. In *International Conference on Machine Learning*, pages 14786–14801. PMLR. 1.3, 2.2, 7.1

Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander Gelbukh, and Erik Cambria. 2019. Dialoguernn: An attentive rnn for emotion detection in conversations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6818–6825. 7.3.2

Magdalena Markowska, Mohammad Taghizadeh, Adil Soubki, Seyed Mirroshandel, and Owen Rambow. 2023. Finding common ground: Annotating and predicting common ground in spoken conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8221–8233, Singapore. Association for Computational Linguistics. 7.5

James R Martin and Peter R White. 2003. *The language of evaluation*, volume 2. Springer. 2.2.1, 7.2

James Robert Martin and David Rose. 2003. *Working with discourse: Meaning beyond the clause*. Bloomsbury Publishing. 2.2.1, 7.2

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *CoRR*, abs/1806.08730. 10.1

R. Thomas McCoy, Junghyun Min, and Tal Linzen. 2020a. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227, Online. Association for Computational Linguistics. 10.2.2

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2020b. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *57th Annual Meeting of the Association for Computational Linguistics, ACL 2019*, pages 3428–3448. Association for Computational Linguistics (ACL). 10.1.1

Shikib Mehri. 2022. *Towards Generalization in Dialog through Inductive Biases*. Ph.D. thesis, Language Technologies Institute, Carnegie Mellon University. 2.2.2

Zaiqiao Meng, Fangyu Liu, Thomas Clark, Ehsan Shareghi, and Nigel Collier. 2021. Mixture-of-partitions: Infusing large biomedical knowledge graphs into BERT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4672–4681, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 2.1

Miriam Meyerhoff. 2019. In pursuit of social meaning. *Journal of Sociolinguistics*, 23(3):303–315. 2.2.1

Mayank Mishra, Prince Kumar, Riyaz Bhat, Rudra Murthy V, Danish Contractor, and Srikanth Tamilselvam. 2023. Prompting with pseudo-code instructions. *arXiv preprint arXiv:2305.11790*. 8.1.2

Saif Mohammad. 2018. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 174–184, Melbourne, Australia. Association for Computational Linguistics. 9.3.3, 2.3

Salman Mohammed, Peng Shi, and Jimmy Lin. 2018. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 291–296, New Orleans, Louisiana. Association for Computational Linguistics. 2.1.2

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. *Foundations of machine learning*. MIT press. 1.1

José David Moreno, Jose A Martinez-Huertas, Ricardo Olmos, Guillermo Jorge-Botana, and Juan Botella. 2021. Can personality traits be measured analyzing written language? a meta-analytic study on computational methods. *Personality and Individual Differences*, 177:110818. 2.2.1, 7.2

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics. 9.1.4, 2.3

Melissa Mulcahy and Bethanie Gouldthorp. 2016. Positioning the reader: the effect of narrative point-of-view and familiarity of experience on situation model construction. *Language and Cognition*, 8(1):96–123. 9.1.1

Sheshera Mysore, Zachary Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. 2019. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic

structures. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 56–64, Florence, Italy. Association for Computational Linguistics. 3.2.1

Arijit Nag, Bidisha Samanta, Animesh Mukherjee, Niloy Ganguly, and Soumen Chakrabarti. 2021. A data bootstrapping recipe for low-resource multilingual relation classification. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 575–587, Online. Association for Computational Linguistics. 2.1.1, 3.2.2

Aakanksha Naik, Jill Lehman, and Carolyn Rosé. 2022. Adapting to the long tail: A meta-analysis of transfer learning research for language understanding tasks. *Transactions of the Association for Computational Linguistics*, 10:956–980. 1.1, 2.3

Rungsiman Nararatwong, Natthawut Kertkeidkachorn, and Ryutaro Ichise. 2022. KIQA: Knowledge-infused question answering model for financial table-text data. In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 53–61, Dublin, Ireland and Online. Association for Computational Linguistics. 2.1

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy. Association for Computational Linguistics. 4.1.1

Dong Nguyen, A Seza Doğruöz, Carolyn P Rosé, and Franciska De Jong. 2016. Computational sociolinguistics: A survey. *Computational linguistics*, 42(3):537–593. 7.2

Dong Nguyen, Laura Rosseel, and Jack Grieve. 2021a. On learning and representing social meaning in nlp: a sociolinguistic perspective. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 603–612. 2.2.1, 7.2

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021b. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 80–90, Online. Association for Computational Linguistics. 3.2.2, 4.2.1

Noriki Nishida and Yuji Matsumoto. 2022. Out-of-domain discourse dependency parsing via bootstrapping: An empirical analysis on its effectiveness and limitation. *Transactions of the Association for Computational Linguistics*, 10:127–144. 2.3

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666. 4.2.1

Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5153–5176, Singapore.

Association for Computational Linguistics. 2.1

Daniel J O'Keefe. 2002. Guilt as a mechanism of persuasion. *The persuasion handbook: Developments in theory and practice*, pages 329–344. 7.3.1

Junwoo Park, Youngwoo Cho, Haneol Lee, Jaegul Choo, and Edward Choi. 2020. Knowledge graph-based question answering with electronic health records. *arXiv preprint arXiv:2010.09394*. 2.1.2

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021. Soloist: Building task bots at scale with transfer learning and machine teaching. *Transactions of the Association for Computational Linguistics*, 9:807–824. 2.2.2

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020a. Few-shot natural language generation for task-oriented dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 172–182. 2.2.2

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020b. Learning from Context or Names? An Empirical Study on Neural Relation Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3661–3672, Online. Association for Computational Linguistics. 2.1.1

Javiera Perez Gomez. 2021. Verbal microaggressions as hyper-implicatures. *Journal of Political Philosophy*, 29(3):375–403. 9.1.1

Bryan Perozzi, Vivek Kulkarni, Haochen Chen, and Steven Skiena. 2017. Don't walk, skip! online learning of multi-scale network embeddings. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 258–265. 5.1.2, 5.2.2, 6.2.3

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics. 2.2

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536, Florence, Italy. Association for Computational Linguistics. (document), 7.3.2, 7.2

Matt Post and Daniel Gildea. 2008. Parsers as language models for statistical machine translation. In *Proceedings of the 8th Conference of the Association for Machine Translation in the Americas: Research Papers*, pages 172–181, Waikiki, USA. Association for Machine Translation in the Americas. 2.1

Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. 2021. What to pre-train on? Efficient intermediate task selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10585–10605, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 2.3

Nima Pourdamghani, Yang Gao, Ulf Hermjakob, and Kevin Knight. 2014. Aligning English strings with Abstract Meaning Representation graphs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 425–429, Doha, Qatar. Association for Computational Linguistics. 4.1.1

Jakob Prange, Nathan Schneider, and Lingpeng Kong. 2022. Linguistic frameworks go toe-to-toe at neuro-symbolic language modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4375–4391, Seattle, United States. Association for Computational Linguistics. 2.1, 2.1.1

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247, Online. Association for Computational Linguistics. 2.3

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020a. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics. 3.2.2

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020b. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 4.1.1, 4.2.1

Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Zhong Zhang, Jie Zhou, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 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)*, pages 1088–1113, Bangkok, Thailand. Association for Computational Linguistics. 7.5

Meng Qu, Tianyu Gao, Louis-Pascal Xhonneux, and Jian Tang. 2020. Few-shot relation extraction via Bayesian meta-learning on relation graphs. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7867–7876. PMLR. 2.1.1

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. 1.2, 8.2.2

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551. 8.2.2, 10.1.3

Vipul Raheja and Joel Tetreault. 2019. Dialogue Act Classification with Context-Aware Self-Attention. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3727–3733, Minneapolis, Minnesota. Association for Computational

Linguistics. 7.3.2

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics. 2.2

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. 10.1.2

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in NLP—A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, Barcelona, Spain (Online). International Committee on Computational Linguistics. 2.3

Kavel Rao, Liwei Jiang, Valentina Pyatkin, Yuling Gu, Niket Tandon, Nouha Dziri, Faeze Brahman, and Yejin Choi. 2023. What makes it ok to set a fire? iterative self-distillation of contexts and rationales for disambiguating defeasible social and moral situations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12140–12159, Singapore. Association for Computational Linguistics. 2.2, 7.5

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 9.1.4, 2.3

Hongyu Ren, Hanjun Dai, Bo Dai, Xinyun Chen, Michihiro Yasunaga, Haitian Sun, Dale Schuurmans, Jure Leskovec, and Denny Zhou. 2021. Lego: Latent execution-guided reasoning for multi-hop question answering on knowledge graphs. In *International Conference on Machine Learning*, pages 8959–8970. PMLR. 2.1.2

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics. 2.3

Shachar Rosenman, Alon Jacovi, and Yoav Goldberg. 2020. Exposing Shallow Heuristics of Relation Extraction Models with Challenge Data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3702–3710, Online. Association for Computational Linguistics. 4.1.1

Alvin E Roth. 1988. Introduction to the shapley value. *The Shapley value*, 1. 9.3.5

Benedek Rozemberczki and Rik Sarkar. 2020. Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1325–1334. 5.2.2

Qian Ruan, Malte Ostendorff, and Georg Rehm. 2022. HiStruct+: Improving extractive text summarization with hierarchical structure information. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1292–1308, Dublin, Ireland. Association for Computational Linguistics. 2.1

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*. 2.3

Devendra Sachan, Yuhao Zhang, Peng Qi, and William L. Hamilton. 2021. Do syntax trees help pre-trained transformers extract information? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2647–2661, Online. Association for Computational Linguistics. 2.1, 2.1.1, 4.2

Amrita Saha, Ghulam Ahmed Ansari, Abhishek Laddha, Karthik Sankaranarayanan, and Soumen Chakrabarti. 2019. Complex program induction for querying knowledge bases in the absence of gold programs. *Transactions of the Association for Computational Linguistics*, 7:185–200. 2.1.2, 2.1.3

Swarnadeep Saha, Yixin Nie, and Mohit Bansal. 2020. ConjNLI: Natural language inference over conjunctive sentences. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8240–8252, Online. Association for Computational Linguistics. 10.1.1

Victor Tejedor San José. 2019. The role of humor and threat on predicting resistance and persuasion. 7.3.1

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, et al. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*. 2.2.2

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2020a. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490, Online. Association for Computational Linguistics. 2.2

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2020b. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490, Online. Association for Computational Linguistics. 9.1.1

Maarten Sap, Ronan Le Bras, Daniel Fried, and Yejin Choi. 2022. Neural theory-of-mind? on the limits of social intelligence in large LMs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3762–3780, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. 2.2, 7.5

Gabriel Sarch, Yue Wu, Michael Tarr, and Katerina Fragkiadaki. 2023. Open-ended instructable embodied agents with memory-augmented large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3468–3500, Singapore. Association for Computational Linguistics. 2.1

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507, Online. Association for Computational Linguistics. 2.1.2, 3.3.1, 5.2.1, 5.2.2

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80. 4.2.1

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018a. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer. 4.1.1, 5.1.2, 5.2.2

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018b. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer. 2

Hendrik Schuff, Hsiu-Yu Yang, Heike Adel, and Ngoc Thang Vu. 2021. Does external knowledge help explainable natural language inference? automatic evaluation vs. human ratings. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 26–41, Punta Cana, Dominican Republic. Association for Computational Linguistics. 2.2

Ashish Sharma, Adam Miner, David Atkins, and Tim Althoff. 2020. A computational approach to understanding empathy expressed in text-based mental health support. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5263–5276, Online. Association for Computational Linguistics. (document), 2.2.1, 4, 7.5, **??**, 9.2.1, **??**

Yikang Shen, Shawn Tan, Alessandro Sordoni, Siva Reddy, and Aaron Courville. 2021. Explicitly modeling syntax in language models with incremental parsing and a dynamic oracle. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1660–1672, Online. Association for Computational Linguistics. 2.1

Steven J Sherman and Larry Gorkin. 1980. Attitude bolstering when behavior is inconsistent with central attitudes. *Journal of Experimental Social Psychology*, 16(4):388–403. 7.3.1

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR. 9.3.2

Yiheng Shu, Zhiwei Yu, Yuhan Li, Börje F Karlsson, Tingting Ma, Yuzhong Qu, and Chin-Yew Lin. 2022. Tiara: Multi-grained retrieval for robust question answering over large knowledge bases. *arXiv preprint arXiv:2210.12925*. 3.3.2, 3.3.2

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180. 2.2

Paul Surgi Speck and Michael T Elliott. 1997. Predictors of advertising avoidance in print and broadcast media. *Journal of Advertising*, 26(3):61–76. 7.3.1

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31. 2.1

Saurabh Srivastava, Mayur Patidar, Sudip Chowdhury, Puneet Agarwal, Indrajit Bhattacharya, and Gautam Shroff. 2021. Complex question answering on knowledge graphs using machine translation and multi-task learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3428–3439, Online. Association for Computational Linguistics. 5.1.1, 5.1.2

Yu Su, Huan Sun, Brian Sadler, Mudhakar Srivatsa, Izzeddin Gür, Zenghui Yan, and Xifeng Yan. 2016. On generating characteristic-rich question sets for qa evaluation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 562–572. 3.3.2, 3.3.2

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2380–2390, Hong Kong, China. Association for Computational Linguistics. 2.1.2, 5.1.1

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242, Brussels, Belgium. Association for Computational Linguistics. 2.1.2

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics. 2.3

Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith. 2018. Syntactic scaffolds for semantic structures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3772–3782, Brussels, Belgium. Association for Computational Linguistics. 1.3

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics. 3.3.2

Niket Tandon, Aman Madaan, Peter Clark, and Yiming Yang. 2022. Learning to repair: Repairing model output errors after deployment using a dynamic memory of feedback. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 339–352, Seattle, United States. Association for Computational Linguistics. 2.1

Gemma Team. 2024. Gemma. 9.2.3

Dung Thai, Dhruv Agarwal, Mudit Chaudhary, Wenlong Zhao, Rajarshi Das, Jay-Yoon Lee, Hannaneh Hajishirzi, Manzil Zaheer, and Andrew McCallum. 2023. Machine reading comprehension using case-based reasoning. In *Findings of the Association for Computational*

Linguistics: EMNLP 2023, pages 8414–8428, Singapore. Association for Computational Linguistics. 2.2

Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2022a. Contrastive representation distillation. 6.2.2

Yuanhe Tian, Yan Song, and Fei Xia. 2022b. Improving relation extraction through syntax-induced pre-training with dependency masking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1875–1886, Dublin, Ireland. Association for Computational Linguistics. 2.1

Zakary L Tormala. 2008. A new framework for resistance to persuasion: The resistance appraisals hypothesis. *Attitudes and attitude change*, pages 213–234. 7.3.1

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*. 1.2, 8.1.4, 8.2.2

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554. 10.1.2

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR. 5.2.1

Anita L Vangelisti, John A Daly, and Janine Rae Rudnick. 1991. Making people feel guilty in conversations: Techniques and correlates. *Human Communication Research*, 18(1):3–39. 7.3.1

Shikhar Vashishth, Manik Bhandari, Prateek Yadav, Piyush Rai, Chiranjib Bhattacharyya, and Partha Talukdar. 2019. Incorporating syntactic and semantic information in word embeddings using graph convolutional networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3308–3318, Florence, Italy. Association for Computational Linguistics. 2.1

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. RESIDE: Improving distantly-supervised neural relation extraction using side information. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1257–1266, Brussels, Belgium. Association for Computational Linguistics. 2.1

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85. 2.1

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou', and Daniel Cer. 2022. SPoT: Better frozen model adaptation through soft prompt transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5039–5059, Dublin, Ireland. Association for Computational Linguistics. 2.3

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022a. DeepStruct: Pretraining of language models for structure prediction. In *Findings of the Association*

for Computational Linguistics: ACL 2022, pages 803–823, Dublin, Ireland. Association for Computational Linguistics. 2.1.1

Cunxiang Wang, Pai Liu, and Yue Zhang. 2021a. Can generative pre-trained language models serve as knowledge bases for closed-book QA? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3241–3251, Online. Association for Computational Linguistics. 2.2

Ke Wang, Jiayi Wang, Niyu Ge, Yangbin Shi, Yu Zhao, and Kai Fan. 2020a. Computer assisted translation with neural quality estimation and automatic post-editing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2175–2186, Online. Association for Computational Linguistics. 2.1

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023a. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*. 9.1.4, 2.3

Minjia Wang, Pingping Lin, Siqi Cai, Shengnan An, Shengjie Ma, Zeqi Lin, Congrui Huang, and Bixiong Xu. 2025. Stand-guard: A small task-adaptive content moderation model. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 1–20. 2.2.2

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021b. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194. 2.1

Xu Wang, Shuai Zhao, Bo Cheng, Jiale Han, Yingting Li, Hao Yang, and Guoshun Nan. 2020b. Hgman: multi-hop and multi-answer question answering based on heterogeneous knowledge graph (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13953–13954. 2.1.2, 5.1.2, 5.2.1, 5.2.2

Xu Wang, Shuai Zhao, Jiale Han, Bo Cheng, Hao Yang, Jianchang Ao, and Zhenzi Li. 2020c. Modelling long-distance node relations for KBQA with global dynamic graph. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2572–2582, Barcelona, Spain (Online). International Committee on Computational Linguistics. 2.1.2, 5.1.1, 5.1.2, 5.2.1, 5.2.2, 5.3

Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. 2019a. Persuasion for good: Towards a personalized persuasive dialogue system for social good. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5635–5649, Florence, Italy. Association for Computational Linguistics. (document), 7.1, 1, 7.8, **??**, 9.2.1, **??**

Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. 2019b. Persuasion for good: Towards a personalized persuasive dialogue system for social good. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5635–5649, Florence, Italy. Association for Computational Linguistics. 2.2.1, 7.3.1

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022b. Self-consistency improves chain of thought reasoning in

language models. In *The Eleventh International Conference on Learning Representations*. 2.2

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. 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)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics. 1.2, 1.3

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023c. 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)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics. 2.2

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. 2022c. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109. 1.2, 1.2, 1.3, 2.2.2

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392. 6.1.2

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641. 6.1.2

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark. Association for Computational Linguistics. 2.3

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. 8.1.3

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. 2.2

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022c. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837. 1.3, 2.2, 7.1

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics. 2.3

Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. 2021. Measuring association between labels and free-text rationales. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10266–10284, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 1.3, 2.2, 7.1, 7.5

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics. 10.1.1

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*. 10.1.3

Wendy Wood and Carl A Kallgren. 1988. Communicator attributes and persuasion: Recipients' access to attitude-relevant information in memory. *Personality and Social Psychology Bulletin*, 14(1):172–182. 7.3.1

Peter Wright. 1975. Factors affecting cognitive resistance to advertising. *journal of Consumer Research*, 2(1):1–9. 7.3.1

Chien-Sheng Wu, Steven CH Hoi, Richard Socher, and Caiming Xiong. 2020. Tod-bert: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929. 2.2.2

Zhaofeng Wu, Hao Peng, and Noah A. Smith. 2021. Infusing finetuning with semantic dependencies. *Transactions of the Association for Computational Linguistics*, 9:226–242. 1.3, 2.1

Zhen Wu, Ritam Dutt, Luke M Breitfeller, Armineh Nourbakhsh, Siddharth Parekh, and Carolyn Rosé. 2025. r^2-cod: Understanding text-graph complementarity in relational reasoning via knowledge co-distillation. *arXiv preprint arXiv:2508.01475*. 11.2.2

Zhen Wu, Ritam Dutt, and Carolyn Penstein Rosé. 2024. Evaluating large language models on social signal sensitivity: An appraisal theory approach. In *The First Human-Centered Large Language Modeling Workshop*, page 67. 9.3.3, 2.3

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. UnifiedSKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 602–631, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. 3.3.2, 6.2.3

Weimin Xiong, Yifan Song, Peiyi Wang, and Sujian Li. 2023. Rationale-enhanced language models are better continual relation learners. In *Proceedings of the 2023 Conference on Empirical*

Methods in Natural Language Processing, pages 15489–15497, Singapore. Association for Computational Linguistics. 2.2

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Improving question answering over incomplete KBs with knowledge-aware reader. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4258–4264, Florence, Italy. Association for Computational Linguistics. 2.1.2

Runxin Xu, Fuli Luo, Zhiyuan Zhang, Chuanqi Tan, Baobao Chang, Songfang Huang, and Fei Huang. 2021. Raise a child in large language model: Towards effective and generalizable fine-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9514–9528, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 2.3

Runxin Xu, Peiyi Wang, Tianyu Liu, Shuang Zeng, Baobao Chang, and Zhifang Sui. 2022. A two-stream AMR-enhanced model for document-level event argument extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5025–5036, Seattle, United States. Association for Computational Linguistics. 2.1.1

Lilong Xue, Dan Zhang, Yuxiao Dong, and Jie Tang. 2024. Autore: Document-level relation extraction with large language models. 2.1.1

Yoko Yamakata, Shinsuke Mori, and John Carroll. 2020a. English recipe flow graph corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5187–5194, Marseille, France. European Language Resources Association. 3.2.1

Yoko Yamakata, Shinsuke Mori, and John Carroll. 2020b. English recipe flow graph corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5187–5194, Marseille, France. European Language Resources Association. 3.2.1

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. 4.2.2, 4.2.3

Diyi Yang, Jiaao Chen, Zichao Yang, Dan Jurafsky, and Eduard Hovy. 2019. Let's make your request more persuasive: Modeling persuasive strategies via semi-supervised neural nets on crowdfunding platforms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3620–3630, Minneapolis, Minnesota. Association for Computational Linguistics. 7.3.1

Kaijia Yang, Nantao Zheng, Xinyu Dai, Liang He, Shujian Huang, and Jiajun Chen. 2020. Enhance

prototypical network with text descriptions for few-shot relation classification. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2273–2276. 2.1.1

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, pages 2369–2380. Association for Computational Linguistics. 10.1.2

Hao-Ren Yao, Luke Breitfeller, Aakanksha Naik, Chunxiao Zhou, and Carolyn Rose. 2024. Distilling multi-scale knowledge for event temporal relation extraction. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 2971–2980. 6.2.2

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics. 2.1.2

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021a. CrossFit: A few-shot learning challenge for cross-task generalization in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7163–7189, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 2.3

Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. 2021b. Rng-kbqa: Generation augmented iterative ranking for knowledge base question answering. *arXiv preprint arXiv:2109.08678*. 3.3.2, 6.1.1

Zhi-Xiu Ye and Zhen-Hua Ling. 2019a. Distant supervision relation extraction with intra-bag and inter-bag attentions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2810–2819, Minneapolis, Minnesota. Association for Computational Linguistics. 2.1.1

Zhi-Xiu Ye and Zhen-Hua Ling. 2019b. Multi-level matching and aggregation network for few-shot relation classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2872–2881, Florence, Italy. Association for Computational Linguistics. 2.1.1

Akhila Yerukola, Saujas Vaduguru, Daniel Fried, and Maarten Sap. 2024. Is the pope catholic? yes, the pope is catholic. generative evaluation of non-literal intent resolution in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 265–275, Bangkok, Thailand. Association for Computational Linguistics. 7.5

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany. Association for Computational Linguistics. 3.3.1, 3.3.2, 3.3.2

Donghan Yu, Sheng Zhang, Patrick Ng, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Yiqun Hu, William Wang, Zhiguo Wang, and Bing Xiang. 2022a. Decaf: Joint decoding of answers and logical forms for question answering over knowledge bases. *arXiv preprint arXiv:2210.00063*. 3.3.2

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2022b. KG-FiD: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4961–4974, Dublin, Ireland. Association for Computational Linguistics. 2.1

Tianshu Yu, Min Yang, and Xiaoyan Zhao. 2022c. Dependency-aware prototype learning for few-shot relation classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2339–2345, Gyeongju, Republic of Korea. International Committee on Computational Linguistics. 2.1.1

Zulipiye Yusupujiang and Jonathan Ginzburg. 2023. Unravelling indirect answers to wh-questions: Corpus construction, analysis, and generation. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 336–348, Prague, Czechia. Association for Computational Linguistics. 7.5, 9.1.1

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Human language technologies 2007: The conference of the North American chapter of the association for computational linguistics; proceedings of the main conference*, pages 260–267. 2.2

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488. 2.2, 7.5

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Noëmi Aepli, Željko Agić, Lars Ahrenberg, Gabrielė Aleksandravičiūtė, Lene Antonsen, Katya Aplonova, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, Colin Batchelor, John Bauer, Sandra Bellato, Kepa Bengoetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guino-

vart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Johannes Heinecke, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Takumi Ikeda, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Markus Juutinen, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Kamil Kopacewicz, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phng Lê H`ông, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Maria Liovina, Yuan Li, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Tomohiko Morioka, Shinsuke Mori, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lng Nguy˜ên Thị, Huy`ên Nguy˜ên Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ̀ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Daria Petrova, Slav Petrov, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roșca, Olga Rudina, Jack Rueter, Shoval Sadde, Benoît Sagot, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Muh Shohibussirri, Dmitry Sichinava, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Takaaki Tanaka, Isabelle Tellier, Guillaume Thomas, Liisi Torga, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska,

Mary Yako, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Manying Zhang, and Hanzhi Zhu. 2019. Universal dependencies 2.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. 4.2.1

Chiyu Zhang and Muhammad Abdul-Mageed. 2022. Improving social meaning detection with pragmatic masking and surrogate fine-tuning. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 141–156, Dublin, Ireland. Association for Computational Linguistics. 2.2.1, 7.2

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*. 1.1

Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022a. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5773–5784. 3.3.2

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022b. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*. 10.1.3

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. Greaselm: Graph reasoning enhanced language models for question answering. 2.2

Yuan Zhang, Regina Barzilay, and Tommi Jaakkola. 2017. Aspect-augmented adversarial networks for domain adaptation. *Transactions of the Association for Computational Linguistics*, 5:515–528. 2.3

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics. 2.1

Zhiwei Zhang, Jiyi Li, Fumiyo Fukumoto, and Yanming Ye. 2021a. Abstract, rationale, stance: A joint model for scientific claim verification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3580–3586, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 1.2

Zixuan Zhang and Heng Ji. 2021. Abstract Meaning Representation guided graph encoding and decoding for joint information extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–49, Online. Association for Computational Linguistics. 2.1, 2.1.1

Zixuan Zhang, Nikolaus Nova Parulian, Heng Ji, Ahmed S Elsayed, Skatje Myers, and Martha Palmer. 2021b. Fine-grained information extraction from biomedical literature based on knowledge-enriched abstract meaning representation. In *Proc. The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*. 2.1.1

Xinyu Zhao, Shih-Ting Lin, and Greg Durrett. 2021. Effective distant supervision for temporal relation extraction. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 195–203, Kyiv, Ukraine. Association for Computational Linguistics. 2.1.1

Li Zhenzhen, Yuyang Zhang, Jian-Yun Nie, and Dongsheng Li. 2022. Improving few-shot relation classification by prototypical representation learning with definition text. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 454–464, Seattle, United States. Association for Computational Linguistics. 2.1.1

Ruiqi Zhong, Mitchell Stern, and Dan Klein. 2020. Semantic scaffolds for pseudocode-to-code generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2283–2295, Online. Association for Computational Linguistics. 1.3

Pei Zhou, Aman Madaan, Srividya Pranavi Potharaju, Aditya Gupta, Kevin R McKee, Ari Holtzman, Jay Pujara, Xiang Ren, Swaroop Mishra, Aida Nematzadeh, et al. 2023a. How far are large language models from agents with theory-of-mind? 2.2

Xuhui Zhou, Hao Zhu, Akhila Yerukola, Thomas Davidson, Jena D. Hwang, Swabha Swayamdipta, and Maarten Sap. 2023b. COBRA frames: Contextual reasoning about effects and harms of offensive statements. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6294–6315, Toronto, Canada. Association for Computational Linguistics. 9.1.1

Yiheng Zhou, He He, Alan W Black, and Yulia Tsvetkov. 2019. A dynamic strategy coach for effective negotiation. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 367–378, Stockholm, Sweden. Association for Computational Linguistics. 7.3.1

Yucheng Zhou, Xiubo Geng, Tao Shen, Wenqiang Zhang, and Daxin Jiang. 2021. Improving zero-shot cross-lingual transfer for multilingual question answering over knowledge graph. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5822–5834, Online. Association for Computational Linguistics. 2.1.2

Shuguang Zhu, X. Cheng, and Sen Su. 2020. Knowledge-based question answering by tree-to-sequence learning. *Neurocomputing*, 372:64–72. 2.1.2

Alex Zhuang, Ge Zhang, Tianyu Zheng, Xinrun Du, Junjie Wang, Weiming Ren, Stephen W Huang, Jie Fu, Xiang Yue, and Wenhu Chen. 2024. Structlm: Towards building generalist models for structured knowledge grounding. *arXiv preprint arXiv:2402.16671*. 1.2, 1.3

Caleb Ziems, Jane Dwivedi-Yu, Yi-Chia Wang, Alon Halevy, and Diyi Yang. 2023. NormBank: A knowledge bank of situational social norms. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7756–7776, Toronto, Canada. Association for Computational Linguistics. 2.1

Julia Zuwerink Jacks and Kimberly A Cameron. 2003. Strategies for resisting persuasion. *Basic and applied social psychology*, 25(2):145–161. 7.3.1

Appendices

Appendix A

Additional Results

We present the comprehensive results to avoid cluttering of the main thesis chapter namely ID and TF results over all datasets with line-plots showing a direct visualization of the same in Figure 1 and zero-shot resulys in Figure 2. We also highlight model mispredictions in terms of confusion matrices (Figures 3.

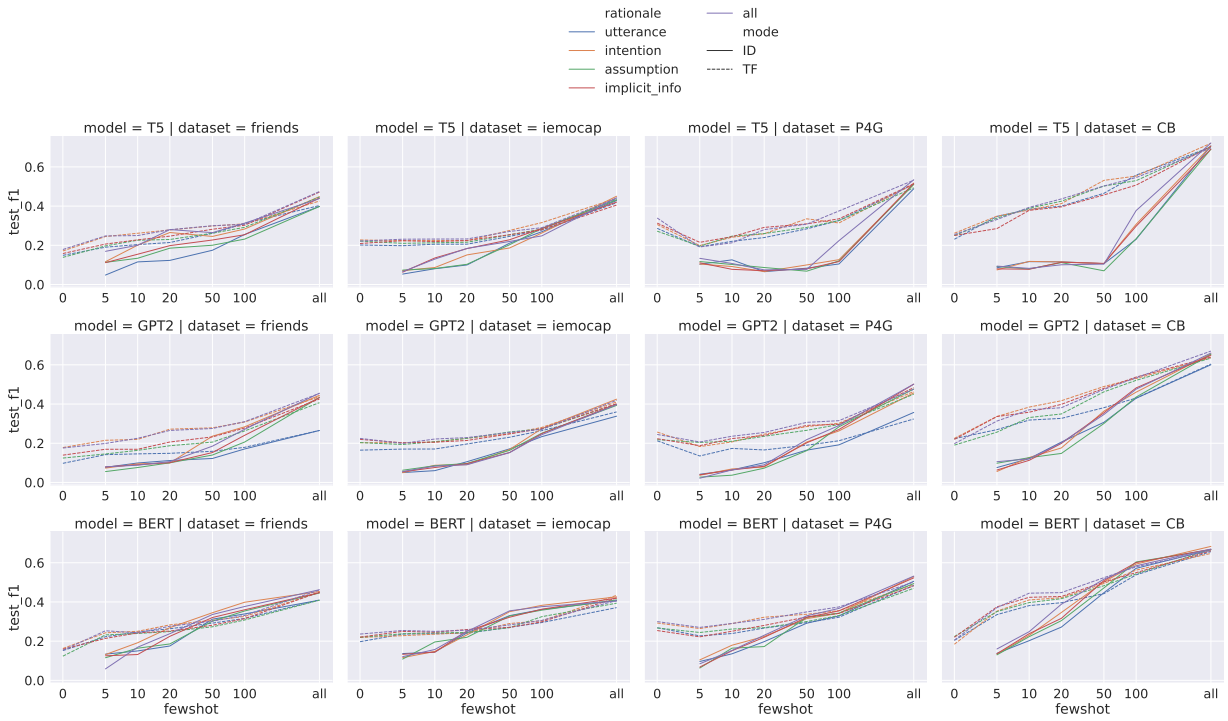


Figure 1: Performance of the base-variants of models (BERT, GPT2, and T5) on the four datasets for different few-shot examples for all rationales. The solid and dashed lines correspond to the indomain (ID) and transfer (TF) case respectively.

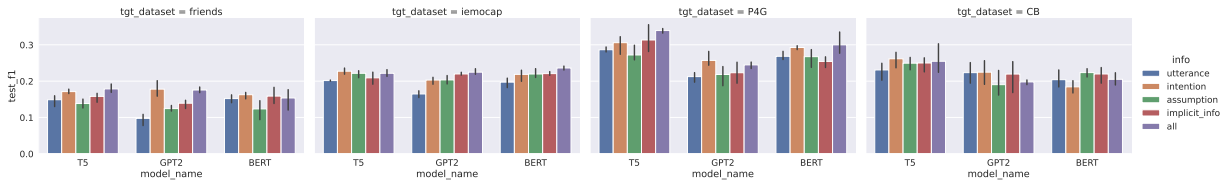


Figure 2: Performance of the base-variants of models (BERT, GPT2, and T5) on the four datasets in a zero-shot transfer setting, where models trained for the similar task on a given source domain was then applied to the new target domain (e.g. P4G $\rightarrow$ CB and CB $\rightarrow$ P4G for RES and friends $\rightarrow$ iemocap and iemocap $\rightarrow$ friends for ERC.)

Table 1: Performance of different models on the **CB (Craigslist Bargain)** dataset for both in-domain (ID) and transfer (TF) setting across different few-shot splits (5, 10, 20, 50, 100) and the entire dataset (denoted by "All"). The different rationales explored in this work are denoted by only utterance (-), utterance with speaker's intention (INT), utterance with the hearer's assumption (ASM), utterance with implicit information (IMP), and utterance with all the aforementioned rationales included i.e. INT, ASM, and IMP, and is denoted by ALL.

Model	Mode	Rationale	5	10	20	50	100	All
bert	ID	-	13.8±4.7	20.2±1.4	27.2±6.8	44.7±2.4	57.2±2.0	66.7±3.6
		INT	13.4±5.5	22.9±0.7	34.6±3.4	50.3±2.5	59.3±1.8	**68.4±1.7**
		ASM	13.0±5.9	22.3±5.0	30.4±1.7	47.2±1.4	**60.4±3.0**	66.6±0.7
		IMP	13.6±5.0	23.8±3.1	31.6±6.7	50.9±2.5	60.1±1.9	66.9±0.3
		ALL	**16.0±6.4**	**24.8±4.4**	**38.8±2.2**	**51.6±1.2**	58.5±1.9	67.0±0.7
	TF	-	33.5±2.1	38.1±3.0	39.6±3.2	44.2±3.3	53.8±1.2	65.7±0.9
		INT	35.3±0.4	41.0±4.0	42.2±1.9	49.4±2.9	56.3±1.2	64.8±3.5
		ASM	35.1±0.8	39.7±2.4	41.7±1.4	48.3±1.8	54.3±2.1	**66.8±0.6**
		IMP	**37.5±1.4**	42.4±1.6	42.8±0.5	50.2±4.5	55.0±1.9	66.1±2.9
		ALL	37.1±1.9	**44.5±2.9**	**44.8±0.7**	**52.4±2.0**	**57.9±0.6**	66.3±1.6
gpt2	ID	-	7.6±6.3	12.1±5.7	**20.6±4.4**	30.7±6.3	43.0±1.0	60.0±0.9
		INT	5.6±1.7	**12.7±5.1**	17.6±2.8	**36.4±5.4**	46.1±0.2	65.6±2.0
		ASM	9.8±5.1	12.6±3.1	14.8±3.7	30.1±0.2	43.5±3.5	65.3±1.3
		IMP	6.3±2.8	11.3±5.1	19.9±5.9	35.5±2.8	**48.2±4.3**	64.9±1.6
		ALL	**10.6±6.1**	12.1±5.8	20.2±4.5	34.7±3.5	47.6±2.5	**66.0±1.5**
	TF	-	26.9±2.4	31.8±1.3	32.7±2.3	38.1±0.6	43.0±2.7	60.4±0.4
		INT	**33.7±7.4**	**38.4±1.7**	**41.7±3.1**	**48.9±0.9**	53.0±0.4	63.7±3.2
		ASM	25.6±6.4	33.1±1.8	34.9±2.6	46.2±1.9	52.1±1.0	63.4±2.9
		IMP	33.6±7.3	35.8±4.2	39.8±2.7	48.0±4.6	**53.8±3.0**	64.0±1.1
		ALL	31.3±4.8	37.0±5.0	38.1±3.4	47.4±1.8	53.4±1.8	**67.0±2.8**
t5-base	ID	-	8.5±3.2	11.7±0.6	**11.6±1.3**	10.6±3.2	23.1±11.4	70.8±1.8
		INT	7.3±2.5	**11.8±1.9**	11.5±1.7	10.7±3.4	30.7±1.6	70.6±2.8
		ASM	9.2±2.6	7.9±0.9	11.2±2.3	7.0±0.3	23.4±3.1	69.0±1.8
		IMP	8.0±4.3	7.8±1.6	11.3±1.1	**10.8±3.0**	29.8±2.8	69.1±2.6
		ALL	**9.4±2.5**	8.2±2.2	10.1±1.5	10.4±4.0	**37.7±3.9**	**72.2±0.5**
	TF	-	34.7±1.8	38.4±2.0	40.0±1.1	46.5±4.4	**55.8±1.8**	70.1±2.9
		INT	**34.9±4.3**	38.1±2.2	41.3±3.5	53.1±0.8	55.3±3.3	**72.1±0.7**
		ASM	33.9±3.0	38.9±0.5	42.5±2.6	50.2±3.0	53.0±3.1	70.3±3.4
		IMP	28.5±2.2	37.8±3.1	39.6±5.3	45.7±0.8	50.7±1.6	70.5±1.3
		ALL	33.2±4.5	**39.4±2.0**	**43.7±2.2**	**50.3±3.9**	54.6±3.7	69.6±1.7

Table 2: Performance of different models on the **P4G (Persuasion for Good)** dataset for both in-domain (ID) and transfer (TF) setting across different few-shot splits (5, 10, 20, 50, 100) and the entire dataset (denoted by "All"). The different rationales explored in this work are denoted by only utterance (-), utterance with speaker's intention (INT), utterance with the hearer's assumption (ASM), utterance with implicit information (IMP), and utterance with all the aforementioned rationales included i.e. INT, ASM, and IMP, and is denoted by ALL.

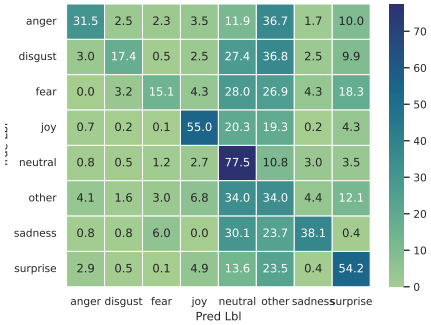
Model	Mode	Rationale	5	10	20	50	100	ALL
bert	ID	-	9.6±0.2	13.5±3.8	19.8±0.6	29.2±0.7	32.9±1.1	50.6±2.5
		INT	**10.4±5.0**	**17.9±2.9**	22.4±3.6	31.5±1.4	34.2±1.8	53.0±1.6
		ASM	6.3±3.2	16.4±1.6	17.2±5.8	32.1±0.5	34.3±1.4	49.4±8.1
		IMP	6.8±4.6	15.1±2.2	22.0±1.9	32.2±0.7	35.5±1.5	52.3±1.7
		ALL	8.4±7.5	15.3±6.6	**23.0±1.1**	**32.9±0.7**	**36.7±1.0**	**53.2±1.4**
	TF	-	22.7±0.3	23.9±0.9	26.7±1.5	29.5±2.6	32.2±0.4	48.4±1.4
		INT	26.4±1.1	**29.0±2.7**	**32.2±1.0**	33.7±0.4	35.6±2.0	49.0±0.6
		ASM	24.4±2.8	26.2±2.0	26.9±1.0	30.0±0.6	33.0±1.6	47.0±3.5
		IMP	22.2±3.3	25.1±2.3	28.0±1.2	32.4±0.6	34.2±1.5	48.2±0.7
		ALL	**27.0±0.9**	29.0±2.2	31.1±0.7	**34.9±2.0**	**37.5±2.3**	**50.2±3.5**
gpt2	ID	-	4.2±2.5	6.3±4.1	**10.0±3.2**	16.5±1.1	19.2±1.9	35.7±4.4
		INT	3.7±2.4	**6.9±2.7**	7.9±3.0	20.6±1.4	26.3±3.3	45.7±1.6
		ASM	2.7±1.1	3.7±1.2	7.3±1.9	16.2±5.3	28.4±5.7	47.7±2.4
		IMP	3.9±2.5	6.4±3.5	8.2±4.5	20.2±5.5	27.0±2.6	50.1±2.6
		ALL	2.1±0.8	6.4±1.9	9.0±3.9	**21.8±1.7**	**29.0±4.5**	**50.1±1.4**
	TF	-	13.5±1.8	16.3±0.5	16.6±2.9	19.0±0.6	21.3±1.3	32.4±3.6
		INT	18.3±2.0	20.7±0.4	24.6±0.6	28.5±2.1	30.2±0.3	46.4±1.8
		ASM	20.4±1.2	21.0±1.1	23.6±1.5	26.6±1.9	29.6±0.9	45.0±2.0
		IMP	18.8±3.6	22.4±1.8	23.9±1.6	29.1±1.7	29.7±2.3	**48.4±2.1**
		ALL	**20.6±2.7**	**23.6±0.3**	**25.5±2.2**	**30.6±0.2**	**31.5±2.0**	47.5±2.0
t5-base	ID	-	10.3±0.9	12.6±2.6	6.5±2.3	8.3±2.6	10.5±1.9	48.8±0.9
		INT	10.4±1.0	9.2±5.6	6.6±0.2	10.0±0.8	12.7±0.7	51.2±1.4
		ASM	11.7±2.1	10.2±4.3	8.7±3.9	6.8±0.8	12.0±3.2	51.1±0.8
		IMP	11.1±1.8	7.7±3.5	7.0±2.7	8.0±4.2	11.7±8.9	51.7±3.0
		ALL	13.4±1.1	10.7±4.6	7.4±3.9	7.7±1.4	**22.4±7.5**	**53.4±2.7**
	TF	-	19.2±1.6	22.0±2.0	23.9±1.6	28.4±0.9	32.6±0.9	51.2±2.3
		INT	19.9±3.5	24.1±2.1	25.9±2.6	**33.5±2.6**	31.4±4.4	51.3±1.6
		ASM	19.6±2.0	**24.7±3.9**	26.0±1.3	29.1±1.3	32.6±1.6	49.3±0.8
		IMP	**21.5±1.5**	24.4±0.5	**29.1±1.8**	30.9±1.0	33.5±3.6	51.4±2.9
		ALL	19.2±2.1	21.3±1.7	28.0±2.8	30.8±3.0	**37.5±0.8**	**53.2±1.2**

Table 3: Performance of different models on the **Friends** dataset for the task of ERC for both in-domain (ID) and transfer (TF) setting across different few-shot splits (5, 10, 20, 50, 100) and the entire dataset (denoted by "All"). The different rationales explored in this work are denoted by only utterance (-), utterance with speaker's intention (INT), utterance with the hearer's assumption (ASM), utterance with implicit information (IMP), and utterance with all the aforementioned rationales included i.e. INT, ASM, and IMP, and is denoted by ALL.

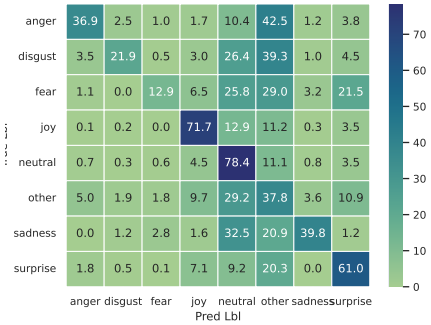
Model	Mode	Rationale	5	10	20	50	100	All
bert	ID	-	13.4±2.1	15.0±2.1	17.5±3.7	31.2±0.8	33.9±0.7	40.9±0.9
		INT	13.2±1.2	**19.2±2.6**	**26.9±5.8**	**34.5±3.0**	**39.9±1.8**	45.3±0.8
		ASM	11.5±5.0	16.4±3.4	18.6±3.8	30.2±0.8	35.4±1.3	44.6±0.1
		IMP	12.5±4.9	13.2±3.2	22.5±3.9	32.1±1.6	36.0±1.0	44.7±1.7
		ALL	5.8±3.7	16.9±3.4	23.8±5.3	33.6±2.0	37.7±1.0	**46.2±1.3**
	TF	-	22.3±1.1	24.7±0.8	26.3±2.1	29.2±2.0	31.6±1.6	41.0±1.3
		INT	24.2±2.0	**25.0±2.4**	**28.3±1.5**	**30.6±1.0**	32.6±1.1	44.9±0.4
		ASM	23.2±3.0	23.9±2.4	24.9±2.7	27.3±1.4	30.8±1.0	40.9±0.8
		IMP	21.4±1.2	24.2±1.5	25.1±1.6	28.1±0.9	31.5±1.5	45.0±0.6
		ALL	**25.3±2.2**	24.3±1.9	27.6±1.2	30.2±1.3	**33.1±1.0**	**46.1±1.8**
gpt2	ID	-	7.7±0.9	9.9±0.9	11.2±0.2	12.2±1.0	17.1±1.1	26.5±0.8
		INT	7.3±1.8	9.5±0.3	10.0±1.5	**23.6±2.1**	**28.4±3.2**	44.5±1.0
		ASM	5.7±0.8	7.6±0.5	10.0±1.4	14.0±1.3	20.6±3.4	43.4±1.2
		IMP	7.9±2.4	9.0±1.1	10.1±0.9	15.2±1.7	24.0±1.4	43.3±1.9
		ALL	7.9±1.1	8.8±0.4	10.6±3.6	18.5±1.3	27.1±1.6	**45.5±0.8**
	TF	-	14.2±1.2	14.6±0.2	14.9±0.9	15.9±1.0	17.9±1.4	26.5±1.3
		INT	**21.5±2.6**	22.0±1.1	**27.2±1.5**	**27.9±0.8**	30.8±1.5	43.7±1.7
		ASM	14.5±3.1	16.4±3.8	18.7±0.9	20.6±1.6	26.3±1.8	40.7±0.7
		IMP	16.9±2.3	16.9±3.3	20.6±1.6	23.2±1.5	27.7±2.5	42.6±1.1
		ALL	19.9±3.1	**22.5±1.5**	26.5±0.9	27.5±1.8	**31.0±2.9**	**45.4±1.1**
t5-base	ID	-	4.8±4.3	11.5±0.3	12.3±1.4	16.2±3.8	25.5±0.4	39.8±3.4
		INT	11.6±5.4	**20.1±1.5**	26.5±2.7	24.5±2.5	28.3±2.3	**44.8±2.6**
		ASM	11.3±1.5	13.4±1.3	18.5±2.6	20.0±2.3	23.2±2.8	39.8±0.6
		IMP	11.1±0.3	15.4±4.0	19.8±2.8	22.7±3.1	25.4±5.1	44.1±3.3
		ALL	**16.9±2.5**	20.0±1.1	**28.0±2.1**	**26.5±1.2**	**31.3±1.8**	43.8±3.1
	TF	-	19.0±0.5	20.4±1.3	21.4±1.7	26.1±2.5	**31.2±1.3**	40.3±2.9
		INT	24.5±2.4	**26.1±2.7**	27.8±2.6	29.9±1.2	30.9±1.3	42.6±2.9
		ASM	19.7±2.0	22.6±2.4	23.0±1.0	26.2±0.9	29.2±1.3	44.6±2.3
		IMP	20.6±1.4	22.8±1.0	24.7±1.2	28.2±1.3	30.2±1.7	47.2±0.4
		ALL	**24.8±2.3**	25.0±1.1	**28.0±1.5**	**30.0±0.9**	30.7±0.7	**47.4±0.7**

Table 4: Performance of different models on the **IEMOCAP** dataset for the task of ERC for both in-domain (ID) and transfer (TF) setting across different few-shot splits (5, 10, 20, 50, 100) and the entire dataset (denoted by "All"). The different rationales explored in this work are denoted by only utterance (-), utterance with speaker's intention (INT), utterance with the hearer's assumption (ASM), utterance with implicit information (IMP), and utterance with all the aforementioned rationales included i.e. INT, ASM, and IMP, and is denoted by ALL.

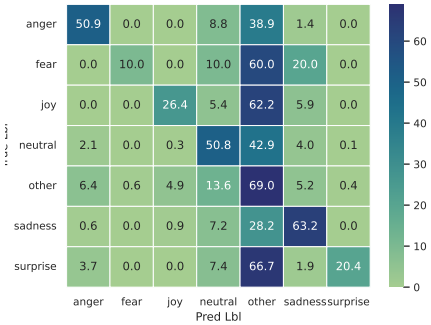
Model	Mode	Rationale	5	10	20	50	100	all
bert	ID	-	13.7±7.2	16.1±3.1	24.3±2.7	33.0±1.4	36.1±1.1	40.7±1.5
		INT	11.6±4.6	14.8±0.7	23.2±1.2	35.0±1.5	**38.3±1.5**	**42.6±1.3**
		ASM	10.8±5.2	**19.6±2.7**	22.0±1.4	32.8±1.5	35.8±4.3	41.0±1.8
		IMP	13.3±1.7	14.4±5.6	**25.2±1.6**	32.2±3.4	36.3±2.7	42.0±1.2
		ALL	12.1±5.4	15.7±2.8	25.0±1.4	**35.5±2.6**	37.6±1.5	40.4±1.0
	TF	-	23.6±1.9	23.8±3.4	24.0±2.4	27.1±1.0	29.5±0.4	37.1±0.8
		INT	22.8±2.2	23.6±1.8	24.3±1.3	**29.0±2.4**	30.4±0.9	**43.4±1.5**
		ASM	23.8±1.0	24.2±0.5	24.4±1.0	26.9±2.5	**32.5±4.0**	39.4±2.4
		IMP	25.0±1.0	24.6±1.7	**25.9±1.3**	27.0±1.4	29.9±0.3	42.1±0.9
		ALL	**25.4±0.4**	**25.0±1.8**	25.6±0.7	28.3±0.5	30.6±1.3	40.7±5.3
gpt2	ID	-	5.0±4.2	6.0±4.7	**10.6±2.2**	**17.1±2.2**	23.4±3.0	35.3±2.4
		INT	5.0±3.3	8.8±2.1	9.1±1.7	16.8±0.3	**27.5±1.1**	**42.5±2.4**
		ASM	6.2±1.7	8.5±2.9	9.7±1.9	16.4±1.7	25.1±2.3	39.3±3.2
		IMP	5.4±1.5	7.6±1.4	9.6±0.7	15.3±3.1	24.9±3.4	39.9±0.9
		ALL	5.6±3.2	8.3±2.2	9.0±1.6	15.2±0.5	24.3±1.8	39.7±1.8
	TF	-	17.0±1.1	17.0±0.8	19.6±0.9	23.1±2.0	26.4±0.7	36.0±0.8
		INT	**20.3±1.5**	20.6±1.0	22.5±1.4	24.8±1.7	**28.1±0.1**	41.0±3.4
		ASM	19.3±0.0	20.8±1.1	22.5±1.2	**25.8±0.5**	27.3±1.7	40.0±0.4
		IMP	20.2±1.4	20.5±2.4	21.4±0.3	24.8±1.0	27.5±1.1	40.1±2.8
		ALL	19.9±1.8	**22.1±0.7**	**22.9±1.2**	25.5±1.2	27.1±1.6	**41.9±1.4**
t5-base	ID	-	5.3±4.6	8.0±4.0	10.0±2.1	21.1±2.6	26.9±0.6	42.8±1.7
		INT	7.1±0.2	8.7±4.0	15.1±0.9	18.6±1.8	26.3±3.1	**45.0±0.7**
		ASM	8.9±2.2	8.1±0.8	10.3±5.6	20.3±1.1	**28.6±0.2**	43.1±0.6
		IMP	6.3±1.3	**13.5±2.1**	18.3±2.7	**22.8±1.7**	**28.6±1.8**	42.0±0.8
		ALL	6.5±3.8	12.9±3.2	**18.4±1.7**	22.0±2.0	24.8±0.8	44.2±1.2
	TF	-	19.8±0.7	20.6±0.2	20.6±1.0	24.6±2.1	27.8±1.0	41.7±1.0
		INT	22.3±0.7	22.5±0.9	22.3±0.6	**27.6±0.6**	**31.6±1.7**	**43.9±0.5**
		ASM	21.0±0.6	21.3±1.2	21.6±1.1	25.4±1.2	28.2±1.0	43.5±0.6
		IMP	22.4±1.0	21.9±0.5	22.5±1.3	25.4±1.1	27.9±3.7	40.5±2.6
		ALL	**23.1±0.5**	**23.2±0.3**	**23.2±0.5**	27.1±1.9	29.0±1.0	**43.9±1.6**



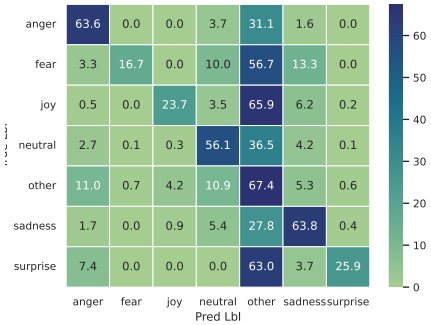
(a) friends with UTT (BERT)



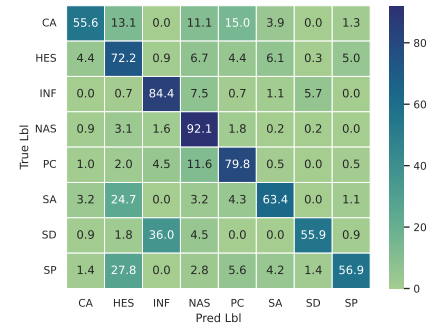
(b) friends with ALL (BERT)



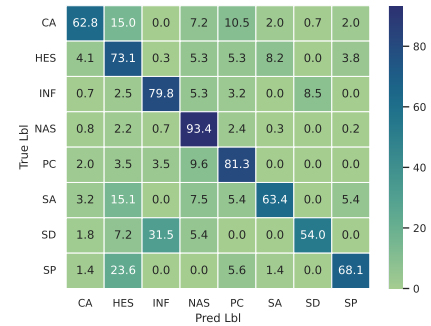
(c) iemocap with UTT (T5)



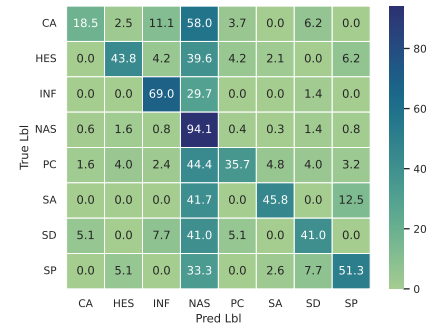
(d) iemocap with INT (T5)



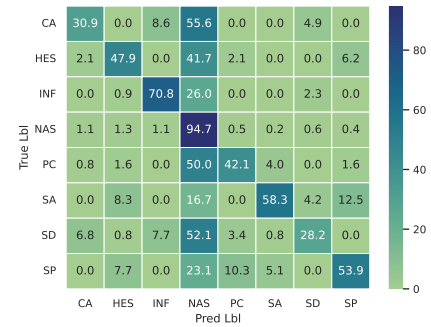
(e) CB with UTT (T5)



(f) CB with ALL (T5)

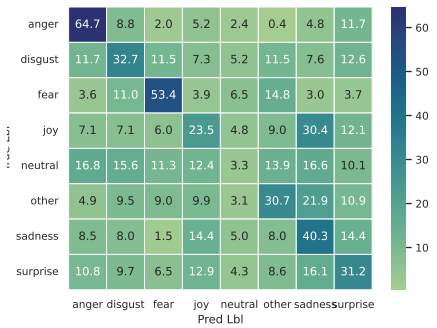


(g) P4G with UTT (T5)

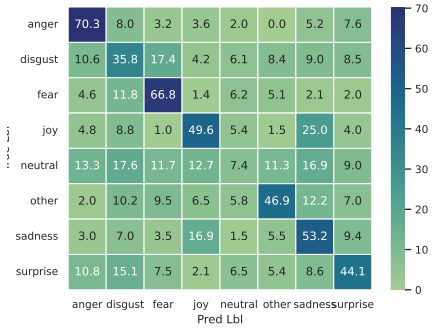


(h) P4G with ALL (T5)

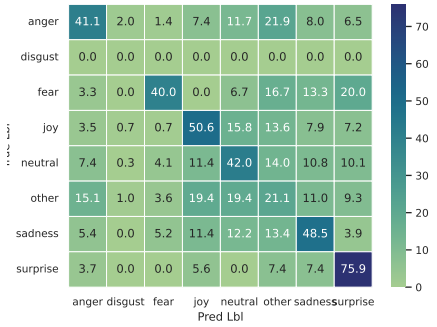
Figure 3: We present here the confusion matrices of the best performing pair of models and rationales in the in-domain setting for the 4 datasets and the corresponding model in absence of any rationale (UTT) in the in-domain setting (ID)



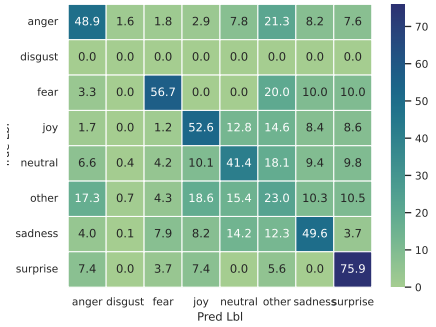
(a) friends with UTT (T5)



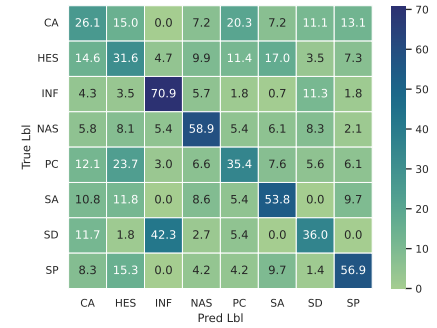
(b) friends with ALL (T5)



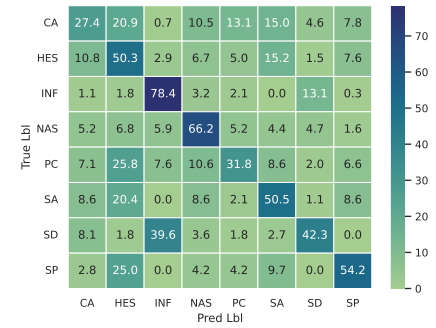
(c) iemocap with UTT (BERT)



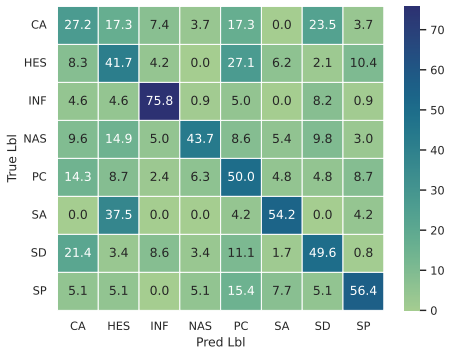
(d) iemocap with INT (BERT)



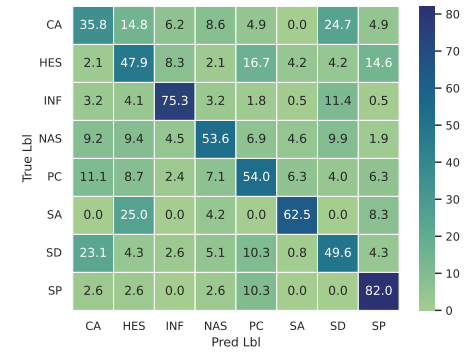
(e) CB with UTT (BERT)



(f) CB with ALL (BERT)



(g) P4G with UTT (BERT)



(h) P4G with INT (BERT)

Figure 4: We present here the confusion matrices of the best performing pair of models and rationales in the transfer setting at k=20-shot case for the 4 datasets and the corresponding model in absence of any rationale (UTT).

Appendix B

Additional Experiments and Results

Supervised Full Fine-tuning Setup

Generator	Rationale	P4G	CaSiNo	res_CB	PROP	EMH	IMP_HATE
GPT-4o	UTT	69.70 +/- 2.42	71.22 +/- 1.70	66.77 +/- 1.02	82.38 +/- 1.21	**90.91 +/- 0.13**	62.68 +/- 0.79
	+ INT	69.36 +/- 1.45	**72.35 +/- 0.50**	**70.91 +/- 0.71**	**84.66 +/- 1.07**	89.35 +/- 1.35	**67.91 +/- 1.49**
	+ HR	**70.54 +/- 1.70**	71.71 +/- 0.84	68.80 +/- 0.97	82.88 +/- 1.69	90.26 +/- 0.32	65.08 +/- 0.34
	+ PreSup	68.12 +/- 2.30	71.81 +/- 1.39	69.69 +/- 1.51	80.11 +/- 2.86	89.37 +/- 0.16	62.88 +/- 2.55
GPT-3.5-turbo	UTT	69.70 +/- 2.42	71.22 +/- 1.70	66.77 +/- 1.02	82.38 +/- 1.21	**90.91 +/- 0.13**	62.68 +/- 0.79
	+ INT	67.64 +/- 3.16	**72.35 +/- 0.38**	**71.22 +/- 3.03**	81.52 +/- 1.47	90.01 +/- 1.12	62.82 +/- 0.62
	+ HR	68.90 +/- 1.54	71.95 +/- 2.67	70.87 +/- 1.17	**83.61 +/- 2.00**	89.18 +/- 0.73	**64.16 +/- 0.97**
	+ PreSup	**72.21 +/- 0.25**	70.43 +/- 1.27	69.28 +/- 1.45	78.61 +/- 2.97	90.00 +/- 0.96	59.85 +/- 0.52

Table 5: Performance of FLAN-T5 model in an in-domain setting across six tasks. The baseline includes only the utterance (UTT), which we compare against the three kinds of rationales, i.e. intentions (INT), hearer-reactions (HR), and presuppositions (PreSup). We represent the mean and standard deviation across three runs.

Indomain Results: We present additional results of our supervised instruction-tuning experiments in this section. Table 5 showcases the impact of adding rationales i.e. the intentions, hearer reactions, and presuppositions, generated by GPT-4o and GPT-3.5-turbo LLMs on the six datasets using the FLAN-T5 model. We see that apart from the EMH dataset, adding in the rationale improves performance for a majority of the cases.

Cross Task Results: We present the results of our cross task transfer experiments using the FLAN-T5-base model augmented with rationales generated by GPT-4o and GPT-3.5-turbo in Figures 5 and 6 respectively. We observe significant gains over the utterance (or the baseline case) when rationales are added for different datasets and few-shot settings.

We also inspect which category of rationales are the most effective for a given source and target pair in Figures 7 and 8 respectively by the net relative improvement in F1 score across different few-shot settings. We observe that for the intentions rationales, transfer almost always yields a positive relative improvement for any source and target pair, showcasing their effectiveness across different tasks. After intentions, we observe that the hearer reactions have the most impact followed by presuppositions.

Figure 5: Impact of rationales on cross-task performance for instruction-tuned models across the six datasets for different fewshot settings using the GPT-4o generated rationales.

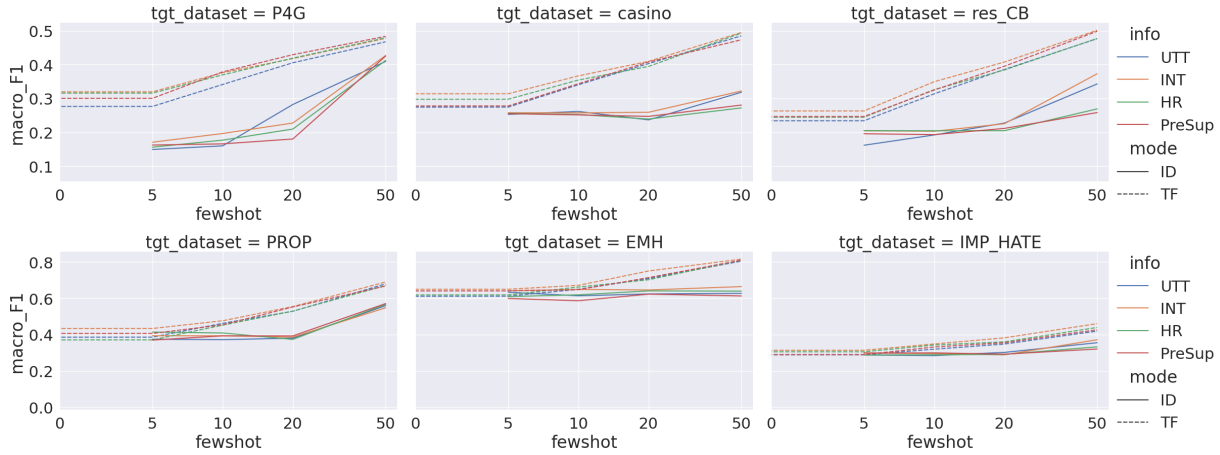


Figure 6: Impact of rationales on cross-task performance for instruction-tuned models across the six datasets for different fewshot settings using the GPT-3.5-turbo generated rationales.

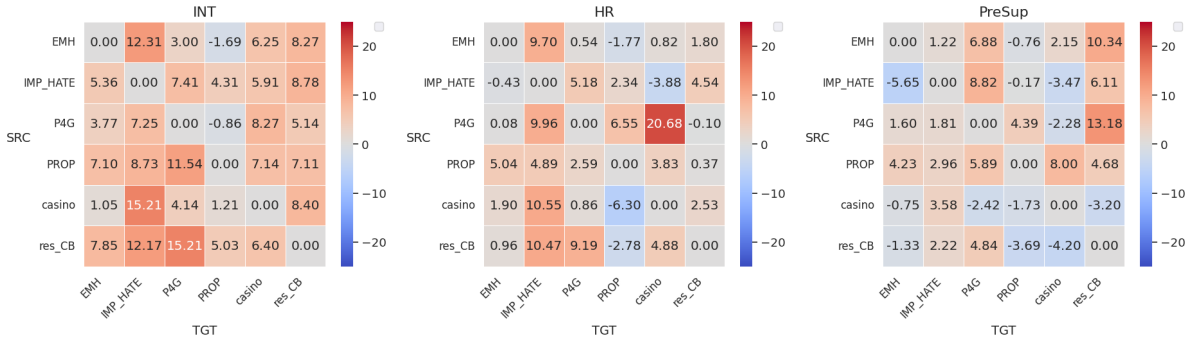


Figure 7: Relative change in performance measured in terms of F1 score over the baseline when incorporating the GPT-4o generated rationales for different source and target pairs for the cross-task transfer setting.

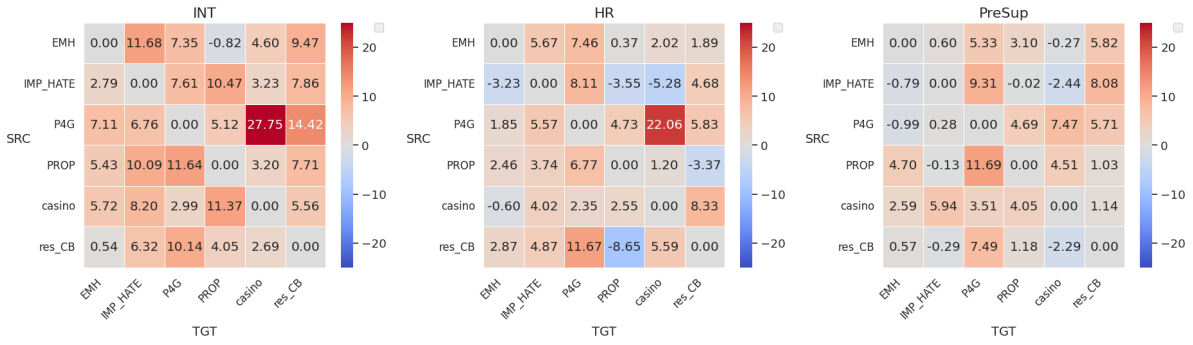


Figure 8: Relative change in performance measured in terms of F1 score over the baseline when incorporating the GPT-3.5-turbo generated rationales for different source and target pairs for the cross-task transfer setting.

Table 6: Spearmann's rank correlation between model lists for the source and target.

Dataset	Instances	Rationales
P4G	-0.04	0.46
CaSiNo	-0.07	0.00
res_CB	0.01	0.06
PROP	0.15	0.18
EMH	-0.02	-0.15
IMP_HATE	0.07	0.28

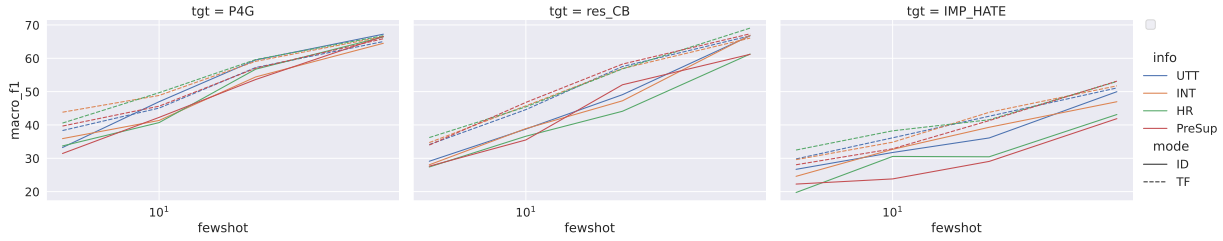


Figure 9: Impact of rationales on both in-domain and cross-task performance for PEFT-based LLama models across the three datasets for different few-shot settings. We use the rationales generated by GPT-4o

Table 7: Performance of our FLAN-T5 model against previous SOTA performance.

Dataset	FLAN-T5	Repoted SOTA
P4G	**69.7**	59.6
res_CB	66.8	66.2
CaSiNo	**71.2**	68.3
PROP	**83.4**	72.1
EMH	**90.9**	69.9
IMP_HATE	**62.7**	58.6

PEFT-based Fine-tuning Setup

We also explore the impact of adding rationales in a PEFT-based fine-tuning setup. We fine-tune a pre-trained LLama-3-8B-it (AI@Meta, 2024) with 4-bit double quantization and low-rank adapter (LoRA) to ensure efficient fine-tuning (Hu et al., 2021b; Dettmers et al., 2024).

Due to the limited compute budget and the large number of experiments (360 in-domain and 1440 cross-task transfer runs) for a single SFT model, we experiment on only three out of six datasets, i.e. P4G, res_CB, and IMP_HATE. We chose these three datasets because they had the lowest performance in the in-domain setting.

We present the in-domain results in Table 9.7 and the cross-task transfer performance in Figure 9. We observe trends similar to our instruction-tuned results, i.e. rationales aid dialogue understanding and generalization for PEFT based models.

Performance against SOTA baselines

We compare the performance of our baseline, i.e. FLAN-T5 in the in-domain setting without any rationale information, against the previous reported SOTA performance (which were mostly trained on BERT based models) on all datasets as reported in their original paper. It is evident from Table 7 that our FLAN-T5 serves as a competitive baseline and achieves higher performance (in terms of macro F1 score) on all six tasks.

ICL Results

We note the effect of adding rationales for different in-context learning settings. We experiment with LLama-3-8B-it and Gemma-2-9B-it as the ICL LLM, and prompt them for different few-shot settings, i.e. 0-shot, 2-shot, and 5-shot. We present these results in Table 8 where we observe that adding rationales generally yielded higher performance over the baseline (i.e. using only the UTT). We observe that performance mostly plateaus at the 2-shot setting.

We also explore the impact of adding rationales generated by different LLMs, i.e. GPT-4o, GPT-3.5-turbo, and Gemma-2-27B-it, in Table 9 and note similar performance in all three cases, highlighting that the rationales generated by open-source models aid downstream task performance similar to proprietary models.

Table 8: Performance for in-context learning models for different datasets and few-shot settings aggregated over different rationale categories generated by different LLMs, i.e. GPT-4o, GPT-3.5-turbo, and Gemma-2-27B-it.

		Gemma-2-9B-it						LLama-3-8B-it					
Rationale	#fshot	P4G	casino	res_CB	PROP	EMH	HATE	P4G	casino	res_CB	PROP	EMH	HATE
UTT	0	30.23	36.87	33.92	46.36	51.33	35.16	20.55	30.08	26.44	44.28	**67.72**	32.05
+ INT	0	**32.95**	38.08	**35.31**	**49.23**	54.47	**36.46**	**21.23**	30.62	28.46	**46.63**	67.4	32.54
+ HR	0	28.27	36.27	32.7	43.69	55.26	35.02	20.64	30.5	**28.82**	46.11	65.24	32.18
+ PreSup	0	30.41	38.34	35.15	43.34	50.16	35.17	20.1	**30.99**	26.98	43.94	67.25	32.33
+ ALL	0	32.78	**40.4**	34.23	46.73	**55.75**	35.89	21.06	29.36	27.43	44.89	67.57	**32.73**
UTT	2	36.24	38.69	39.67	46.72	60.82	35	**24.64**	30.96	29.17	41.85	64.73	**30.74**
+ INT	2	37.85	**39.75**	**45.01**	49.06	66.39	**37.43**	21.87	**33.17**	**33.92**	**44.07**	65.61	30.58
+ HR	2	37.86	38.89	39.56	47.87	61.31	33.63	22.5	30.54	30.98	41.75	64.37	29.75
+ PreSup	2	36.21	37.61	41.58	48.24	58.7	36.31	24.11	30.93	30.4	41.82	61.59	29.35
+ ALL	2	**38.48**	39.4	43.38	**49.69**	**66.77**	36.61	21.72	31.1	30.56	42.32	**66**	29.91
UTT	5	37.72	39.33	38.23	46.2	60.51	35.66	20.59	29.12	27.91	41.81	**66.58**	29.58
+ INT	5	37.3	**39.96**	**43.23**	49.8	63.42	37.19	19.52	29.41	**32.64**	**43.44**	64.87	29.57
+ HR	5	**38.02**	39.57	38.67	48.91	61.4	35.16	**20.81**	**29.82**	31.42	44	65.42	29.57
+ PreSup	5	36.11	37.6	39.39	46.86	64.55	34.65	20.36	29.25	32.29	43.29	63.75	29.57
+ ALL	5	36.34	36.9	40.67	**53.26**	**66.03**	**37.36**	19.31	29.38	29.88	43.43	63.54	29.57

Table 9: Performance for in-context learning models for different datasets and few-shot settings aggregated over different few-shot settings.

		Gemma-2-9B-it						LLama-3-8B-it					
Rationale	LLM	P4G	casino	res_CB	PROP	EMH	HATE	P4G	casino	res_CB	PROP	EMH	HATE
UTT	-	34.73	38.3	37.27	46.42	57.55	35.27	21.93	30.05	27.84	42.65	66.34	30.79
+ INT	gpt-4o	35.31	**40.28**	**41.88**	**48.91**	62.35	**37.57**	21.56	**31.55**	30.89	**45.16**	**66.88**	**30.98**
+ HR	gpt-4o	32.91	37.33	37.92	44.73	61.62	35.56	**22.30**	30.03	31.16	43.33	66.54	30.69
+ PreSup	gpt-4o	35.13	37.89	39.48	47.65	58.79	36.63	21.3	31.1	**31.52**	44.23	65.38	30.62
+ ALL	gpt-4o	**36.75**	39.52	39.18	47.19	**62.50**	36.93	21.11	29.93	29.77	43.5	66.91	30.51
+ INT	gpt-3.5	**36.48**	**39.74**	**41.06**	**51.84**	62.52	**36.65**	19.92	**31.26**	**32.89**	**44.87**	64.46	31.18
+ HR	gpt-3.5	35.96	35.72	35.85	48.08	57.23	34.45	19.82	30.38	30.43	43.43	65	30.32
+ PreSup	gpt-3.5	32.88	38.5	38.87	45.25	56.59	34.7	20.8	29.45	29.2	41.66	63.18	30.48
+ ALL	gpt-3.5	34.84	37.9	39.3	50.78	**63.75**	36.01	19.1	29.48	28.27	43.72	64.99	**31.27**
+ INT	Gemma	**36.32**	37.78	**40.61**	47.35	59.42	36.85	21.14	30.39	**31.24**	44.1	**66.54**	30.52
+ HR	Gemma	35.28	**41.68**	37.16	47.67	59.12	33.8	21.83	30.45	29.63	**45.1**	63.48	30.49
+ PreSup	Gemma	34.72	37.16	37.78	45.54	58.03	34.8	**22.48**	**30.61**	28.96	43.16	64.04	30.16
+ ALL	Gemma	36.02	39.29	39.81	**51.71**	**62.31**	**36.91**	21.88	30.43	29.83	43.41	65.21	30.44

Table 10: In-context learning performance of different LLMs (Gemma-2-9B-it and Llama-3-8B-it) with the best rationale of each category (i.e. INT, HR, PreSup, and ALL) against the Chain-of-Thought (CoT) prompting setting.

	Gemma-2-9B-it						Llama-3-8B-it					
RAT	P4G	CaSiNo	res_CB	PROP	EMH	HATE	P4G	CaSiNo	res_CB	PROP	EMH	HATE
UTT	29.24	35.88	33.84	47.62	48.84	33.9	20.11	30.04	26.88	43.95	66.84	32.42
+ COT	33.78	38.66	34.27	**58.08**	**61.99**	32.66	21.36	**33.61**	27.92	**48.64**	50.92	32.03
+ INT	**34.78**	39.04	**35.99**	50.98	57.49	**38.79**	**21.66**	31.32	**29.49**	47.25	67.15	32.74
+ HR	27.98	38.84	35.07	44.25	56.63	36.65	21.25	31.77	29.00	45.70	67.06	32.93
+ PreSup	32.32	40.51	38.17	45.87	53.33	38.25	20.39	32.16	28.24	45.55	67.69	33.39
+ ALL	33.74	**40.60**	33.99	46.81	56.93	37.14	21.15	29.53	27.92	46.25	**69.59**	**34.11**

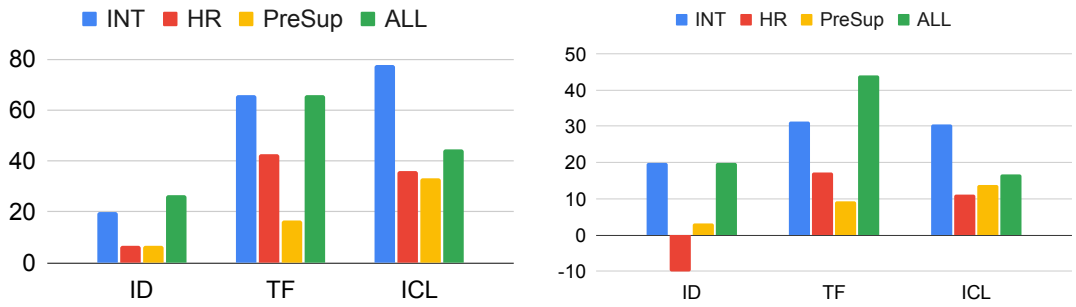


Figure 10: Proportion of cases adding rationales improve performance overall (left) and significantly (right) for different settings

Table 11: Hyperparameters used for fine-tuning the FLAN-T5-base model for all the experiments.

Hyperparameter	**Value**
SFT- Instruction Tuned Setup	
Max sequence length	1024
Learning rate	$2e^{-5}$
Batch size	8
Num. epochs	10
Optimizer	Adam
Patience	5
Seeds	3
Model	`FLAN-T5-base`
SFT- PEFT Setup	
Quantization	4-bit double
Precision training	bf16
LoRA reduction factor	64
LoRA dropout	0.05
LoRA Alpha	32
Batch size	4
Weight decay	0.01
Learning Rate	$2e^{-5}$
Max sequence length	1024
Num. epochs	10
Patience	5
Seeds	3
LLM	`LLama-3-8B-it`
ICL	
Temperature	0.9
Fewshot examples	[0, 2, 5]
Batch size	8
GPUs	A6000 *2

Table 12: Versions of Library used in our work.

SFT + ICL setup	
Libraries	**Version**
Python	3.9.12
torch	1.12.1+cu113
transformers	4.40.2
numpy	1.24.2
sklearn	1.2.2
PEFT setup	
Libraries	**Version**
sentence-transformers	2.7.0
flash_attn	2.7.4.post1
huggingface-hub	0.30.2
numpy	2.2.4
transformers	4.51.3
peft	0.10.0
bitsandbytes	0.45.5
accelerate	1.5.2
evaluate	0.4.3
scikit-learn	1.6.1
tokenizers	0.21.1
torch	2.5.1

Experimental Details and Hyper-Parameter Tuning

We present the hyperparameters for our experiments in Table 11. We carry out the experiments over 3 seeds on a A6000 GPU with early stopping with patience of 5 over the validation set for all experiments. We implement the entire experiments in Python, with help of the Pytorch library and use the pre-trained models as specified in Huggingface under the agreed upon license agreements. We explicitly specify the software libraries and their corresponding versions in Table 12

Our experimental suite comprises encompasses 6 datasets in the indomain setting for the FLAN-T5 models for 5 few-shot settings (5, 10, 20, 50, and all) across 3 seeds and for 9 cases, corresponding to the 3 types of rationales individually (INT, HR, PreSup) and combined (i.e. ALL), for each of the two LLMs (GPT-3.5-turbo and GPT-4o) and the baseline (UTT). Furthermore, for a model pre-trained on a given source task, we further fine-tune it for 4 k-shot settings (5, 10, 20, and 50) for each of the 5 different target tasks. This results in a massive experimental suite of 810 in-domain experiments and 4050 cross-task experiments.

For our in-context learning setting, we experiment with instruct-tuned versions of two open-sourced models, i.e. LLama-3-8B and the Gemma-9B. To account for prompt sensitivity, the

prompts used for inference were first validated on the development split for each of the 6 datasets. We use rationales generated by both proprietary and open-sourced LLMs, i.e. GPT-4o, GPT-3.5-turbo, and Gemma-2-27B-it. Our experiments thus comprise 5 different kinds of rationales (i.e. None, INT, HR, PreSup, and ALL), 2 LLMs for doing ICL, 3 LLMs that generate the rationales, 3 few-shot settings, for the 6 datasets resulting in an additional 540 experiments.

The total cost of the OpenAI credits during the course of our experiments to generate the rationales was approximately USD 265 USD, with the cost of the GPT-4o model being approximately 10 times as costly as the GPT-3.5-turbo version.

Characteristics of the Generated Rationales

Rationale Similarity

We measure the similarity of the generated rationales across three fronts:

(i) How similar are the three different categories of rationales to each other?

(ii) How similar are the rationales generated by different LLMs for the same rationale category?

(iii) How similar is a generated rationale to its corresponding utterance?

We use cosine distance between the sentential representations as the metric for quantifying similarity. We explore two models to generate these representations, i.e., the popular MPNET model of (Reimers and Gurevych, 2019) for its simplicity and the instruction-tuned version of Mistral-7B (Wang et al., 2023a) for its superior performance on the MTEB leaderboard (Muennighoff et al., 2023). We present the similarity scores across different LLMs, different rationale categories, and between the utterance and the rationale in Figures 10, 9.6, and 9.5 respectively.

Instance-wise Performance

We investigate several factors that could predict the performance of rationales on an instance-wise basis. The covariates observed, i.e. the factors include (i) the length of the rationale, (ii) the length of the preceding dialogue history, (iii) the similarity between the rationale and the utterance, (iv) the similarity between the rationale and the label description being classified, (v) the readability score measured using the Flesch'as readability ease (Farr et al., 1951; Kincaid, 1975), (vi) the valence, arousal, and dominance scores measured via the VAD NRC lexicon (Mohammad, 2018), and (vii) scores corresponding emotional intensity, emotional polarity and empathy (Wu et al., 2024). The correlation between each of the factors and instance-wise task performance is highlighted in Table 13.

Generalization Characteristics

We inspect the factors that characterize generalizability over the different experimental settings using the different rationales. We perform a multivariate ANOVA analysis with the absolute performance difference over the baseline as the dependent variable. The independent variables chosen were the rationale category, the LLM used to generate the rationales, the choice of the

Table 13: Correlation of different rationale characteristiscs with classification accuracy. We explore intentions, hearer reactions, and presuppositions for in-domain, cross-task transfer, and in-context learning settings.

	In-domain			Cross-task Transfer			In-context Learning		
Factor	INT	HR	PreSup	INT	HR	PreSup	INT	HR	PreSup
#RAT Length	-0.07	-0.05	-0.06	-0.09	-0.07	-0.07	-0.15	-0.15	-0.13
# Dial Length	0.05	0.05	0.05	0.09	0.09	0.09	0.08	0.10	0.10
LBL Sim	-0.06	-0.06	-0.04	-0.07	-0.07	-0.04	-0.08	-0.11	-0.05
UTT Sim	-0.02	0.02	-0.02	-0.02	0.01	-0.02	-0.09	-0.01	-0.07
Valence	0.02	0.06	0.04	0.07	0.07	0.04	0.08	0.13	0.07
Arousal	-0.01	-0.01	0.00	-0.04	-0.03	-0.01	-0.08	-0.06	-0.04
Dominance	0.01	0.05	0.03	0.04	0.05	0.01	-0.01	0.10	0.01
Emo Intensity	-0.01	-0.04	-0.02	-0.05	-0.05	-0.02	-0.09	-0.14	-0.07
Emo Polarity	-0.01	-0.04	-0.02	-0.05	-0.05	-0.02	-0.09	-0.14	-0.07
Empathy	-0.01	-0.04	-0.02	-0.05	-0.05	-0.02	-0.09	-0.14	-0.07
Flesch's Readability	0.02	0.03	0.02	0.00	0.04	0.00	0.03	0.08	0.02

source and target dataset [1], and the few-shot setting; we also consider the effects of pairwise interaction of each of these variables. We note the F-statistic and their corresponding p-value for the indomain, cross-task and incontext-learning setting respectively in Tables 14, 15, and 16 in the Appendix 2.3.

For the in-domain setting, the performance change hinges on the rationale category, the number of few-shot examples, the target dataset, and also their pairwise interactions. We also observe mild significant pairwise effects between the LLM and the rationale category. A similar trend emerges during cross-task transfer; the rationale category, the target dataset, and the number of few-shot examples play a significant effect in influencing performance. However, the choice of the source dataset is significant only when we consider its pairwise interaction with the other covariates. The story differs slightly for the ICL setup; the choice of the dataset, the rationale, and the LLM (but not the few-shot setting) significantly impact performance.

Ablation Results

Importance of the utterance information

We carry out ablation studies to investigate the role of the utterance on task performance i.e. how does the performance vary when we omit out the utterance and evaluate the fine-tuned model using only the rationale. We explore two settings: (i) where the model is provided with both the utterance and rationale information during training, but use only the rationale during inference, (see Figures 12) and (ii) where we train and test the model with only the rationale as an

[1]For the indomain setting we consider only the target dataset

Table 14: The F-statistics and corresponding p-value for the multi-variate ANOVA analysis to investigate the factors that characterize the performance difference in an indomain setting for SFT setup.

Category	F-statistic	p-value
C(LLM)	0.363057	5.47E-01
C(RAT)	21.073603	**1.69E-09**
C(Dataset)	5.252105	**1.05E-04**
C(fewshot)	11.699875	**4.50E-09**
C(Dataset):C(LLM)	1.642512	1.47E-01
C(RAT):C(Dataset)	2.680245	**3.36E-03**
C(LLM):C(RAT)	3.627177	**2.73E-02**
C(fewshot):C(LLM)	0.566543	6.87E-01
C(RAT):C(fewshot)	4.213318	**6.76E-05**
C(fewshot):C(Dataset)	10.810497	**4.69E-28**

Table 15: The F-statistics and corresponding p-value for the multi-variate ANOVA analysis to investigate the factors that characterize the performance difference in a cross-task transfer setting for SFT setup.

Category	F-statistic	p-value
C(LLM)	2.350972	1.25E-01
C(RAT)	31.459235	**3.17E-14**
C(fewshot)	2.599193	**3.45E-02**
C(src_dataset)	1.806214	1.25E-01
C(tgt_dataset)	5.282518	**3.09E-04**
C(LLM):C(RAT)	3.847212	**2.15E-02**
C(LLM):C(fewshot)	1.138982	3.36E-01
C(LLM):C(src_dataset)	2.245978	**4.73E-02**
C(LLM):C(tgt_dataset)	3.028266	**9.92E-03**
C(fewshot):C(RAT)	1.161916	3.18E-01
C(src_dataset):C(fewshot)	4.966472	**3.11E-12**
C(fewshot):C(tgt_dataset)	4.083211	**3.01E-09**
C(RAT):C(src_dataset)	2.137128	**1.90E-02**
C(RAT):C(tgt_dataset)	2.86715	**1.47E-03**
C(src_dataset):C(tgt_dataset)	3.242511	**1.52E-06**

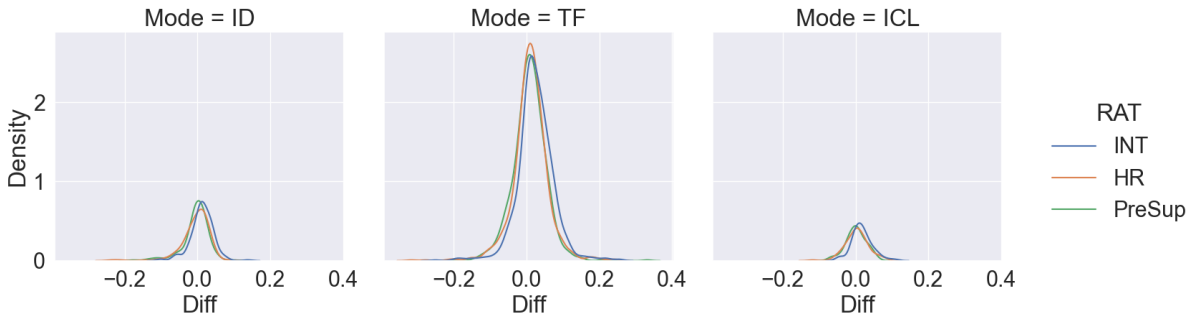


Figure 11: Distribution of the net performance difference across the three different settings, i.e. in-domain (ID), cross-task transfer (TF), and in-context learning (ICL) for the three rationales, i.e. intentions (INT), hearer reactions (HR), and presuppositions (PreSup).

augmentation (see Figure 13).

We observe a noticeable degradation in performance compared to the baseline (the model is trained only on the utterance) in the former case for both the indomain and cross-task setting; the drop progressively increases with the amount of training data, highlighting that fine-tuned models do not solely rely on the rationale to make its predictions. The latter scenario where the model is fine-tuned with only the rationales fares better, albeit still falling short of the baseline in the in-domain setting. When trained on only the rationale information, the impact of the rationale category on the task performance becomes more pronounced. We see higher gains from adding the hearer reactions to P4G, the presuppositions to IMP_HATE, and the intentions to casino, and EMH. In the cross-task setting, the performance drop is almost negligible; in fact we see marked improvements for res_CB, IMP_HATE and EMH with the intention rationales over the baseline. In short, we see that the utterance information is crucial for task performance and though rationales provides a useful augmentation, they cannot be used as a replacement or substitute for the utterance. Future work needs to inspect how to design free-text rationales that can capture all the salient aspects of the utterance (Chen et al., 2023).

Perturbation of the Rationales

We also carry out sensitivity analysis of the rationales by observing how perturbing the rationale text affects task performance. We compare different kinds of perturbations such as synonym swap using Checklist (Ribeiro et al., 2020) and WordNet, different kinds of augmentations (EmbedDA), deletions or combination of them (EDA) (Wei and Zou, 2019). We also control for the fraction of words being perturbed in the rationale text i.e. 10%, 50% and 90%. We depict the change in task performance due to perturbations in Table 14

Overall, on a macro scale, we observe that perturbations indeed decrease task performance with the deterioration becoming more pronounced as the proportion of words being perturbed increases. We also note that certain methods are more effective than others such as deletion as opposed to synonym matching or entity replacement. Such an analysis highlights that the instruct-tuned model does rely on the rationales for classification.

Table 16: The F-statistics and corresponding p-value for the multi-variate ANOVA analysis to investigate the factors that characterize the performance difference in fewshot setting for in-context learning models.

Category	F-statistic	p-value
C(LLM)	5.202281	**6.10E-03**
C(RAT)	10.668473	**3.50E-05**
C(dataset)	7.535951	**1.00E-06**
C(fewshot)	0.356484	7.00E-01
C(model_name)	1.22807	2.69E-01
C(LLM):C(RAT)	1.561942	1.85E-01
C(LLM):C(dataset)	0.734409	6.92E-01
C(LLM):C(fewshot)	1.258991	2.87E-01
C(LLM):C(model_name)	0.831352	4.37E-01
C(RAT):C(dataset)	0.647286	7.72E-01
C(RAT):C(fewshot)	0.750312	5.59E-01
C(RAT):C(model_name)	2.665021	7.15E-02
C(dataset):C(fewshot)	2.14782	**2.15E-02**
C(dataset):C(model_name)	3.456222	**4.85E-03**
C(fewshot):C(model_name)	0.938185	3.93E-01

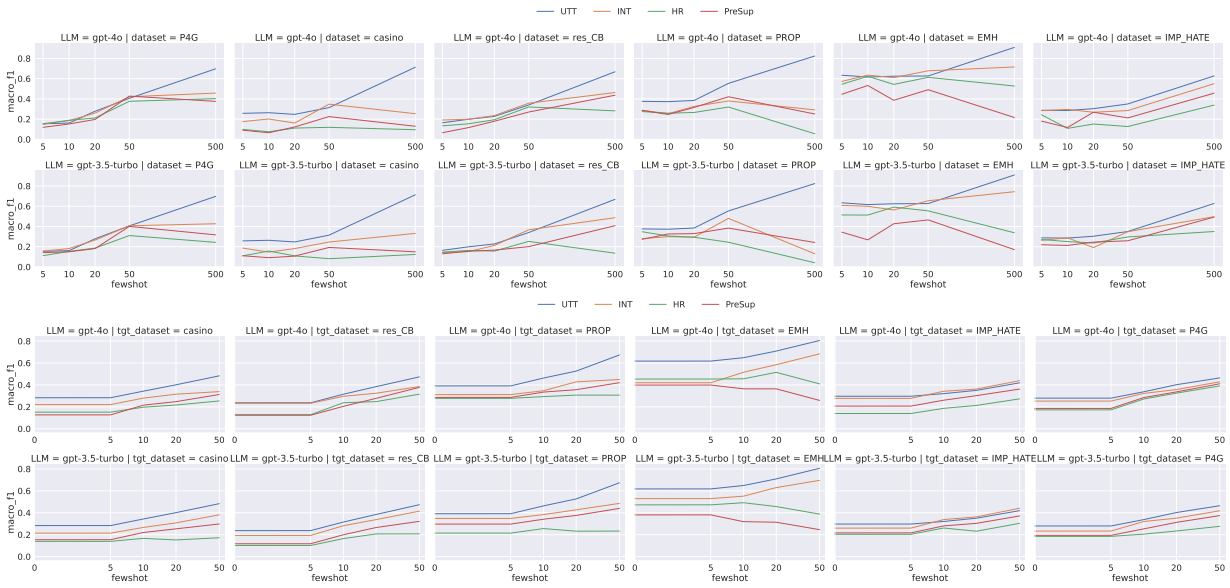


Figure 12: In-domain performance (top) and cross-task performance of models in presence of only the rationale across different few-shot cases. Note that the model was trained on BOTH the rationale and utterance.

Table 17: We present instances across different datasets where adding the rationale information was crucial in predicting the correct label always. We compute Shapley values for each token in the rationale to observe its contribution to the model's decision; the highlighted portions correspond to high positive associations with the label.

Dataset	Label	Utterance text	Rationale Text	CAT
casino	showing-empathy	Are you sure that's enough firewood for you and the baby? I know that babies can easily get very sick from dropping temperatures.	**Expressing concern about the** adequacy of **firewood for the baby**	INT
casino	vouch-fair	hey buddy I hope we both end up with a good deal:)	Expressing hope for a **mutually beneficial outcome**	INT
IMP_HATE	white_grievance	is zuma talking about seizing white land or not ? are you denying it ?	Seeking clarification and confirmation about Zuma's stance on **seizing white land**; challenging the respondent to deny it.	INT
IMP_HATE	irony	why do we have an immigration policy? should just let them have a free for all and all can pay!	Expressing frustration **and sarcasm** about the current immigration policy and suggesting an extreme alternative to highlight perceived issues.	INT
IMP_HATE	irony	mt : ""we must resist ebolaphobia. these viruses just come here for a better life, to do jobs that american viruses refuse to do.	The speaker is using **satire to comment** on the fear of Ebola (ebolaphobia) by comparing it to immigration issues suggesting that the fear is irrational and **mocking the idea that viruses have intentions** similar to human immigrants.	INT
IMP_HATE	white_grievance	but that wouldn't enable them to destroy white neighbourhoods .	There is a belief or concern that certain actions or policies could lead to **the destruction of white neighborhoods.**	PreSup
IMP_HATE	threatening	and they should have to pay every dime back, plus interest. from their own damn pockets.	People **should be held** financially accountable for their actions and **should not be able** to avoid paying back debts.	PreSup
P4G	credibility-appeal	As a donor I wanted you to see the site and the children that you would be helping	**EE is informed about the purpose of the link and the impact of their donation.**	HR
P4G	foot-in-the-door	Every little bit help.	EE feels reassured **that their small donation** is still valuable.	HR
P4G	foot-in-the-door	Every little bit help.	**Reassure the listener that any contribution is valuable.**	INT
P4G	foot-in-the-door	Your right, but I'm not asking for much.	**Minimizing** the financial impact of the donation	INT
res_CB	Source Derogation	Too be honest don't like the front bumper would be better without that black cover	**The seller might feel a need to address the buyer's concern about the bumper.**	HR
res_CB	Self Pity	at this i can only pay about 1600 could you do that	Seller **realizes the buyer's budget constraints.**	HR
res_CB	Source Derogation	Yes. What didn't your wife like about the bed?	Seller feels questioned about the **reason for selling the bed.**	HR

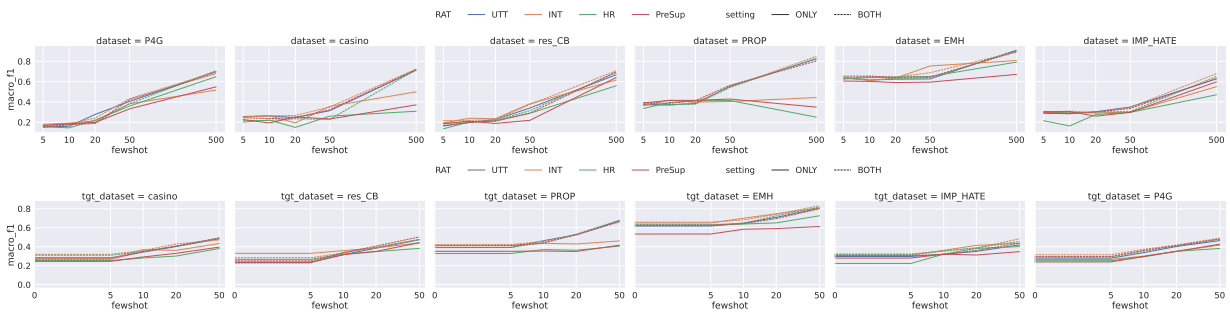


Figure 13: In-domain performance (top) and cross-task performance (below) of models using only the rationale across different few-shot cases. Note that the model was trained on ONLY the rationale.

Qualitative Analysis

We now carry out a qualitative analysis to investigate the specific instances where including the rationales actively improves the model's predictions in an indomain setting.

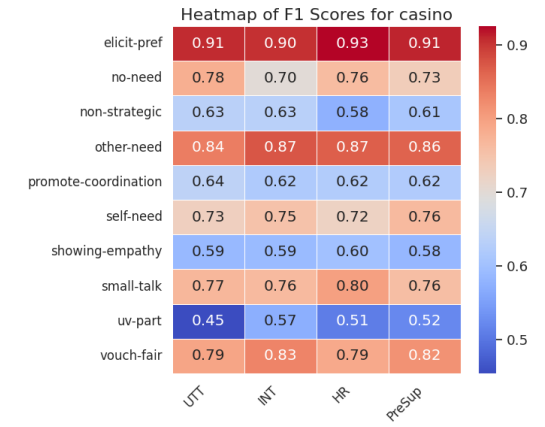
We depict the fraction of cases that benefit from adding rationales in the form of a Venn Diagram in Figure 17 in the Appendix. The overlapping areas indicate the fraction of instances that benefit from more than one types of rationale; for example, 10.0% of all instances benefit from all three rationales in CaSiNo. We consider only those instances where the baseline (i.e., only the utterance text) fails to predict the label correctly a majority of times, but succeeds when the rationale is provided.

The rationale with the greatest impact on performance is dependent on the nature of the task. The hearer reaction or HR has the highest impact on P4G, possibly because it captures the thought processes of the persuadee (EE) as they are being persuaded to donate. For example, the utterance "Anything would help even small donations add up when everyone pitches in." evokes a sense of reassurance from the persuadee (EE) that any contribution is valuable and is thus recognized as a "foot-in-the-door" strategy. Presuppositions are useful for IMP_HATE, a dataset that directly references stereotypes and thus requires generic knowledge to infer the type of implicit hatred. Tasks that are centered around the outcome the speaker is invested in, i.e. strategies employed to resist persuasion (res_CB), or signaling empathy to someone in therapy (EMH) benefit mostly from intentions. Furthermore, similar tasks e.g., CaSiNo and res_CB which deal with negotiation have similar relative performance for the same rationales.

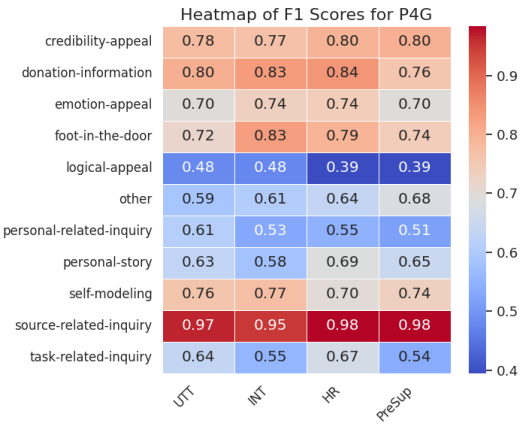
However, it should also be noted that a given rationale category does not serve as a silver bullet for all instances. We highlight some examples where model improvements were due to only one type of rationale in Table 17 in the Appendix and the possible reasoning for the same. While all three rationales are valid with respect to the utterance, we hypothesize that certain phrases or terms in the given generation might make it easier to predict the label category. For example, the phrase "feels questioned" in the HR hints at source derogation, which is not observed for the other rationales for the res_CB example. Likewise, the wording "how one might treat a dog" in the presupposition conveys the sense of inferiority more prominently than the generic idea of mistreatment in IMP_HATE. Since the rationales were not generated with a particular task in

Figure 14: Impact of different kinds of perturbation on the rationale text for classification performance.

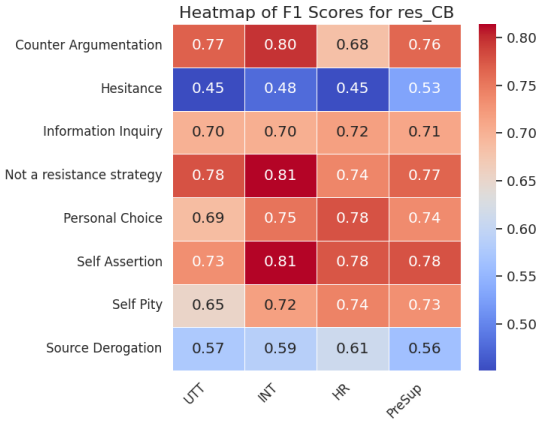
mind, the number of instances where the wording aligns with one of the task label's definition is also infrequent.



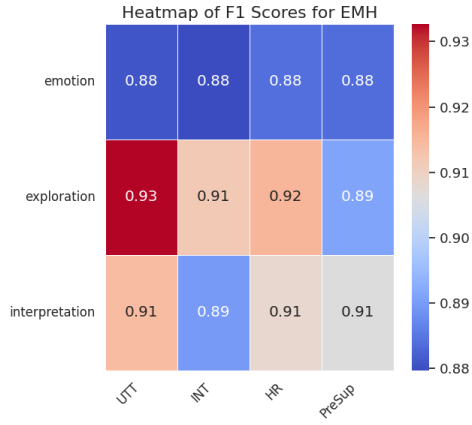
(a) Casino



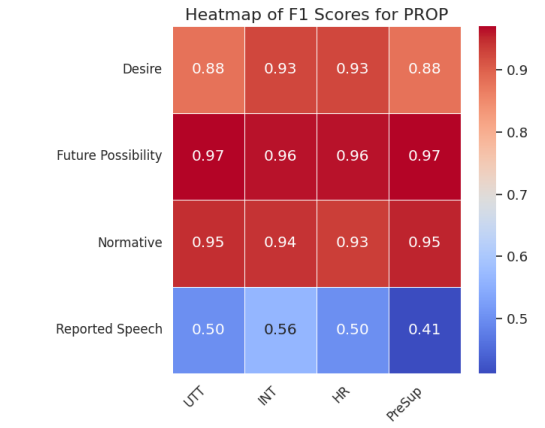
(b) P4G



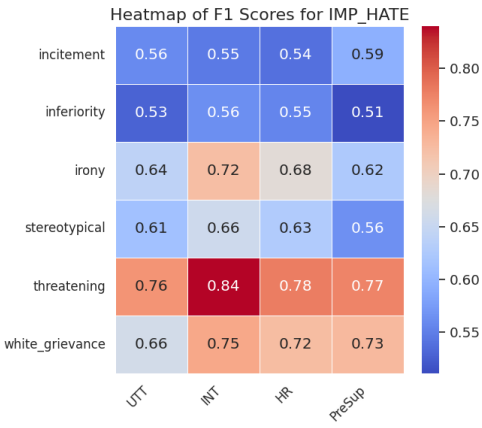
(c) res_CB



(d) EMH

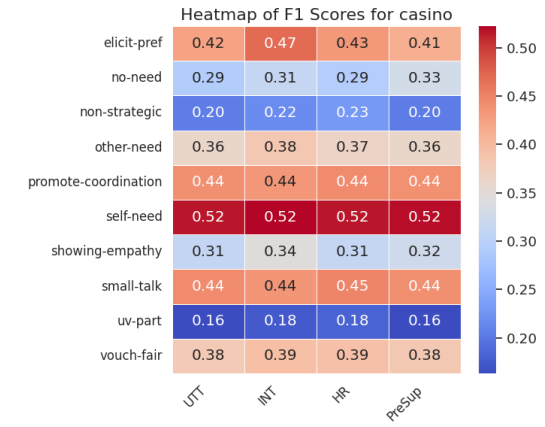


(e) PROP

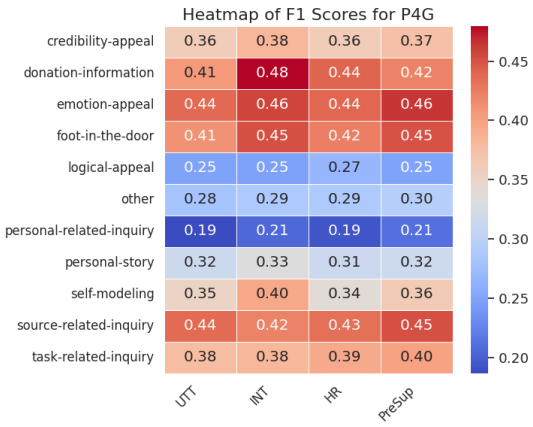


(f) IMP_HATE

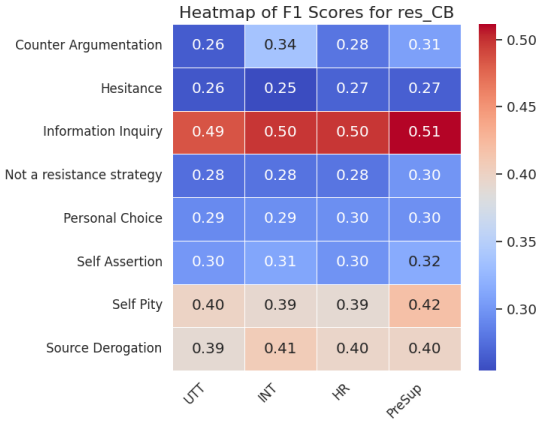
Figure 15: Comparative performance of rationales in terms of macro F1 score across different labels for different tasks in an indomain setting.



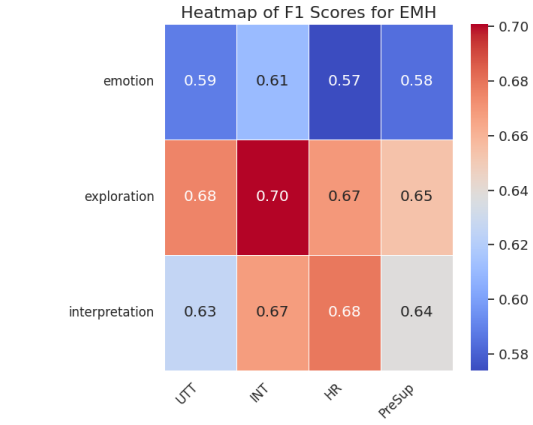
(a) Casino



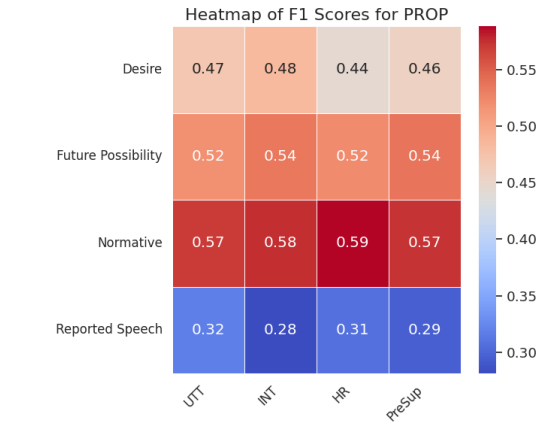
(b) P4G



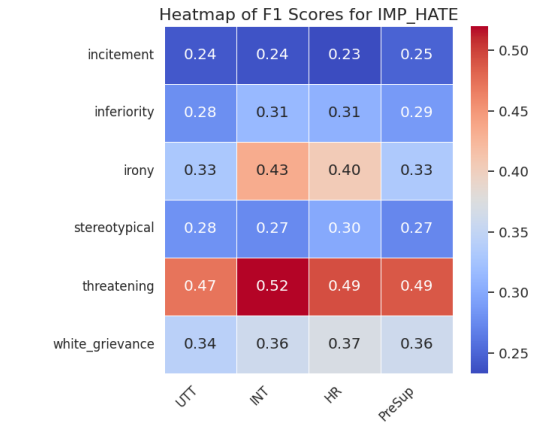
(c) res_CB



(d) EMH

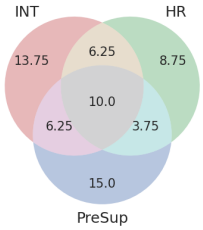


(e) PROP

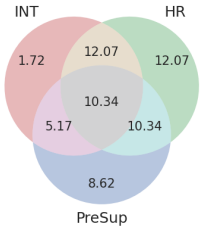


(f) IMP_HATE

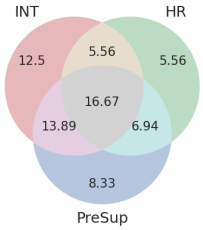
Figure 16: Comparative performance of rationales in terms of macro F1 score across different labels for the different target tasks in a cross-task setting



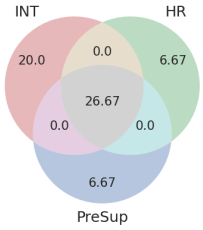
(a) CaSiNo



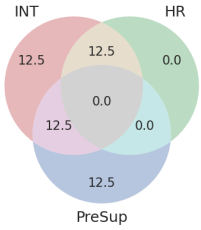
(b) P4G



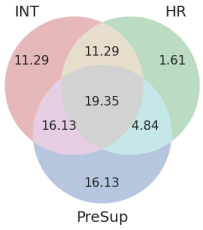
(c) res_CB



(d) EMH

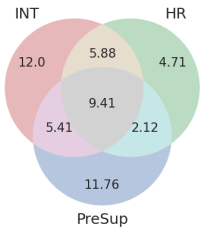


(e) PROP

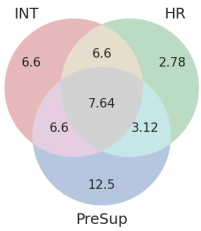


(f) IMP_HATE

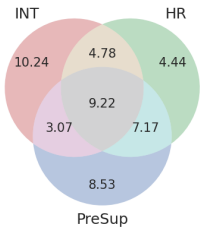
Figure 17: Venn Diagram showing the proportion of instances where including the rationales fared better than the baseline in an in domain setting.



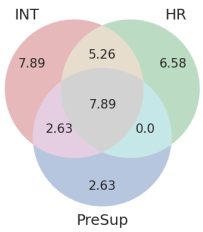
(a) CaSiNo



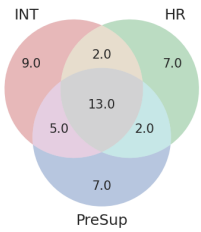
(b) P4G



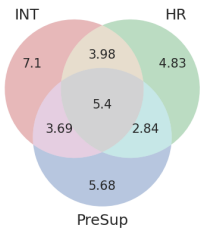
(c) res_CB



(d) EMH



(e) PROP



(f) IMP_HATE

Figure 18: Venn Diagram showing the proportion of instances where including the rationales fared better than the baseline in a 5-shot transfer setting.

Appendix C

Dataset	Source	Encoder	Parser	Sent Length		Lex Length		Dep Length		# Docs	# Rels
				Mean	Median	Mean	Median	Mean	Median		
Indore	en	mBERT	stanza	31.23	29.0	13.92	11.0	5.43	5.0	8486	51
	hi	mBERT	stanza	66.76	56.0	27.29	21.0	5.70	5.0	6963	51
	te	mBERT	stanza	151.45	140.0	44.74	31.0	6.05	6.0	8154	51
	en	mBERT	trankit	31.23	29.0	13.92	11.0	5.42	5.0	8486	51
	hi	mBERT	trankit	66.76	56.0	27.29	21.0	5.85	6.0	6963	51
	te	mBERT	trankit	151.45	140.0	44.74	31.0	5.69	6.0	8154	51
	en	XLMR	stanza	34.40	32.0	15.95	13.0	5.43	5.0	8486	51
	hi	XLMR	stanza	56.25	48.0	22.85	17.0	5.70	5.0	6963	51
	te	XLMR	stanza	125.60	113.0	37.43	26.0	6.05	6.0	8154	51
	en	XLMR	trankit	34.40	32.0	15.95	13.0	5.42	5.0	8486	51
	hi	XLMR	trankit	56.25	48.0	22.85	17.0	5.85	6.0	6963	51
	te	XLMR	trankit	125.60	113.0	37.43	26.0	5.69	6.0	8154	51
RedFM	en	mBERT	stanza	117.53	107.0	27.96	17.0	6.40	6.0	10899	32
	es	mBERT	stanza	103.77	91.0	25.97	17.0	6.24	6.0	6538	32
	fr	mBERT	stanza	92.18	78.0	23.19	15.0	5.86	5.0	7383	32
	it	mBERT	stanza	79.31	65.0	20.56	14.0	5.80	5.0	6812	32
	de	mBERT	stanza	88.79	79.0	22.97	15.0	5.28	5.0	7497	32
	ar	mBERT	stanza	107.15	90.0	29.68	22.0	6.08	6.0	1846	32
	zh	mBERT	stanza	118.18	101.0	35.25	22.0	6.69	6.0	1384	32
	en	mBERT	trankit	117.53	107.0	27.96	17.0	6.37	6.0	10899	32
	es	mBERT	trankit	103.77	91.0	25.97	17.0	6.16	6.0	6538	32
	fr	mBERT	trankit	92.18	78.0	23.19	15.0	5.68	5.0	7383	32
	it	mBERT	trankit	79.31	65.0	20.56	14.0	5.64	5.0	6812	32
	de	mBERT	trankit	88.79	79.0	22.97	15.0	5.16	5.0	7497	32
	ar	mBERT	trankit	107.15	90.0	29.68	22.0	6.18	6.0	1846	32
	zh	mBERT	trankit	118.18	101.0	35.25	22.0	6.75	6.0	1384	32
	en	XLMR	stanza	130.33	119.0	31.52	19.0	6.40	6.0	10899	32
	es	XLMR	stanza	112.22	100.0	28.86	19.0	6.24	6.0	6538	32
	fr	XLMR	stanza	103.20	86.0	26.77	17.0	5.86	5.0	7383	32
	it	XLMR	stanza	85.14	71.0	22.72	16.0	5.80	5.0	6812	32
	de	XLMR	stanza	96.23	87.0	25.53	17.0	5.28	5.0	7497	32
	ar	XLMR	stanza	95.81	81.0	26.64	19.0	6.08	6.0	1846	32
	zh	XLMR	stanza	96.10	81.0	28.35	18.0	6.69	6.0	1384	32
	en	XLMR	trankit	130.33	119.0	31.52	19.0	6.37	6.0	10899	32
	es	XLMR	trankit	112.22	100.0	28.86	19.0	6.16	6.0	6538	32
	fr	XLMR	trankit	103.20	86.0	26.77	17.0	5.68	5.0	7383	32
	it	XLMR	trankit	85.14	71.0	22.72	16.0	5.64	5.0	6812	32
	de	XLMR	trankit	96.23	87.0	25.53	17.0	5.16	5.0	7497	32
	ar	XLMR	trankit	95.81	81.0	26.64	19.0	6.18	6.0	1846	32
	zh	XLMR	trankit	96.10	81.0	28.35	18.0	6.75	6.0	1384	32

Table 18: Combined Statistics for Indore and RedFM Datasets

Without Any Dependency Information:

Given the sentence: ”The Porsche Panamera is a mid/full-sized luxury vehicle (E-segment in Europe) manufactured by the $< e2 >$German¡/e2¿ automobile manufacturer $< e1 >$Porsche¡/e1¿. It is front-engined and has a rear-wheel-drive layout, with all-wheel drive versions also available.”, which one of the following relations between the two entities $< e1 >$ and $< e2 >$ is being discussed? Choose one from this list of 32 options:\n0: country\n1: place of birth \n2: spouse\n3: country of citizenship\n4: instance of\n5: capital\n6: child\n7: shares border with\n8: author\n9: director\n10: occupation\n11: founded by\n12: league\n13: owned by\n14: genre\n15: named after\n16: follows\n17: headquarters location\n18: cast member\n19: manufacturer\n20: located in or next to body of water\n21: location\n22: part of\n23: mouth of the watercourse\n24: member of\n25: sport \n26: characters\n27: participant\n28: notable work\n29: replaces\n30: sibling\n31: inception\n\n. The answer is :

Tuple Format Prompt:

Given the sentence: ”The Porsche Panamera is a mid/full-sized luxury vehicle (E-segment in Europe) manufactured by the $< e2 >$German¡/e2¿ automobile manufacturer $< e1 >$Porsche¡/e1¿. It is front-engined and has a rear-wheel-drive layout, with all-wheel drive versions also available.”, which one of the following relations between the two entities $< e1 >$ and $< e2 >$ is being discussed? We also provide the dependency parse in the form of head, rel, and word: {”head”: ”Panamera”, ”rel”: ”det”, ”word”: ”The”}, {”head”: ”Panamera”, ”rel”: ”compound”, ”word”: ”Porsche”}, {”head”: ”vehicle”, ”rel”: ”nsubj”, ”word”: ”Panamera”}, {”head”: ”vehicle”, ”rel”: ”cop”, ”word”: ”is”}, {”head”: ”vehicle”, ”rel”: ”det”, ”word”: ”a”}, {”head”: ”sized”, ”rel”: ”compound”, ”word”: ”mid”}, {”head”: ”sized”, ”rel”: ”punct”, ”word”: ”/”}, {”head”: ”sized”, ”rel”: ”amod”, ”word”: ”full”}, {”head”: ”sized”, ”rel”: ”punct”, ”word”: ”-”}, {”head”: ”vehicle”, ”rel”: ”amod”, ”word”: ”sized”}, {”head”: ”vehicle”, ”rel”: ”compound”, ”word”: ”luxury”}, {”head”: ”ROOT”, ”rel”: ”root”, ”word”: ”vehicle”}, {”head”: ”segment”, ”rel”: ”punct”, ”word”: ”(”}, {”head”: ”segment”, ”rel”: ”compound”, ”word”: ”E”}, {”head”: ”segment”, ”rel”: ”punct”, ”word”: ”-”}, {”head”: ”vehicle”, ”rel”: ”appos”, ”word”: ”segment”}, {”head”: ”Europe”, ”rel”: ”case”, ”word”: ”in”}, {”head”: ”segment”, ”rel”: ”nmod”, ”word”: ”Europe”}, {”head”: ”segment”, ”rel”: ”punct”, ”word”: ”)”}, {”head”: ”vehicle”, ”rel”: ”acl”, ”word”: ”manufactured”}, {”head”: ”manufacturer”, ”rel”: ”case”, ”word”: ”by”}, {”head”: ”manufacturer”, ”rel”: ”det”, ”word”: ”the”}, {”head”: ”manufacturer”, ”rel”: ”amod”, ”word”: ”German”}, {”head”: ”manufacturer”, ”rel”: ”compound”, ”word”: ”automobile”}, {”head”: ”manufactured”, ”rel”: ”obl”, ”word”: ”manufacturer”}, {”head”: ”manufacturer”, ”rel”: ”appos”, ”word”: ”Porsche”}, {”head”: ”vehicle”, ”rel”: ”punct”, ”word”: ”.”}, {”head”: ”engined”, ”rel”: ”nsubj”, ”word”: ”It”}, {”head”: ”engined”, ”rel”: ”cop”, ”word”: ”is”}, {”head”: ”engined”, ”rel”: ”obl:npmod”, ”word”: ”front”}, {”head”: ”engined”, ”rel”: ”punct”, ”word”: ”-”}, {”head”: ”ROOT”, ”rel”: ”root”, ”word”: ”engined”}, {”head”: ”has”, ”rel”: ”cc”, ”word”: ”and”}, {”head”: ”engined”, ”rel”: ”conj”, ”word”: ”has”}, {”head”: ”layout”, ”rel”: ”det”, ”word”: ”a”}, {”head”: ”drive”, ”rel”: ”amod”, ”word”: ”rear”}, {”head”: ”drive”, ”rel”: ”punct”, ”word”: ”-”}, {”head”: ”drive”, ”rel”: ”compound”, ”word”: ”wheel”}, {”head”: ”drive”, ”rel”: ”punct”, ”word”: ”-”}, {”head”: ”layout”, ”rel”: ”amod”, ”word”: ”drive”}, {”head”: ”has”, ”rel”: ”obj”, ”word”: ”layout”}, {”head”: ”layout”, ”rel”: ”punct”, ”word”: ”,”}, {”head”: ”available”, ”rel”: ”mark”, ”word”: ”with”}, {”head”: ”drive”, ”rel”: ”det”, ”word”: ”all”}, {”head”: ”drive”, ”rel”: ”punct”, ”word”: ”-”}, {”head”: ”drive”, ”rel”: ”compound”, ”word”: ”wheel”}, {”head”: ”versions”, ”rel”: ”compound”, ”word”: ”drive”}, {”head”: ”available”, ”rel”: ”nsubj”, ”word”: ”versions”}, {”head”: ”available”, ”rel”: ”advmod”, ”word”: ”also”}, {”head”: ”layout”, ”rel”: ”acl”, ”word”: ”available”}, {”head”: ”engined”, ”rel”: ”punct”, ”word”: ”.”}. Choose one from this list of 32 options:\n0: country\n1: place of birth \n2: spouse\n3: country of citizenship\n4: instance of\n5: capital\n6: child\n7: shares border with\n8: author\n9: director\n10: occupation\n11: founded by\n12: league\n13: owned by\n14: genre\n15: named after\n16: follows\n17: headquarters location\n18: cast member\n19: manufacturer\n20: located in or next to body of water\n21: location\n22: part of\n23: mouth of the watercourse\n24: member of\n25: sport \n26: characters\n27: participant\n28: notable work\n29: replaces\n30: sibling\n31: inception\n\n. The answer is :

Table 19: Prompt without dependency information and the tuple format prompt are used for relation extraction on the English subset of the RedFM dataset with Trankit as the dependency parser.

Text Prompt:

Given the sentence: T̈he Porsche Panamera is a mid/full-sized luxury vehicle (E-segment in Europe) manufactured by the $< e2 >$German¡/e2¿ automobile manufacturer $< e1 >$Porsche¡/e1¿. It is front-engined and has a rear-wheel-drive layout, with all-wheel drive versions also available.̈, which one of the following relations between the two entities $< e1 >$ and $< e2 >$ is being discussed?\We also provide the dependency parses as follows: T̈he is Determiner of Panamera, Porsche is Compound noun modifier of Panamera, Panamera is Nominal subject of vehicle, is is Copula of vehicle, a is Determiner of vehicle, mid/ is Adverbial modifier of sized, full is Adjectival modifier of sized, - is Punctuation of sized, sized is Adjectival modifier of vehicle, luxury is Compound noun modifier of vehicle, vehicle is the root word, (is Punctuation of E, E is Appositional modifier of vehicle, - is Punctuation of segment, segment is Unspecified dependency of E, in is Case marker of Europe, Europe is Nominal modifier of segment,) is Punctuation of segment, manufactured is Clausal modifier of noun of vehicle, by is Case marker of Porsche, the is Determiner of Porsche, German is Adjectival modifier of Porsche, automobile is Compound noun modifier of manufacturer, manufacturer is Compound noun modifier of Porsche, Porsche is Oblique nominal of manufactured, . is Punctuation of vehicle, It is Nominal subject of engined, is is Copula of engined, front is Adjectival modifier of engined, - is Punctuation of front, engined is the root word, and is Coordinating conjunction of has, has is Conjunction of engined, a is Determiner of layout, rear is Compound noun modifier of drive, - is Punctuation of wheel, wheel is Compound noun modifier of drive, - is Punctuation of drive, drive is Compound noun modifier of layout, layout is Object of has, , is Punctuation of available, with is Marker of available, all is Determiner of wheel, - is Punctuation of all, wheel is Compound noun modifier of drive, drive is Compound noun modifier of versions, versions is Nominal subject of available, also is Adverbial modifier of available, available is Adverbial clause modifier of has, . is Punctuation of engined, \Choose one from this list of 32 options:\n0: country\n1: place of birth \n2: spouse\n3: country of citizenship\n4: instance of\n5: capital\n6: child\n7: shares border with\n8: author\n9: director\n10: occupation\n11: founded by\n12: league\n13: owned by\n14: genre\n15: named after\n16: follows\n17: headquarters location\n18: cast member\n19: manufacturer\n20: located in or next to body of water\n21: location\n22: part of\n23: mouth of the watercourse\n24: member of\n25: sport \n26: characters\n27: participant\n28: notable work\n29: replaces\n30: sibling\n31: inception\n\nThe answer is : ”

Filtered Text Prompt:

Given the sentence: T̈he Porsche Panamera is a mid/full-sized luxury vehicle (E-segment in Europe) manufactured by the $< e2 >$German¡/e2¿ automobile manufacturer $< e1 >$Porsche¡/e1¿. It is front-engined and has a rear-wheel-drive layout, with all-wheel drive versions also available.̈, which one of the following relations between the two entities $< e1 >$ and $< e2 >$ is being discussed?\n We also provide the dependency parses as follows: P̈orsche is Adjectival modifier of German, \n Choose one from this list of 32 options:\n0: country\n1: place of birth \n2: spouse\n3: country of citizenship\n4: instance of\n5: capital\n6: child\n7: shares border with\n8: author\n9: director\n10: occupation\n11: founded by\n12: league\n13: owned by\n14: genre\n15: named after\n16: follows\n17: headquarters location\n18: cast member\n19: manufacturer\n20: located in or next to body of water\n21: location\n22: part of\n23: mouth of the watercourse\n24: member of\n25: sport \n26: characters\n27: participant\n28: notable work\n29: replaces\n30: sibling\n31: inception\n\n. The answer is :

Table 20: Text prompt and Filtered Text prompts used for relation extraction on the English subset of the RedFM dataset with Trankit as the dependency parser.

Table 21: Zero-shot cross-lingual performance for Relation Extraction on the RedFM dataset using mBERT, dependency parse information and GNN. Highest values in each column are in bold. The rows and columns correspond to the source and target language respectively.

			mBERT						
Src	DEP	GNN	en	es	fr	it	de	ar	zh
en	-	-	-	80.4±0.2	80.7±0.4	**77.3±1.3**	78.8±0.9	72.7±0.8	70.4±0.6
en	stanza	rgcn	-	79.6±0.8	**80.9±1.4**	76.2±1.0	**80.2±0.5**	74.4±0.9	72.0±0.8
en	stanza	rgat	-	80.3±0.4	80.3±0.2	74.8±1.2	79.5±0.3	74.1±0.9	**72.3±0.4**
en	trankit	rgcn	-	80.1±0.4	80.8±0.5	73.8±0.2	79.3±0.7	73.8±1.8	69.5±0.6
en	trankit	rgat	-	**80.8±0.3**	80.7±0.2	74.4±1.8	79.0±0.7	**74.5±0.7**	70.1±0.6
es	-	-	77.6±0.1	-	77.2±0.8	76.4±0.6	75.9±0.7	70.9±1.6	70.8±1.1
es	stanza	rgcn	78.0±0.4	-	**82.6±0.8**	77.6±1.4	76.9±1.3	73.2±0.5	69.2±0.6
es	stanza	rgat	79.1±0.2	-	78.4±0.5	77.4±1.3	76.2±0.7	**73.5±0.9**	69.7±1.5
es	trankit	rgcn	79.3±0.9	-	80.6±1.6	76.3±0.6	77.1±1.0	73.3±0.3	**71.5±1.6**
es	trankit	rgat	**80.0±1.1**	-	78.7±0.5	**78.3±1.0**	**77.7±0.8**	72.6±1.2	71.2±2.6
fr	-	-	76.6±2.9	80.4±1.3	-	76.9±2.0	74.8±1.6	70.2±1.1	66.4±2.6
fr	stanza	rgcn	76.6±0.3	82.1±1.0	-	77.7±0.7	76.6±0.2	70.4±0.8	66.8±0.9
fr	stanza	rgat	80.0±0.7	82.1±0.9	-	77.0±1.0	77.5±1.5	71.5±1.0	67.5±1.2
fr	trankit	rgcn	78.6±0.3	**83.3±1.6**	-	**78.7±1.1**	**78.8±2.5**	**72.4±0.5**	**69.7±0.7**
fr	trankit	rgat	**80.1±0.8**	79.7±2.1	-	76.6±1.5	77.4±0.1	70.9±0.8	68.4±0.5
it	-	-	75.4±0.4	83.1±0.5	77.7±1.1	-	72.9±1.1	**73.0±2.0**	70.8±1.0
it	stanza	rgcn	**79.0±0.6**	83.0±0.7	77.2±1.0	-	74.7±1.4	70.8±0.3	70.0±0.7
it	stanza	rgat	76.7±0.9	**83.8±0.7**	77.5±0.5	-	75.7±1.5	72.2±1.6	70.5±0.4
it	trankit	rgcn	77.1±1.4	82.3±0.3	77.2±0.6	-	76.0±1.2	71.0±1.0	69.2±1.9
it	trankit	rgat	77.1±0.1	82.5±0.4	**77.8±0.5**	-	**76.3±0.1**	71.7±1.0	**71.5±0.9**
de	-	-	80.4±1.0	80.0±0.4	**78.3±0.1**	76.1±1.5	-	**75.8±1.9**	71.6±1.2
de	stanza	rgcn	80.0±0.2	80.4±0.7	76.7±0.3	75.8±0.8	-	74.2±0.8	70.0±1.9
de	stanza	rgat	79.2±0.4	**81.3±1.1**	78.1±1.4	76.6±2.7	-	74.6±0.5	**71.7±0.6**
de	trankit	rgcn	79.7±0.3	80.6±1.4	77.9±0.3	75.1±0.4	-	73.3±1.0	70.1±0.1
de	trankit	rgat	**80.7±0.7**	79.2±0.1	77.8±0.6	**77.4±0.5**	-	73.7±0.0	70.6±0.8

Table 22: Zero-shot cross-lingual performance for Relation Extraction on the RedFM dataset using XLMR and dependency parse information and GNN. Highest values in each column are in bold. The rows and columns correspond to the source and target language respectively.

Src	DEP	GNN	en	es	fr	it	de	ar	zh
						XLMR			
en	-	-	-	73.1±1.8	**72.8±2.8**	64.2±3.7	**75.6±1.7**	61.7±1.8	64.4±1.0
en	stanza	rgcn	-	74.4±1.3	**72.7±0.5**	**67.4±1.3**	74.6±0.7	**63.2±1.5**	65.1±0.9
en	stanza	rgat	-	73.1±0.7	**72.7±1.4**	66.5±3.5	71.1±1.0	59.6±2.7	62.2±0.4
en	trankit	rgcn	-	74.4±1.5	72.0±1.8	65.4±2.2	71.5±1.6	62.6±1.8	64.6±1.3
en	trankit	rgat	-	**74.9±0.7**	70.3±0.1	62.4±1.6	73.9±0.4	61.5±1.7	**66.5±1.7**
es	-	-	73.3±0.4	-	74.3±0.4	**70.1±1.4**	**70.6±0.7**	63.2±3.1	**65.9±1.9**
es	stanza	rgcn	73.4±2.2	-	75.1±0.3	68.3±2.5	67.3±0.6	61.9±1.2	62.4±1.4
es	stanza	rgat	72.7±1.9	-	75.2±1.0	69.3±1.6	67.3±0.3	60.5±1.4	62.8±1.8
es	trankit	rgcn	**73.8±1.0**	-	75.9±1.5	69.8±1.8	70.0±2.5	**64.3±2.1**	65.6±2.7
es	trankit	rgat	71.4±1.2	-	**76.2±1.2**	68.0±1.5	68.7±2.0	60.0±0.9	62.5±2.3
fr	-	-	71.1±0.9	75.0±0.6	-	68.9±0.6	68.5±1.3	**61.5±1.2**	59.4±2.6
fr	stanza	rgcn	**74.3±1.7**	74.1±1.1	-	**69.7±0.6**	**72.2±1.3**	58.7±0.6	**62.9±2.7**
fr	stanza	rgat	70.1±1.5	73.9±1.3	-	67.0±1.5	66.2±1.0	59.0±0.9	60.3±1.6
fr	trankit	rgcn	70.0±0.2	74.4±0.5	-	68.4±0.7	66.4±0.7	58.9±2.2	59.5±1.8
fr	trankit	rgat	71.8±1.3	**76.0±0.7**	-	68.2±0.8	70.6±1.0	**61.5±1.2**	59.9±1.3
it	-	-	71.2±1.1	76.1±1.6	72.2±0.9	-	68.2±1.7	60.8±0.5	62.0±1.7
it	stanza	rgcn	73.3±2.0	76.1±0.8	74.3±1.3	-	67.2±2.1	61.8±0.3	63.1±0.3
it	stanza	rgat	**74.9±1.0**	76.0±0.2	74.2±1.3	-	68.9±0.2	62.2±0.1	**64.7±1.5**
it	trankit	rgcn	73.3±1.2	77.0±0.7	74.8±1.6	-	70.0±1.7	**64.5±1.0**	**64.7±1.0**
it	trankit	rgat	72.6±1.9	**78.7±0.5**	**76.6±0.2**	-	**70.2±1.0**	63.6±3.4	64.6±1.5
de	-	-	75.0±1.5	72.4±0.9	69.3±1.3	64.1±0.3	-	60.8±0.7	64.0±1.2
de	stanza	rgcn	72.6±1.5	73.4±2.1	70.8±1.9	65.2±0.5	-	60.6±0.8	**66.0±1.9**
de	stanza	rgat	**76.1±1.5**	**73.5±0.2**	**71.5±1.3**	**69.0±2.8**	-	**64.0±1.6**	65.8±1.7
de	trankit	rgcn	74.1±1.0	72.8±0.8	69.6±1.8	63.6±2.3	-	63.4±1.0	64.5±1.9
de	trankit	rgat	75.0±0.5	73.2±1.6	70.3±1.3	64.9±1.0	-	63.7±0.5	64.4±3.5

Table 23: Zero-shot cross-lingual performance for Relation Extraction on the IndoRE dataset using different combinations of multi-lingual encoder and dependency parse information and GNN. Highest values in each column are in bold. The rows and columns correspond to the source and target language respectively.

mBERT					
Src	DEP	GNN	en	hi	te
en	-	-	-	60.7±0.6	35.3±0.8
en	stanza	rgcn	-	60.1±0.4	38.3±1.2
en	stanza	rgat	-	58.7±0.3	**40.6±2.2**
en	trankit	rgcn	-	**62.5±0.8**	38.0±1.4
en	trankit	rgat	-	61.8±1.0	37.8±1.8
hi	-	-	**69.7±1.9**	-	49.5±2.3
hi	stanza	rgcn	68.6±0.6	-	49.4±0.8
hi	stanza	rgat	67.8±2.3	-	49.7±0.6
hi	trankit	rgcn	68.1±0.8	-	49.6±2.2
hi	trankit	rgat	68.0±1.6	-	**53.9±0.9**
te	-	-	45.3±1.7	54.4±2.6	-
te	stanza	rgcn	45.6±1.4	54.0±1.3	-
te	stanza	rgat	44.8±0.3	**56.6±0.3**	-
te	trankit	rgcn	**47.7±0.8**	54.2±0.1	-
te	trankit	rgat	46.1±1.2	54.2±2.5	-
XLMR					
en	-	-	-	57.4±2.3	37.2±2.5
en	stanza	rgcn	-	55.3±1.2	37.0±1.6
en	stanza	rgat	-	55.5±2.3	37.8±1.9
en	trankit	rgcn	-	58.8±0.5	36.4±3.8
en	trankit	rgat	-	**61.0±2.5**	**39.0±4.0**
hi	-	-	59.1±1.8	-	53.7±1.0
hi	stanza	rgcn	57.4±1.3	-	**54.7±1.2**
hi	stanza	rgat	**61.0±2.5**	-	**54.8±2.1**
hi	trankit	rgcn	59.5±0.8	-	54.3±1.8
hi	trankit	rgat	57.3±2.4	-	**54.8±2.3**
te	-	-	40.9±2.6	52.8±0.7	-
te	stanza	rgcn	41.2±2.2	**55.5±0.9**	-
te	stanza	rgat	39.0±0.7	52.0±3.2	-
te	trankit	rgcn	**41.8±0.6**	53.7±0.6	-
te	trankit	rgat	41.4±0.3	53.7±1.8	-